Highly Scalable Genotype Phasing by Entropy Minimization

Alexander Gusev, Ion Măndoiu, and Bogdan Paşaniuc

***Abstract*— A *Single Nucleotide Polymorphism (SNP)* is a position in the genome at which two or more of the possible four nucleotides occur in a large percentage of the population. SNPs account for most of the genetic variability between individuals, and mapping SNPs in the human population has become the next high-priority in genomics after the completion of the Human Genome project. In diploid organisms such as humans, there are two non-identical copies of each autosomal chromosome. A description of the SNPs in a chromosome is called a haplotype. At present, it is prohibitively expensive to directly determine the haplotypes of an individual, but it is possible to obtain rather easily the conflated SNP information in the so called genotype. Computational methods for genotype phasing, i.e., inferring haplotypes from genotype data, have received much attention in recent years as haplotype information leads to increased statistical power of disease association tests. However, many of the existing algorithms have impractical running time for phasing large genotype datasets such as those generated by the international HapMap project. In this paper we propose a highly scalable algorithm based on entropy minimization. Our algorithm is capable of phasing both unrelated and related genotypes coming from complex pedigrees. Experimental results on both real and simulated datasets show that our algorithm achieves a phasing accuracy worse but close to that of best existing methods while being several orders of magnitude faster. The open source code implementation of the algorithm and a web interface are publicly available at `http://dna.engr.uconn.edu/~software/ent/`.**

***Index Terms*— Single Nucleotide Polymorphism, haplotype, genotype phasing, algorithm.**

I. Introduction

After the completion of the Human Genome Project has provided us with a blueprint of the DNA present in each human cell, genomics research is now focusing on the study of DNA variations that occur between individuals and understanding how these variations confer susceptibility to common diseases such as diabetes or cancer. The most common form of genomic variation are the so called *single nucleotide polymorphisms* (SNPs), i.e., the presence of different DNA nucleotides, or *alleles*, at certain chromosomal locations. Close to 12 million common SNPs have been catalogued in the most recent build (126) of the dbSNP database maintained by NCBI (`http://www.ncbi.nlm.nih.gov/projects/SNP/`).

In diploid organisms such as humans, there are two non-identical copies of each autosomal chromosome, one inherited from the mother and one inherited from the father. The combinations of SNP alleles in the maternal and paternal chromosomes are referred to as the individual's *haplotypes*. Although it is possible to directly determine the haplotypes of an individual by experimental techniques, such methods are prohibitively expensive and time consuming. In contrast, there are many cost-effective high-throughput techniques for determining the conflated SNP information called *genotype*, which specifies the identities of the two alleles at each SNP position, but does not assign the alleles to specific chromosomes for *heterozygous* SNP positions, i.e., SNP positions at which the individual has two different alleles.

Since haplotypes determine the exact sequence (and hence function) of proteins encoded by the genes, finding the haplotypes in human populations is an important step in determining the genetic basis of complex diseases. For this reason, computational inference of haplotypes from genotype data, known as the *genotype phasing problem*, has received much attention in the past few years, see, e.g., [2]–[5] for recent surveys.

Renewed interest in phasing algorithms is currently driven by the need to handle increasingly larger datasets. High-end genotyping platforms from Affymetrix an Illumina already allow typing over half a million SNP genotypes per experiment, with one million SNP genotypes per experiment expected in the very near future. Furthermore, due to decreasing genotyping costs, future association studies are expected to comprise thousands of typed individuals [6]. While many of the existing methods achieve high haplotype reconstruction accuracy, their runtimes do not scale well with the number of SNPs and the number of typed individuals. In particular, all commonly used phasing methods are vastly inadequate for handling datasets of the size envisioned to be produced by next generation of genome-wide association studies.

In this paper we propose a highly scalable algorithm based on the entropy minimization principle that was previously proposed in the context of genotype phasing and haplotype missing data recovery by Halperin and Karp [7]. As shown in Section II, entropy minimization can be viewed as the maximization of phasing likelihood under a simple count-based estimate of haplotype frequencies. Unlike the simple greedy algorithm employed in [7], we use a local optimization algorithm, which in practice results in genotype phasings with lower entropy. For phasing long genotypes, the local optimization algorithm is extended using a novel overlapping-window approach. Combined with a simple batched implementation, this results in a runtime that grows linearly with the number of SNPs, and nearly linearly with the number of typed individuals. Comprehensive experiments on both real and simulated datasets show that the

This work was supported in part by NSF CAREER award IIS-0546457 and NSF award DBI-0543365. A preliminary version of this work has appeared in [1].

