HAPLOTYPE PHASING BY MULTI-ASSEMBLY OF SHARED HAPLOTYPES: PHASE-DEPENDENT INTERACTIONS BETWEEN RARE VARIANTS*

BJARNI V. HALLDÓRSSON[1,†], DEREK AGUIAR[2,3,†] AND SORIN ISTRAIL[2,3]

[1]*School of Science and Engineering, Reykjavik University, Reykjavik, Iceland*

[2]*Department of Computer Science, Brown University, Providence, RI, USA*

[3]*Center for Computational Molecular Biology, Brown University, Providence, RI, USA*

In this paper we propose algorithmic strategies, Lander-Waterman-like statistical estimates, and genome-wide software for haplotype phasing by multi-assembly of shared haplotypes. Specifically, we consider four types of results which together provide a comprehensive workflow of GWAS data sets: (1) statistics of multi-assembly of shared haplotypes (2) graph theoretic algorithms for haplotype assembly based on conflict graphs of sequencing reads (3) inference of pedigree structure through haplotype sharing via tract finding algorithms and (4) multi-assembly of shared haplotypes of cases, controls, and trios. The input for the workflows that we consider are any of the combination of: (A) genotype data (B) next generation sequencing (NGS) (C) pedigree information.

(1) We present Lander-Waterman-like statistics for NGS projects for the multi-assembly of shared haplotypes. Results are presented in Sec. 2. (2) In Sec. 3, we present algorithmic strategies for haplotype assembly using NGS, NGS + genotype data, and NGS + pedigree information. (3) This work builds on algorithms presented in Halldórsson *et al.*[1] and are part of the same library of tools co-developed for GWAS workflows. (4) Section 3.3.1 contains algorithmic strategies for multi-assembly of GWAS data. We present algorithms for assembling large data sets and for determining and using shared haplotypes to more reliably assemble and phase the data. Workflows 1-4 provide a set of rigorous algorithms which have the potential to identify phase-dependent interactions between rare variants in linkage equilibrium which are associated with cases. They build on our extensive work on haplotype phasing,[1–3] haplotype assembly,[4,5] and whole genome assembly comparison.[6]

Keywords: haplotype assembly; haplotype inference; rare variants; phasing; phase inference

1. Introduction

> *Improving data quality is crucial, because if a human genome cannot be independently assembled then the sequence data cannot be sorted into the two sets of parental chromosomes, or haplotypes. This process – haplotype phasing – will become one of the most useful tools in genomic medicine.* – J. Craig Venter, 2010[7]

A genome-wide association study (GWAS) is a leading approach to find genetic determinants associated with a particular phenotype.[8–10] GWAS proceed by identifying a set of individuals carrying a disease or trait (cases) and a set of individuals that do not (controls).

*corresponding authors are Bjarni V. Halldórsson bjarnivh@ru.is and Sorin Istrail sorin@cs.brown.edu
†Authors contributed equally to this work

The cases and controls are then genotyped for a large number of common single nucleotide polymorphism (SNP) genetic variants which are then statistically tested for association to some disease or trait. GWAS have been successful in identifying many common genetic risk variants to many diseases,[8,11,12] but many associations appear to have no known connection to biological mechanisms and thus cannot be targeted for clinical intervention. Furthermore, some of these studies reveal a paradoxically encouraging and, at the same time, disappointing theme for complex traits: a set of SNPs are found to be highly statistically significant (and are often replicated in subsequent studies) yet individually, and in aggregate, these SNPs only explain a very small proportion of genetic variance.[13]

The problem of interpreting the low explicative and predictive power of these variants has been deemed the "missing heritability" problem. Many hypotheses have been presented to explain the missing heritability.[14] Most echo caveats frequently associated with GWAS such as difficulties with defining phenotypes of cases and controls, cryptic population stratification, common variation that is often left out of GWAS (copy number variation or gaps in SNP coverage), or environmental factors.[14,15] We concern ourselves with two explanations that have received much attention recently: *phase-dependent interactions* and *rare variation*.

Knowledge of haplotypes can greatly increase the power of GWAS studies and also highlight associations that are impossible to detect without haplotype phase (e.g. loss of heterozygosity). Even more complicated phase-dependent interactions of variants in linkage equilibrium have also been suggested as possible causes of missing heritability.[14] The actual haplotypes in the typed region can be obtained only at considerably higher experimental cost or via computational haplotype phasing for which most algorithms fail to work on genome-wide data. For these reasons, GWAS have generally ignored phase-dependent interactions or associations.

Although the significance of phase-dependent interactions is yet to be determined, rare variation is now accepted as playing a significant role in many common diseases[16–18] as well as rare diseases.[19,20] SNP arrays used for GWAS are designed to tag common variants only, thus rare variant associations are ignored. However, with cost of next-generation sequencing decreasing rapidly and the sequencing of tens of thousands of individuals already underway,[21,22] GWAS are likely to develop novel approaches for association. Anticipating this data will soon be available, we have developed algorithms to simultaneously identify rare variation and determine the haplotype phase of a number of individuals using sequence reads.

The class of algorithms that use sequence reads to infer the haplotypes of a diploid organism are called *haplotype assembly* algorithms.[5,23] Early formulations focused on assembling the haplotypes from the reads of one individual. Because most bases on a read are identical regardless of the chromosome of origin, the reads can be mapped to a reference genome. After mapping, reads are translated into *haplotype fragments* containing only the polymorphic (SNP) sites. A fragment *covers* a site if the corresponding read contains the SNP. Fragments that cover more than one SNP site provide valuable phase information, that is, if two SNPs co-occur on one fragment then they exist on the same haplotype. Thus, the input to the haplotype assembly problem is an $m \times n$ SNP matrix M whose m rows correspond to fragments $f_1, ..., f_m$ and each fragment f_i covers at least 2 of the n SNP sites. Formally, we define a fragment f_i as a vector of $\{0, 1, -\}$ where 0 and 1 represent the major and minor alleles at some site and '-'

represent a lack of information either because the read does not cover the SNP site or there was a technical failure (e.g. in mapping or sequencing). Two fragments i and j *conflict* if they cover a common SNP and have different alleles at that site (Fig. 1).

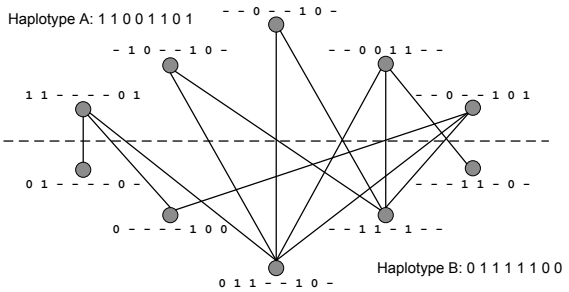

Fig. 1. Fragment conflict graph. Major and minor alleles are denoted by 0 and 1 respectively. Fragments from haplotype A (11001101) appear above the dotted line while fragments from haplotype B (01111100) appear below. The bipartition which separates the two sets of fragments is denoted by the dotted line. The haplotypes may be reconstructed by combining the shores of the bipartite graph.

Most haplotype assembly algorithms refer to an abstraction on M called the *fragment conflict graph*. The *fragment conflict graph*, $G_F(M) = (V, E)$, has nodes $f_i \in V \ \forall i$ and edges $\{f_i, f_j\} \in E$ if f_i and f_j conflict $\forall i, j$. Figure 1 demonstrates the translation from fragments to the fragment conflict graph. If the data is free of errors then, for each connected component in the fragment conflict graph, the vertices can be divided into two independent sets, that is, the graph is bipartite. Therefore, haplotype assembly of one individual can be expressed as: *Given the fragment conflict graph $G_F(M)$, find the underlying bipartite graph whose shores define the haplotypes of the individual.*

We present a novel approach to GWAS with sequence data of assembling the haplotypes of cases and controls using paired end sequencing reads and long range sharing information. Multi-assembly of GWAS sequence data has the power to enhance the discrepancies between cases and controls by phasing haplotypes using shared haplotype tracts. By assembling the cases and controls together, we can avoid missing marginal SNP variation at the level of misassembly that are associated with rare SNP variants. If the pedigree structure is known or long-range sharing information can be inferred, we can strengthen the multi-assembly by using the combined fragment coverage on the shared haplotype. First, we give a formula that relates a number of statistics/parameters with the coverage of SNPs on the haplotypes of many individuals. We present an efficient algorithm for finding the shared haplotype of two individuals in the fragment conflict graph. In addition, we present an efficient algorithmic strategy to resolve errors and assemble fragment conflict graph components that is capable of assembling genome-wide data. We employ methods that have been previously used for haplotype assembly as well as methods that have been applied to haplotype phasing.[1]

2. Multi-Assembly of Shared Haplotypes

The input for the haplotype assembly of multiple individuals problem is the same $m \times n$ matrix M in the case of one individual with an additional annotation on the fragments denoting the person of origin. In this section, we estimate the coverage needed to assemble haplotypes of multiple individuals. Consider the parameters:

G The length of the genome
S The number of SNPs in the genome
L The average length of a read
N The number of reads
c Coverage $= \frac{LN}{G}$

For these calculations we assume the distance between SNPs and sequence reads follow a Poisson distribution ($\lambda = 400$ and $\lambda = 10000$ respectively). A read is considered a "good-read" if it covers at least two SNPs. The intuition behind good-reads is that they determine the haplotype phase of two or more SNPs (they are on the same haplotype). The probability that a paired read covers at least 2 SNPs is $\geq \left(\frac{L}{400}\right)^2$. L is assumed to be ≤ 400. Let s_i be a SNP on chromosome i in position p. s_i is covered if a read starts in the interval $(p-L, p]$ on chromosome i. The expected number of reads starting in the interval $(p-L, p]$ on chromosome i is $\geq \frac{LN\left(\frac{L}{400}\right)^2}{2G} = \left(\frac{c}{2}\right)\left(\frac{L}{400}\right)^2$. The probability that no reads start in $(p-L, p]$ is $e^{-\left(\frac{c}{2}\right)\left(\frac{L}{400}\right)^2}$.

$$P\left(> 0 \textit{ reads start in } (p-L, p]\right) = 1 - e^{-\left(\frac{c}{2}\right)\left(\frac{L}{400}\right)^2} \tag{1}$$

Thus, the number of the SNPs covered by good-reads is approximately

$$\left(1 - e^{-\left(\frac{c}{2}\right)\left(\frac{L}{400}\right)^2}\right) 2S \tag{2}$$

Enhanced coverage due to sharing. The coverage needed greatly depends on the probability of a good-read. A high good-read probability may be obtained through targeted sequencing, mate pairs, or larger read lengths. When multiple individuals are considered, the coverage needed may be greatly reduced if the haplotype sharing is high. Sequence reads from different individuals, but on a shared haplotype, can be considered as originating from the same chromosome and assembled together. This increases the effective coverage in Equation 2. For example, three unrelated individuals have 6 unique haplotypes for an effective genome size of 6G. A trio of individuals consisting of a child, mother, and father have 4 unique haplotypes for an effective genome size of 4G. Thus, for the same amount of reads you can achieve 50% more coverage on trios than unrelated individuals.

Building on the Lander-Waterman type of statistical analysis,[24] we can estimate two important statistical parameters of haplotype assemblies that guide our algorithms: (1) What is the coverage needed so that we cover X% of SNPs on both haplotypes of a single individual with a good sequence read? (2) What is the coverage needed so that we cover X% of SNPs on both haplotypes of a trio of individuals with a good sequence read? Table 1 shows estimates of coverage needed to cover a percentage of SNPs for a single individual and trios for different parameters.

3. Algorithmic Strategies

Finding haplotype assemblies for a single individual has been considered by several researchers.[2,25–27] This can be formulated as an approximate bipartition problem, where the bipartition stems from the fact that an individual is expected to have two chromosomes of each type and the approximation stems from the fact that some of the graph edges or vertices

Table 1. Length of genome $G = 3.2 \times 10^9$. Number of SNPs $S = \frac{G}{400} = 8 \times 10^6$.

Number of Individuals	Read Length	Coverage	Mean % of SNPs Covered
1	100	1	3
1	100	2	6
1	100	4	12
1	100	10	27
1	100	20	46
1	200	1	12
1	200	2	22
1	200	4	39
1	200	10	71
1	200	20	92
3 (trio)	200	1	17
3 (trio)	200	2	31
3 (trio)	200	4	53
3 (trio)	200	10	85
3 (trio)	200	20	98

are spurious. Spurious edges occur when there is a genotyping error, some sort of error in the lab protocol or an error in the mapping of reads.

Extensions of the single individual haplotype assembly include those that employ genotype data.[28–30] In these algorithms, genotype data is used to correct errors after sequence reads are mapped. However, genotype data is prone to errors and probe common SNPs only which are not helpful regarding rare and other non-probed variation. For multiple individuals, genotype data can be used to infer evolutionary relationships between haplotypes where pedigree data is not available.[1,31,32]

3.1. *Optimization Formulations*

The minimum fragment removal formulation introduced in Lancia *et al.*[5] and minimum error correction formulation (sometimes referred to as minimum letter flip) introduced in Lippert *et al.*[4] are two optimization formulations useful for the purposes of generalizing to multiple individuals. For k individuals, Li *et al.* show that a fragment conflict graph is feasible if and only if it is $2k$-colorable. However, in the case of identical by descent haplotype sharing, there are less than $2k$ unique haplotypes. Given some haplotype sharing is likely to exist, we can rewrite the optimization problems for multiple individuals.