Authors' address: University of Connecticut, Computer Science & Engineering Department, 371 Fairfield Rd., Storrs, CT 06269-2155, E-mail: `{sasha.gusev,ion.mandoiu,bogdan.pasaniuc}@uconn.edu`.

entropy minimization algorithm is orders of magnitude faster than existing phasing methods, while still maintaining a phasing accuracy close to that of the best existing methods.

We also describe in the paper the extension of our entropy minimization algorithm to genotype data coming from complex pedigrees. As genotyping costs decrease, association studies are likely to increasingly rely on analyses of genotype data from *related* individuals. Indeed, parent-child relationships can be exploited to reliably infer haplotype phase for a substantial fraction of the SNPs based on the no-recombination assumption [6]. Consider for example a nuclear family (trio) composed of two parents and a child. Under the no-recombination assumption each parent passes an entire chromosome to the child. That is, the child shares one haplotype with the mother and the other one with the father. The only situation when there is phasing ambiguity for a given SNP is when all three genotypes are either heterozygous or missing at that SNP. For example, in the CEU and YRI trio datasets of HapMap Phase I [8], the phase of only around 15% of the SNPs is ambiguous, while the phase of the remaining 85% of the SNPs can be inferred based on the no-recombination assumption. We believe that the ability to exploit the entire available pedigree information gives a distinct advantage to our algorithm. Simulation experiments reported in Section IV-D show that incorporating increasing amounts of pedigree information improves not only the absolute accuracy of the entropy minimization algorithm (which is to be expected since the number of ambiguous sites is decreasing), but also its relative accuracy (meaning that a smaller percentage of ambiguous sites is incorrectly resolved). In fact, the results show that, for complex pedigrees the accuracy of our algorithm may exceed that of much slower methods which cannot take into account the full pedigree information.

The rest of the paper is organized as follows. In Section II we introduce some basic terminology and formalize the minimum entropy phasing problem. In Section III we describe the local improvement algorithm for entropy minimization and its extensions to long genotypes and complex pedigrees. Finally, we present experimental results in Section IV and conclude in Section V.

II. Problem Formulation

Following the standard practice, in this paper we restrict our attention to bi-allelic SNPs, which form the vast majority of known SNPs. In this case a haplotype can be represented as a 0/1 vector – typically by representing the most frequent allele as a 0 and the alternate allele as a 1. A genotype will be represented as a 0/1/2/? vector, where 0 (1) means that both chromosomes contain the 0 (1) allele, 2 means that the two chromosomes contain different alleles, and "?" means that allele identities are unknown. The allele at locus i of haplotype h is denoted by $h(i)$. Similarly, for a given genotype vector g, the genotype at locus i is denoted by $g(i)$.

We say that haplotype h is *compatible* with genotype g if $g(i) = h(i)$ whenever $g(i) \in \{0, 1\}$. A pair of haplotypes (h_1, h_2) *explains* genotype g if $h_1(i) = h_2(i) = g(i)$ whenever $g(i) \in \{0, 1\}$, and $h_1(i) \neq h_2(i)$ whenever $g(i) = 2$. For a given pair (h_1, h_2) that explains g we say that h_1 and h_2 are complementing each other with respect to g.

We call a set of genotypes *unrelated* if there are no parent-child relationship between the individuals from which the genotypes were obtained. We next formalize the minimum entropy phasing problem for unrelated genotypes; phasing of related genotypes is discussed in Section III-D.

A *phasing* of a set of unrelated genotypes G, each of length k, is a function $\phi : G \to \{0,1\}^k \times \{0,1\}^k$, such that, for every $g \in G$, $\phi(g)$ is a pair of haplotypes that explain g. For a haplotype h and a phasing ϕ, the *coverage of* h *under* ϕ, denoted by $cvg(h, \phi)$, is the number of genotypes $g \in G$ such that $\phi(g) = (h, h')$ or $\phi(g) = (h', h)$ with $h' \neq h$, plus twice the number of genotypes $g \in G$ such that $\phi(g) = (h, h)$. Notice that, for a fixed phasing, the sum of all haplotype coverages is equal to $2|G|$. As in [7], [9], we define the *entropy* of a phasing ϕ as

$$\mathcal{H}(\phi) = \sum_{h: cvg(h,\phi) \neq 0} -\frac{cvg(h,\phi)}{2|G|} \log \frac{cvg(h,\phi)}{2|G|} \tag{1}$$

Halperin and Karp [7] introduced the following

Minimum entropy phasing problem: Given a set G of unrelated genotypes, find a phasing ϕ of G with minimum entropy.

The use of entropy minimization in genotype phasing can be motivated by the following connection with likelihood maximization. For given haplotype probabilities p_h, the log-likelihood of a phasing ϕ is

$$\begin{aligned} L(\phi) &= \log \left(\prod_h p_h^{cvg(h,\phi)} \right) \\ &= \sum_h cvg(h,\phi) \log p_h \\ &= -2|G| \sum_{h: cvg(h,\phi) \neq 0} -\frac{cvg(h,\phi)}{2|G|} \log p_h \end{aligned}$$

If p_h is estimated by simply counting the number of times h appears in ϕ, i.e., $p_h = \frac{cvg(h,\phi)}{2|G|}$, it can be easily seen that maximizing the log-likelihood $L(\phi)$ is equivalent with minimizing $\mathcal{H}(\phi)$.

III. Algorithm

Halperin and Karp [7] proposed a greedy algorithm for the related *minimum-entropy set cover problem*, and showed that a variant of this algorithm can be applied to unrelated genotype phasing. However, the greedy algorithm cannot be applied directly to phasing long genotypes, i.e., genotypes with large numbers of SNPs. As the number of SNPs increases, each haplotype becomes compatible with at most one genotype, and thus all phasings result in the same entropy of $-\log \frac{1}{2|G|}$, rendering the entropy minimization objective useless. Furthermore, even for short genotypes, the entropy of phasings produced by the greedy algorithm in [7] can be significantly improved. Indeed, although greedy phasings are guaranteed to have an entropy at most 3 bits larger than the optimum entropy, the optimum entropy for short genotypes is typically very small. In this paper we use the entropy minimization objective within a local improvement framework. In Section III-A we describe the local improvement algorithm for phasing short genotypes

```
Input: Set $G$ of genotypes
Output: Phasing $\phi$ of the genotypes in $G$

1.  Generate a random phasing $\phi$ for genotypes in $G$
2.  Repeat forever
      2.1 Find the pair $(g, (h'_1, h'_2))$ such that $\mathcal{H}(\phi')$ is
          minimized, where $\phi'$ is obtained from $\phi$ by
          re-explaining $g$ with $(h'_1, h'_2)$
      2.2 If $\mathcal{H}(\phi') < \mathcal{H}(\phi)$, then $\phi \leftarrow \phi'$
          Else exit the repeat loop
3.  Output $\phi$
```

Fig. 1. ENT phasing of short genotypes.

```
Input: Set $G$ of genotypes
Output: Phasing $\phi$ of the genotypes in $G$

1.  Divide the genotypes in groups of $f$ consecutive SNPs
    from left to right
2.  For each group, add the preceding $l$ SNPs to create a
    window of size $l + f$ SNPs (leftmost window has no
    locked SNPs and is of size $f$)
3.  Run the phasing algorithm in Figure 1 for each
    window, in left to right order, where the haplotypes
    over the locked $l$ SNPs are not allowed to change
4.  Output the resulting phasing $\phi$
```

Fig. 2. ENT phasing of long genotypes.

of unrelated individuals. Then, in Sections III-B and III-C we describe the extension to phasing of long unrelated genotypes and discuss the time complexity of the algorithm. Finally, in Section III-D we describe the extension of the local improvement algorithm to the problem of phasing long genotypes of related individuals.