(1) *Minimum Fragment Removal for k Individuals (k-MFR)*: Given a SNP matrix M of fragments from k individuals, remove the minimum number of fragments (rows) such that the resulting fragments can be combined to form *at most* $2k$ haplotypes.
(2) *Minimum Error Correction for k Individuals (k-MEC)*: Given a SNP matrix M of fragments from k individuals, correct the minimum number of errors in fragments such that the resulting fragments can be combined to form *at most* $2k$ haplotypes.

A correction of an error is defined as a flip from 0 to 1 or 1 to 0. A *gapless fragment* is a fragment covering a contiguous set of SNPs. MFR, MEC, k-MFR, and k-MEC using gapless fragments are tractable and useful problems when the read length is long enough to cover multiple SNPs. However, given the smaller read length sizes of next-generation sequencing, haplotype assembly is most effective with mate paired reads. MFR, MEC, k-MFR, and k-MEC using gapped fragments have been shown to be NP-hard.[5,33]

3.2. *Problem Formulations*

Problem 3.1. *Given a set of reads from k individuals, determine the minimum number of fragments to be removed such that the remaining fragments can be assembled into* $2k$ *haplotypes.*

Li *et al.*[33] give an IP formulation and a parameterized algorithm which is exponential in the number of individuals. The problem formulation is somewhat simplistic as it does not assume that it is known from which individual the reads are from.

Problem 3.2. *Given a set of reads from k individuals, with each read labeled with an individual, determine the minimum number of fragments to be removed such that the remaining fragments can be assembled into* $2k$ *haplotypes and the individual associated with the haplotypes and fragments agree.*

The equivalent IP formulation can be seen by adding the following constraint to the Li *et al.*[33] formulation: reads labeled with an individual must be included in the assembly of that individual. For problem instances with no errors, the integer program has a very nice decomposition, since the set of constraints for each individual require it to perform a bipartite graphs. It is also likely to be quite efficient since finding bipartite graphs is easy. However, real data contains errors from miscalls and erroneous read mappings.

Problem 3.3. *Given a set of reads, each labeled with an individual, find the minimum number of haplotypes such that (1) each individual is phased with exactly two haplotypes, (2) a minimum number of fragments are removed and (3) the individual associated with the haplotypes and fragments agree.*

For general graphs, this problem is NP-hard.[5,33] We suggest a heuristic algorithm which exploits the specific signatures of sequence read errors that we can find in the data and correct. Errors in the fragment conflict graph fall into three categories.

Category 1: A fragment would otherwise conflict with another fragment from the opposite chromosome but, due to an error, is consistent with fragments on the opposite chromosome but conflicts with fragments from the chromosome of origin.

Category 2: A fragment would otherwise not be included in the fragment conflict graph but acquires an error.

Category 3: Due to an error, a fragment conflicts with fragments from both haplotypes of the individual.

Category 1 has little effect on the fragment conflict graph. We would interpret the fragment as belonging to the wrong haplotype but this does not remove the bipartiteness of the graph.

Category 2 and Category 3 can remove bipartiteness of the graph and make the general MFR and MEC problem hard. However, given a high coverage, these two cases produce regular signatures in the fragment conflict graph; namely high degree nodes that conflict with fragments of both haplotypes.

An algorithmic strategy based on the architecture of these errors was implemented in Java and works well on simulations derived from HapMap and Hudson simulated data.[34] The algorithm begins by attempting to create a breadth first search tree of the fragment conflict graph. When the algorithm encounters a level of the BFS tree that does not fit in the biparition, it computes the 3-cliques (small conflicting sub-graphs of the fragment conflict graph) in the current BFS level and subsequent levels until zero 3-cliques are found by the addition of a BFS level. It then removes the fragment belonging to the most 3-clique conflicts. As a tie breaker, the algorithm removes the node with the highest degree. Since an erroneous fragment conflicts with fragments from both chromosomes it should belong to many 3-cliques and/or have a high degree. Also, because the number of conflicting fragments in a dataset is usually small, the algorithm runs in speed comparable to BFS.