A. Short genotype phasing

We have implemented a simple local improvement algorithm for entropy minimization. Our algorithm, which we refer to as ENT, starts from a random phasing, then, at each step, finds the genotype whose re-explanation yields the largest decrease in phasing entropy (see Figure 1). The use of random initial phasings is justified by observing that a random phasing of a genotype with i heterozygous positions matches the real phasing with probability 2^{-i}. E.g., when phasing the children genotypes from the well-known Daly dataset [10], random phasing results in an average of 46% correct haplotypes over windows of 5 consecutive SNPs. We have also experimented with a version of the algorithm in which the initial phasing is obtained by running the greedy algorithm of [7], which repeatedly chooses the haplotype h that explains the maximum number of unexplained genotypes. Preliminary experiments on simulated data [11] have shown that the use of random initial phasings yields convergence to final phasings with same or slightly lower entropy. This suggests that starting from the greedy initial solution traps the local optimization algorithm into a poorer local optimum.

We experimented with two tie-breaking rules in step 2.1 of the algorithm: either picking the first, or a random pair among pairs $(g, (h_1, h_2))$ that yield minimum $\mathcal{H}(\phi')$. Our experiments showed that both approaches yield phasings with similar entropy and accuracy. Also, the runtime of our algorithm was not influenced by the tie-breaking rule. In all experiments reported in this paper we used the first pair whenever we had to break a tie.

B. Long genotype phasing

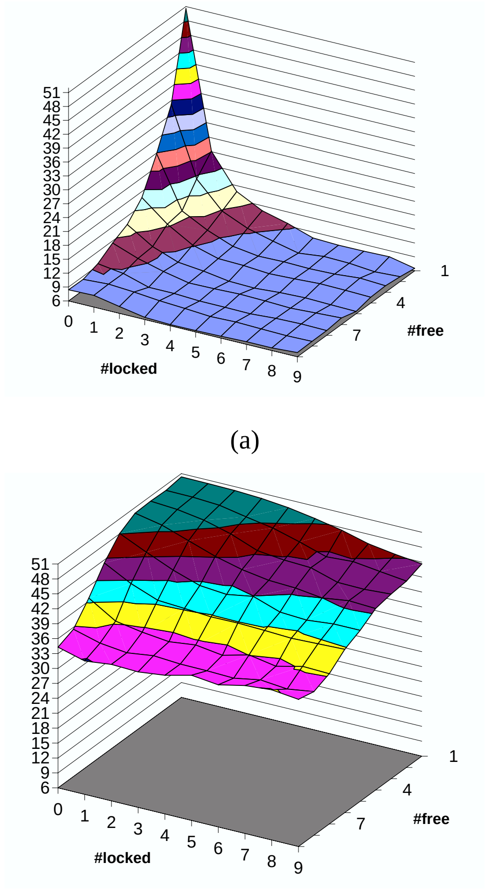
A common approach to phasing long genotypes is to phase short *non-overlapping windows* of the input genotypes and then stitch the resulting haplotypes using various statistical approaches, see, e.g., [6], [12]. Recently, [13] proposed a method that considers phasings over *all* possible short windows in conjunction with a dynamic programming algorithm that finds a global phasing that minimizes the number of disagreements with the local predictions.

We also adopt a window-based approach to phasing long genotypes. Like [13], our algorithm employs a set of short *overlapping windows*. However, instead of using all short windows as in [13], we use a much smaller set of overlapping windows of fixed size. Specifically, each window consists of a set of l "locked" SNPs, which have been previously phased, and a set of f "free" SNPs, which are currently being phased. For each window, the phasing algorithm proceeds as described in the previous section, except that only re-explanations consistent with the already determined haplotypes of the locked SNPs are considered in the local improvement step (see Figure 2).

The basic implementation of the ENT algorithm takes l and f as input parameters. We have also implemented variants of the algorithm that dynamically compute the number of locked, respectively free SNPs based on the input data. These variants pick l and f as large as possible subject to the constraint that the numbers of ambiguous (heterozygous or missing) SNP genotypes in the locked, respectively free region of the current window do not exceed twice the number of genotypes. The number of free SNPs f is further constrained to disallow having more than 7 ambiguous SNPs in the free region of any genotype.

To assess the effect of the windowing strategy (number of free and locked SNPs) on phasing accuracy, we conducted a set of experiments on a well-known dataset from Daly et al. [10]. This dataset contains 129 trios from a European population. Each individual was typed at 103 SNP loci in the 5q31 region of chromosome 5. The trio genotypes were used to infer as much as possible out of the "true" haplotypes of the children under the no-recombination assumption. We used the genotypes of the children as input to ENT and compared the obtained phase with the partially recovered "true" haplotypes.

Figure 3(a) shows the *Relative Switching Error* (RSE) (see Section IV-A for the definition) obtained by running ENT with the number of locked SNPs varied between 0 and 9, and the number of free SNPs varied between 1 and 9. As expected, the RSE is 50% for $l = 0$ and $f = 1$, since for this setting of parameters ENT simply produces a random phasing. As the numbers of free and locked SNPs are increased, the entropy minimization objective quickly becomes informative, and the RSE decreases significantly, with best results (RSE of 6.18%)



(a)

(b)

Fig. 3. Relative switching errors obtained on the Daly children dataset by running the local improvement algorithm with overlapping-windows with 0-9 locked SNPs and 1-9 free SNPs and two optimization objectives: (a) minimizing phasing entropy, (b) minimizing the number of distinct haplotypes.

being obtained for $l = f = 5$ (the RSE is changing very little – within a 1% range – when setting f and l to higher values). For this dataset, the version that dynamically chooses both l and f yields minimal RSE as well. Experiments performed on other datasets confirmed that automatically chosen f and l parameters consistently yield phasings with RSE close to that of the best variant. Therefore, we use this variant in the experiments presented in following sections.

To better understand the significance of using entropy minimization as optimization objective for phasing short windows, we compared it with the objective of minimizing the number of distinct haplotypes used in the phasing. This so called *pure parsimony* objective was introduced in [14], which also proposes an exponential-size integer linear program formulation. A more scalable branch-and-bound algorithm for pure parsimony was given in [15], and polynomial-size integer linear programs were independently proposed in [16], [17]. Figure 3(b) shows that, for the considered window sizes, the RSE obtained with the pure parsimony objective is much worse than that obtained with entropy minimization.

C. Time complexity

When phasing n unrelated genotypes over k SNPs, the algorithm in Figure 1 is run on $\lceil k/f \rceil$ windows. For each window, the algorithm evaluates at most $n \times 2^f$ candidate pairs of haplotypes for finding the best pair in Step 2.1. Computing the entropy gain for each candidate pair takes constant time. Indeed, $\mathcal{H}(\phi')$ differs from $\mathcal{H}(\phi)$ in at most four terms corresponding to the haplotypes that can change their coverages, namely the haplotypes explaining g in ϕ and ϕ'. Empirically, the number of iterations required in Step 2 of the algorithm in Figure 1 is linear in the number n of genotypes (see Figure 4), resulting in an overall runtime of $O(n^2 2^f k/f)$.

To reduce the number of iterations, we implemented a *batched* version of the algorithm in which multiple genotypes are re-explained in each iteration. In this version of the algorithm, an iteration starts by computing *for each genotype* g a pair $(g, (h'_1, h'_2))$ of compatible haplotypes that yield the highest entropy gain. The resulting list of n such pairs is then traversed in order of decreasing gain. For each pair $(g, (h'_1, h'_2))$, the genotype g is re-phased using (h'_1, h'_2) if the entropy gain is still positive with respect to the current phasing. Empirically, the number of iterations required by the batched variant is $O(\log^3 n)$, resulting in an overall runtime of $O(n \log^3 n 2^f k/f)$.

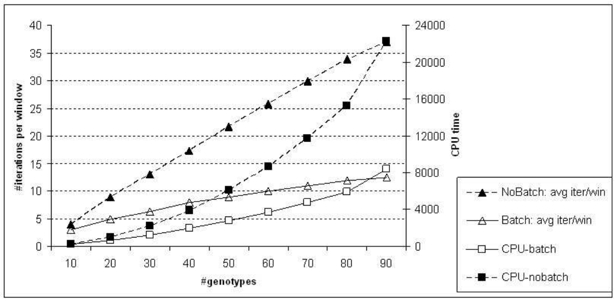


Fig. 4. Total CPU runtime and average number of iterations per window for the ENT algorithm with and without batching ran on the JPT+CHB HapMap Phase II dataset.

Figure 4 gives experimental results comparing the ENT algorithm with and without batching on the JPT+CHB dataset of HapMap Phase II [8], consisting of 90 unrelated individual genotypes with a total of over 3.7 million SNPs (all 22 autosomal chromosomes, see Section IV for more details on this dataset). The two versions of the algorithm give very similar phasing accuracy, with the batched variant being up to 2.5 times faster. As shown in the figure, the speed-up comes from the reduction in number of iterations required by the batched version. All remaining experiments use the batched version of the algorithm.