3.3. *Assembly when Haplotype Sharing is Known*

If sharing information is unknown, assembling multiple individuals can help identify Category 1 errors. If haplotype sharing information is known or can be inferred, assembling multiple individuals simultaneously provides additional information on the coverage of haplotypes. The sharing of haplotypes between individuals could be known from pedigree data[35] or inferred.[1,31,32,36]

Problem 3.4. *We are given a set of individuals and for each pair of individuals, haplotype sharing information is known or can be inferred. We are also given a set of paired end sequencing reads for each individual. Output a pair of haplotypes for each individual such that each individual sharing a haplotype do so.*

If haplotype sharing information is unknown, we begin by inferring pedigree information using the tract finding algorithm[1] or similar methods.[31,32] We then build the fragment conflict graph. *The only edges that are informative are edges between fragments from the same individual and other individuals who share a haplotype identical by descent (IBD).*[35] If a segment of a haplotype is shared identical by state (IBS) then it is likely to conflict in other places on the haplotype and can yield a feasible but erroneous assembly. In addition, if we interpret these non conflicts as IBD, then can obtain the wrong coverage estimate on the haplotype which is essential for phasing the assembly.

3.3.1. *Haplotype sharing algorithm*

When there are no errors in the reads of an individual the fragment reads will form a bipartite graph.[5] The fragments belonging to one of the two shores of the bipartite graph will form one of the haplotypes and the fragments belonging to the other shore will form the other haplotype. In the case when the bipartite graph is disconnected then each connected component may be considered separately. Given fragments from two individuals which are known to share a haplotype, we propose the following algorithm for the joint haplotype assembly of the two

individuals. Let the two individuals be denoted by i and j. In our algorithm α corresponds to the shared haplotype and β corresponds to the non-shared haplotype. We note that $\alpha_i = \alpha_j$, while $\beta_i \neq \beta_j$.

Algorithm 3.1.

```
def Branch( s )
   For each edge with an endpoint in the α_s and other endpoint, e,
   in H, identify the connected component, t, of G_i or G_j that contains e.
      Label the color of t that is connected to e as β.
      Label the other color of t as α.
      H ⟵ H − t
      Branch( t )
   For each tree, t ∈ H that has a color c that has edges connecting to α_s and β_s
      Label c as β
      Label the other color as α
      H ⟵ H − t
      Branch( t )

Determine Sharing
   Construct the fragment conflict graph, G. Let G_i and G_j be
   the restriction of G to i and j.
      Color each component G_i and G_j with two colors.
      If no such coloring exists, the algorithms fails.
   H ⟵ G
   While H ≠ ∅
      Find connected components t, s, s.t. s ∈ G_j ∩ H and t ∈ G_i ∩ H or s ∈ G_i ∩ H
      and t ∈ G_j ∩ H and a color β of s with an edge to both colors of t.
         If no such tree exists, choose s arbitrarily from G_i or G_j
            Arbitrarily label the colors of s as α and β.
      H ⟵ H − s.
      Branch( s)
```

The algorithm is motivated by the key observation that a haplotype cannot be shared if one of its fragments is connected to a color that is shared. We may observe a connection to a color that is shared either from the fact that the color is labeled as shared or it is connected to both colors.

Lemma 3.1. *The algorithm runs in $O(n+mn)$ time, where n is the number of fragments and m is the number of edges between the fragments.*

Proof. The initial step of coloring of a bipartite graph of each individual can be done in time $O(n+m)$. The edges that lie between two individuals can then be labeled with the component and color that they belong to. We then loop over each component of i in an outer loop and each component of j in an inner loop, followed by a loop over each component of j in an outer loop and each component of i in the inner loop. We determine whether there exists an edge from the component in the outer loop to both colors of the component in the inner loop. We observe that each edge will be visited at most as many times as there components in G. The number of components is upper bounded by n, for a total upper bound of nm. □

This algorithm presents an approach that may be generalized to more complex patterns of haplotype sharing.

3.4. *Phasing Components in the Fragment Conflict Graph*

Even with error-free data we aren't guaranteed to be able to assemble and phase the data. *Long runs of homozygosity form disconnected components in the fragment conflict graph.* Runs of homozygosity, which are paradoxically simple to phase, cause problems when assembling haplotypes. If the run of homozygosity is longer than the mate pair length no read can connect the two components as there wont be any conflicts in homozygous regions (Fig. 2). The more connected the graph is, the easier it is to phase because you have to eventually phase the shores of each component into two haplotypes. The number of valid haplotype phasings may therefore be large once the haplotypes of each individual have been assembled; if the haplotype assembly of a single individual consists of k disconnected bipartite components then there are 2^{k-1} unique ways to map the shores to haplotypes. Varying the mate pair read length, increasing the read length, adding coverage, or adding more individuals who may share a haplotype IBD help connect components together.

Fragments from haplotypes that are identical by descent can be considered when constructing bipartitions for both individuals. If two components need to be phased and one haplotype is shared then we'd expect the shared haplotype to have twice the coverage of the non-shared haplotype in both components, thus we phase the two shores with greater coverage from different components together. For example, Fig. 2 shows fragments from two haplotypes of two individuals one of which (10000001) is shared. The phasing of the two components is ambiguous but we know that the shared haplotype is likely to have approximately 50% more coverage. Therefore, it is more likely to phase the components such that we maximize the difference of cardinality between the phasings. For Fig. 2 the first phasing (10000001/00000000) yields $|6-3| = 3$ while the second phasing (10000000/00000001) yields $|5-4| = 1$. When phasing disconnected components where sharing is not known, the resulting phasing should try to minimize the difference of cardinality in the overall phasing.

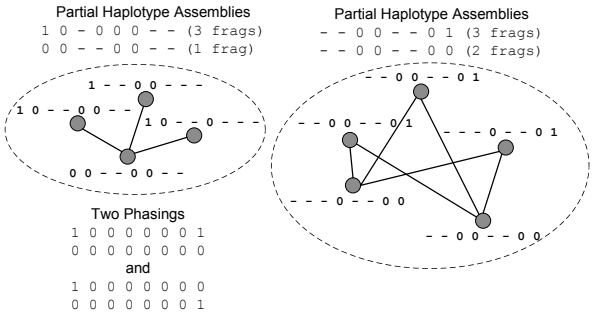

Fig. 2. Fragment conflict graph separated by a run of homozygosity. We assume the maximum distance between fragments is 2 SNPs.

4. Results on Simulated Data

We ran simulations on individuals of the CEU and JPT populations from HapMap[c]. First we sampled individuals randomly and then isolated a subset of the haplotype (30 SNPs for

[c]CEU denotes Utah residents with Northern and Western European ancestry and JPT denotes Japanese individuals from Tokyo, Japan.

visualization purposes). We placed the SNPs from the phased HapMap haplotypes a uniform distance from each other (500bp). Genome length is calculated by *number of SNPs* $\times$ *distance between SNPs*. The distance between sequence reads is calculated using a Poisson distribution and is varied under different models because most NGS technologies are capable of varying the distances between reads (e.g. Solexa or SOLiD). The average read length and coverage are also varied. Figures 3 and 4 show simulations on two unrelated individuals (one from CEU colored green, one from JPT colored red) while Fig. 5 shows simulations from two related individuals.

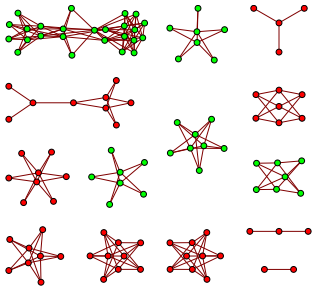

Fig. 3. Fragment conflict graph for unrelated individuals with coverage $c = 4$, read length $L = 50$, and distance between reads is Poisson with $\lambda = 2000$. Green vertices denote fragments originating from the CEU individual.

Figure 3 has many disconnected components due to the low probability of a good-read and regular distance between reads. Figure 4 shows the effect of changing the read length, coverage, and distance between reads. Read length, coverage, and variation of mate pair length correlate strongly with connectivity of the fragment conflict graph. In Fig. 5 two related individuals are shown with the same parameters used in Fig 3. It is clear the more sharing existing in the population, the easier it is to assemble and phase the data.

We also used our haplotype assembly simulator to test the accuracy and scalability of our minimum fragment removal heuristic. The first dataset we tested is the same 30 SNP segment from the HapMap CEU individual; the second dataset is a Hudson simulated chromosome of length 3434 SNPs. We decided to use the ratio of the number of erroneous fragments removed to the number of non-erroneous fragments removed as our metric. After the fragment conflict graph is generated, it may be advantageous to remove non-erroneous fragments to minimize our objective function. Nevertheless, this ratio is a good indicator of the quality of the output. For 1000 runs of the 30 SNP dataset we observed an overall ratio of 6.73; and for 100 runs of the 3434 SNP dataset we observed a ratio of 5.72. Further improvements to this type of algorithmic strategy for this problem is the subject of future work.