D. Phasing related genotypes

We have also extended the ENT algorithm to handle datasets consisting of related genotypes grouped into pedigrees. The algorithm for phasing a short window of related genotypes is similar to the one in Figure 1. For every window we restrict the search to phasings that satisfy the no-recombination assumption. To maintain this property throughout the algorithm, in each local improvement step we re-explain all genotypes in a pedigree rather than a single genotype.

If entropy is computed based on haplotype counts of *all* typed individuals, when re-phasing a pedigree the algorithm may introduce significant biases in haplotype transmission rates. One way to avoid this problem is to compute the entropy over an *independent* set of haplotypes, such as the "founder" haplotypes, i.e., haplotypes inherited from individuals not included in the pedigree. For example, in the case of a trio, computing the entropy over all haplotypes uses six haplotypes, while computing it over the founder haplotypes uses only the four haplotypes of the parents. We implemented both entropy computation methods, and compared their accuracy on CEU and YRI trio datasets from HapMap Phase I. As shown in Table I, for almost all chromosomes, computing the entropy over founder haplotypes yields slightly better accuracy. Therefore, in all remaining trio experiments we use the founder-only entropy calculation.

Chr#	CEU			YRI		
	ALL	Found.	Decrease(%)	ALL	Found.	Decrease(%)
1	1.42	1.35	4.93	2.42	2.27	6.20
2	1.09	1.07	1.83	1.50	1.42	5.33
3	1.11	1.10	0.90	1.59	1.50	5.66
4	1.24	1.21	2.42	1.81	1.76	2.76
5	1.14	1.11	2.63	1.62	1.54	4.94
6	1.12	1.07	4.46	1.58	1.54	2.53
7	1.38	1.36	1.45	2.09	1.99	4.78
8	0.85	0.83	2.35	1.21	1.13	6.61
9	1.02	0.98	3.92	1.36	1.33	2.21
10	1.34	1.30	2.99	1.86	1.81	2.69
11	1.27	1.21	4.72	1.68	1.52	9.52
12	1.34	1.32	1.49	2.02	1.98	1.98
13	1.34	1.26	5.97	1.77	1.66	6.21
14	1.34	1.35	-0.75	1.81	1.66	8.29
15	1.42	1.40	1.41	2.01	2.00	0.50
16	1.68	1.63	2.98	2.48	2.39	3.63
17	1.59	1.53	3.77	2.38	2.30	3.36
18	1.08	1.04	3.70	1.48	1.43	3.38
19	1.99	1.89	5.03	2.71	2.65	2.21
20	1.78	1.67	6.18	3.68	3.59	2.45
21	1.14	1.14	0.00	1.69	1.55	8.28
22	1.22	1.23	-0.82	1.74	1.70	2.30
Avg.	**1.31**	**1.28**	**2.80**	**1.93**	**1.85**	**4.36**

TABLE I

COMPARISON BETWEEN "ALL" AND "FOUNDERS-ONLY" HAPLOTYPE COUNTING STRATEGIES ON HAPMAP PHASE I TRIO POPULATIONS.

```
Input: Mendelian consistent genotype data for a pedigree $P$
together with haplotype inheritance pattern
Output: List $\mathcal{L}$ of feasible phasings of $P$

Let $g_1, \ldots, g_{|P|}$ be the genotypes of $P$ indexed in
  reverse topological order
$\mathcal{L} \leftarrow \emptyset$; $i \leftarrow 1$; $\mathcal{L}_k \leftarrow \emptyset$ for $k = 1, \ldots, |P|$
While $i > 0$ do
    If $\mathcal{L}_i = \emptyset$ then
        If $g_i$ has descendants and their haplotypes are
          incompatible under the given inheritance
          pattern then
            $i \leftarrow i - 1$
        Else
            Set $\mathcal{L}_i$ to the list of phasings of $g_i$
              compatible with existing descendents (if
              any)
            $j_i \leftarrow 1$; $i \leftarrow i + 1$
    Else    // $\mathcal{L}_i \neq \emptyset$
        If $j_i > |\mathcal{L}_i|$ then
            $\mathcal{L}_i \leftarrow \emptyset$; $i \leftarrow i - 1$
        Else
            If $i = |P|$ then
                Add to $\mathcal{L}$ the phasing in which each
                  genotype $g_k$ is explained using $\mathcal{L}_k(j_k)$
                $j_i \leftarrow j_i + 1$
            Else
                $j_i \leftarrow j_i + 1$; $i \leftarrow i + 1$
Output $\mathcal{L}$
```

Fig. 5. Bottom-up enumeration of feasible phasings for short related genotypes.

An implicit representation of zero-recombination phasings for a fixed window can be found in $O(mn^2 + n^3 \log^2 n \log\log n)$ time using a system of linear equations and an efficient method for eliminating redundant equations [18]. However, since the number zero-recombination phasings can be exponential, we chose to generate these phasings iteratively using a backtracking strategy. Each pedigree is represented as a directed acyclic graph with nodes representing genotypes and directed edges connecting parents to children. Nodes that have no incoming edges will be referred to as founder nodes. Two variants of backtracking were implemented. In the *top-down* variant we generate the phasing for a pedigree starting from the founder nodes and then following a topological order. This assures that, when visiting a node, its parents are already visited. At each node, we only generate phasing compatible with the existing parent haplotypes. Once the last node in a pedigree is phased, we compute the entropy gain and backtrack to previous nodes to explore other feasible phasings. The *bottom-up* variant (Figure 5) iterates through feasible phasing in a similar manner, but starts the traversal from the nodes that have no outgoing edges, corresponding to individuals that have no children, and works its way up towards the founder nodes.

To speed-up the enumeration of feasible phasings, for each node in the pedigree graph we generate two templates representing the maternal and paternal haplotypes. These templates are incomplete haplotypes, containing only the alleles that can be unambiguously inferred from the genotype data (possible Mendelian inconsistencies are detected and reported when constructing these templates). Furthermore, after phasing the first window, we determine the grand-parental status of the two haplotypes of each non-founder node, and allow in subsequent windows only phasings consistent with this haplotype inheritance pattern. If the algorithm encounters a window for which a phasing consistent with this pattern cannot be found (either due to the presence of a recombination event or poor initial choice of haplotype inheritance pattern) we repeatedly decrease the number of free SNPs by one unit until a feasible phasing can be found. The algorithm is then restarted with no locked SNPs and the computed phasing is used to infer a new haplotype inheritance pattern.

Enumerating all feasible phasings of a pedigree P for a fixed window with f free SNPs requires $O(2^{f|P|})$ time in the worst case for both backtracking variants. This bound is achieved when all SNP genotypes are missing, and cannot be improved since there are $O(2^{f|P|})$ feasible phasings in this case. However, on typical data the number of feasible phasings and the runtime are much lower than suggested by the worst case bound. Despite having the same worst case runtime, the bottom-up implementation was empirically found to be faster than the top-down variant. We compared the two variants on datasets containing between 6 to 60 trios from the combined CEU and YRI HapMap Phase II consensus datasets. These datasets contain approximately 3.5 million SNPs that are present in both CEU and YRI populations. Genotypes for these SNPs were created by combining the reference phasing given on the HapMap website, and therefore contain no missing data. The runtimes for the top-down and bottom-up versions of the ENT algorithm are summarized in Figure 6. While both runtimes increase nearly linearly with the number of trios, the bottom-up variant is over 10 times faster for each instance size tested. Since the two variants yield phasings with similar accuracy, all remaining experiments use the bottom-up variant of the algorithm.

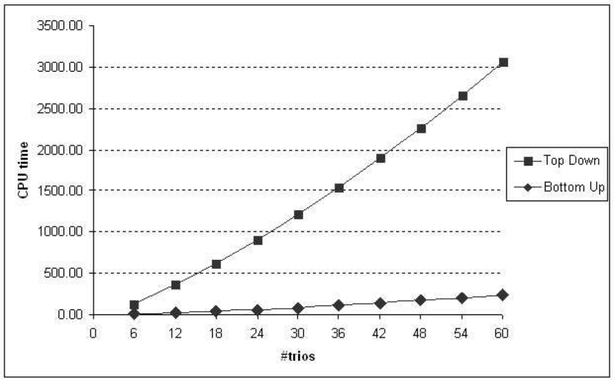


Fig. 6. Runtime of bottom-up and top-down ENT variants on 6-60 trios from the combined CEU+YRI HapMap Phase II consensus datasets.