We've presented statistical estimates of coverage needed to cover a percentage of SNPs on a genome. These estimates could provide valuable insight when deciding sequence coverage per individual in association studies employing NGS technology. We've suggested a practical algorithmic strategy that exploits the high coverage possible with next-generation sequencing technology and the structure of errors in the fragment conflict graph. This algorithm produces promising results on the simulated fragment conflict graphs. We have presented an algorithm

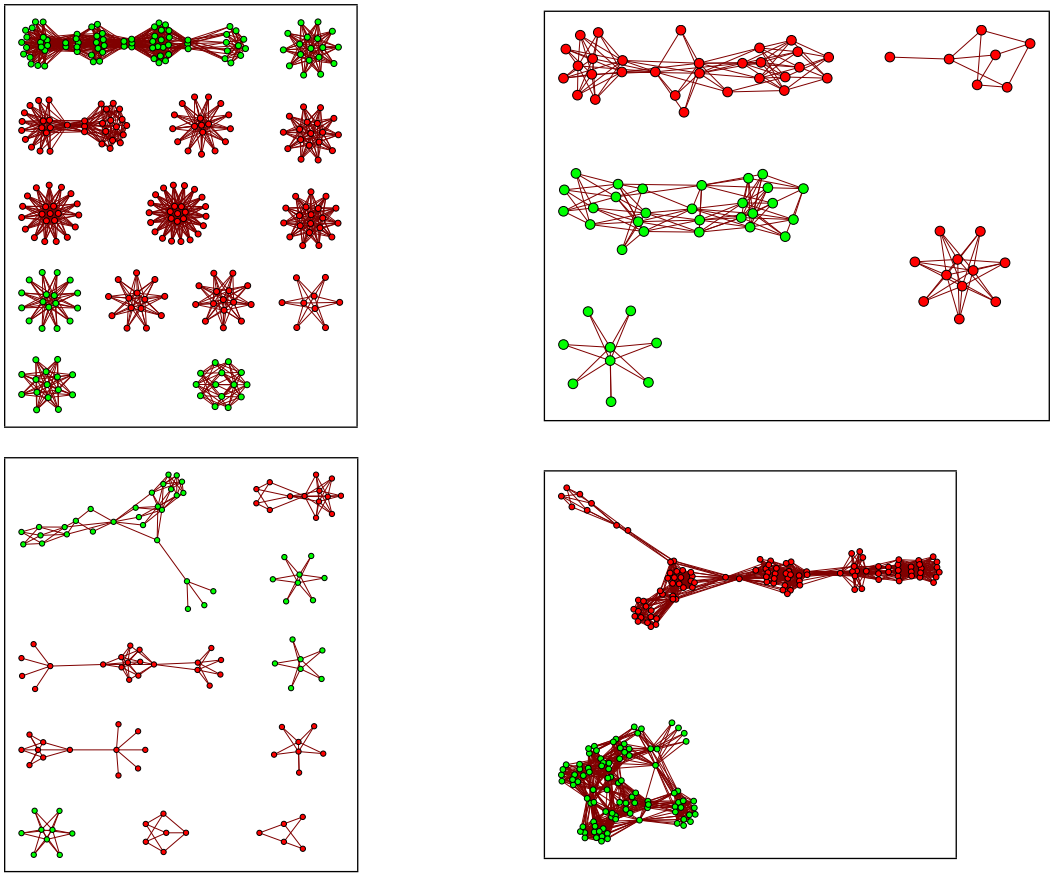

Fig. 4. Fragment conflict graph for two unrelated individuals. Green vertices denote fragments originating from the CEU individual. The baseline for each graph is: Read length $L = 50$; Coverage $c = 4$; distance between reads is Poisson with $\lambda = 2000$. From bottom left clockwise: (1) Distance between reads is Poisson with $\lambda = \{1000, 2000, 5000, 10000\}$ which is selected uniformly at random. (2) Coverage is changed to $c = 10$. (3) Read length is changed to $L = 1000$. (4) Coverage is $c = 10$, read length is $L = 1000$, and distance between reads is varied from $\{1000, 2000, 5000, 10000\}$.

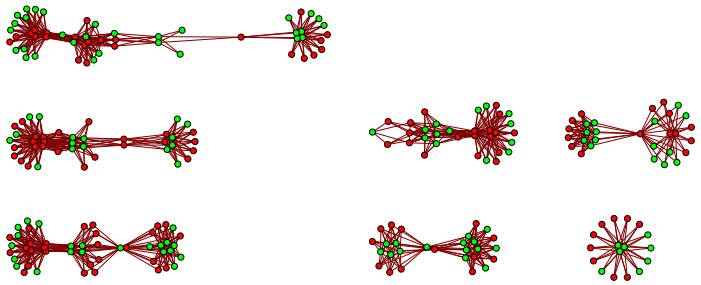

Fig. 5. Fragment conflict graph for two individuals sharing one haplotype. Green and red vertices denote fragments originating from different individuals. Read length $L = 50$. Coverage $c = 4$. The distance between reads is Poisson with $\lambda = 2000$.

for finding and exploiting haplotype sharing in the fragment conflict graph to enable the reliable phasing of disconnected components. We've also shown through simulation how various genomic and experimental parameters impact the quality of the haplotype assembly.

5. Acknowledgments

We thank the National Science Foundation for their support of this research.

References

1. B. V. Halldórsson, D. Aguiar, R. Tarpine and S. Istrail, *International Conference on Research in Computational Molecular Biology (RECOMB)* **6044**, 158 (2010).
2. B. V. Halldórsson *et al.*, *Lecture Notes in Computer Science* (2004).
3. R. Sharan, B. V. Halldórsson and S. Istrail, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **3**, 303 (2006).
4. R. Lippert, R. Schwartz, G. Lancia and S. Istrail, *Brief Bioinform* **3**, 23 (March 2002).
5. G. Lancia, V. Bafna, S. Istrail, R. Lippert and R. Schwartz, *Proceedings of the 3rd European Symposium on Algorithms, (EAS01) Springer Lecture Notes in Computer Science* **2161**, 182 (2001).
6. S. Istrail *et al.*, *Proceedings of the National Academy of Sciences of the United States of America* **101**, 1916 (2004).
7. J. C. Venter, *Nature* **464**, 676 (April 2010).
8. R. Sladek *et al.*, *Nature* **445**, 881 (February 2007).
9. *Nature* **447**, 661 (June 2007).
10. R. J. Klein *et al.*, *Science* **308**, 385 (April 2005).
11. D. A. Hafler *et al.*, *N Engl J Med* **357**, 851 (2007).
12. N. J. Samani *et al.*, *N Engl J Med* **357**, 443 (August 2007).
13. M. N. Weedon *et al.*, *Nature Genetics* **40**, 575 (April 2008).
14. E. E. Eichler *et al.*, *Nature reviews. Genetics* **11**, 446 (June 2010).
15. J. McClellan and M.-C. King, *Cell* **141**, 210 (April 2010).
16. T. Walsh and M.-C. King, *Cancer Cell* **11**, 103 (February 2007).
17. J. C. Cohen *et al.*, *Proceedings of the National Academy of Sciences of the United States of America* **103**, 1810 (February 2006).
18. A. A. Dror and K. B. Avraham, *Annual Review of Genetics* **43**, 411 (2009).
19. H. Stefansson, D. Rujescu, S. Cichon, O. P. H. Pietilainen *et al.*, *Nature* **455**, 232 (Sep 2008).
20. J. Sebat *et al.*, *Science* **316**, 445 (April 2007).
21. N. Siva, *Nature biotechnology* **26**, p. 256 (March 2008).
22. Wellcome trust launches effort to sequence 10,000 human genomes `http://www.genomeweb.com//node/943774?hq_e=el&hq_m=751896&hq_l=1&hq_v=672d790b6d`.
23. R. Schwartz, *Communications in Information and Systems* **10**, 23 (2010).
24. E. S. Lander and M. S. Waterman, *Genomics* **2**, 231 (April 1988).
25. V. Bansal, A. L. Halpern, N. Axelrod and V. Bafna, *Genome Research* **18**, 1336 (August 2008).
26. D. He *et al.*, *Bioinformatics* **26**, i183 (June 2010).
27. V. Bansal and V. Bafna, *Bioinformatics* **24**, i153 (August 2008).
28. Y. Wang *et al.*, *Computational Biology and Chemistry* **31**, 288 (August 2007).
29. J. Wang, M. Xie and J. Chen, *Algorithmica* **56**, 283 (March 2010).
30. S. Kang, I. Jeong, M. Choi and H. Lim, *Lecture Notes in Computer Science* **5059/2008**, 45 (2008).
31. S. Purcell *et al.*, *American journal of human genetics* **81**, 559 (September 2007).
32. S. R. Browning and B. L. Browning, *American journal of human genetics* **86**, 526 (April 2010).
33. Z. Li, L. Wu, Y. Zhao and X. Zhang, *Acta Mathematicae Applicatae Sinica* **22**, 405 (2006).
34. R. R. Hudson, *Bioinformatics* **18**, 337 (2002).
35. A. Kong, G. Masson, M. L. Frigge *et al.*, *Nat Genet* **40**, 1068 (Sep 2008).
36. M. J. Minichiello and R. Durbin, *Am J Hum Genet.* **79**, 910 (Nov 2006).