IV. Experimental Results

A. Experimental Setup

The ENT algorithm was implemented as described in previous section using the C++ language. The experiments presented in this paper were conducted on a 2.8GHz Pentium Xeon machine with 4Gb of memory running the Linux operating system.

For our experiments we used several datasets:

- *HapMap Phase I datasets.* HapMap [8] is a large international project seeking to develop a haplotype map of the human genome. We used two trio panels (CEU and YRI) consisting of 30 trio families each from the HapMap Phase I release 16a. Since the HapMap genotypes for this release were not consistent with the reference haplotypes, we ran the compared methods on genotypes reconstructed from reference haplotypes, which resulted in genotypes with no missing data.
- *HapMap Phase II datasets.* We used all three panels available in HapMap Phase II release 21: the two trio panels (CEU and YRI) and a combined panel consisting of the all 90 individuals from JPT and CHB populations. For these datasets we ran the compared methods on the genotypes available on the HapMap website. Unlike genotypes reconstructed for Phase I datasets, these genotypes contain a small percentage of missing data. Table II shows the number of SNPs, and the percentages of heterozygous and missing SNP genotypes for each of the 22 autosomes in the HapMap Phase II datasets.
- *HapMap-based synthetic datasets.* To allow comparisons of methods that are too slow for handling full chromosome genotype data, Marchini et al. [6] have used the HapMap data to generate a large number of smaller simulated datasets (referred to as “real” in [6]). RT-CEU and RT-YRI trio datasets were obtained by selecting at random 100 1-Mb regions from each one of the HapMap trio populations, CEU and YRI. For each region, 30 new datasets were created by switching the allele transmission status in parent genotypes of one of the trios (thus creating a plausible child genotype, while introducing a minimal amount of missing data). A similar set of 100 datasets of unrelated genotypes (RU) were generated from random 1-Mb regions from the CEU population by simply removing children genotypes.
- *Real dataset from [19].* Datasets for which the haplotypes have been directly determined through molecular techniques such as cloning or strand-specific PCR are the ideal testbed for comparing accuracy of haplotype inference methods. To test if conclusions drawn from synthetic datasets remain applicable to real datasets we used the dataset from [19], consisting of 9 SNPs and 80 phased genotypes collected from unrelated individuals.

Since the true haplotypes are not available for the HapMap datasets, we used as reference the haplotypes inferred by HapMap researchers using the PHASE haplotype inference program [20]. A haplotype inference method can disagree with PHASE reference haplotypes in two ways. For a missing SNP genotype, the alleles inferred by the method can be different from those inferred by PHASE. For non-missing SNP genotypes, the inferred alleles must necessarily agree, but they may be assigned to different haplotypes. We measure the first type of errors using the **Relative Genotype Error (RGE)**, defined as the percentage of missing SNP genotypes that are inferred differently than PHASE. In the case of trio data, a SNP genotype is not considered to be missing if it can be unambiguously inferred from the genotypes of the other members of the trio.

A commonly used measure for the second type of error is the *switching error*, which, for a given genotype, measures the ratio between the number of times we have to switch between the inferred haplotypes to obtain the reference haplotypes. A SNP genotype is called *ambiguous* if its phase cannot be fully inferred from available data. In real data a large fraction of SNP genotypes are non-ambiguous, e.g., homozygous SNPs, or heterozygous SNPs for which other trio members are homozygous. Therefore, in this paper we assess phasing accuracy using

Chr#	CEU			YRI			JPT+CHB		
	#SNPs	%2's	%?'s	#SNPs	%2's	%?'s	#SNPs	%2's	%?'s
1	296976	18.84	1.74	294798	20.79	1.95	300972	17.32	2.04
2	319350	20.74	1.46	311083	23.02	1.69	319895	18.59	1.60
3	249090	21.35	1.90	242356	22.88	1.85	248329	18.85	2.07
4	238489	20.33	1.84	231439	22.46	2.30	237828	18.14	2.32
5	242566	20.90	1.76	236120	22.43	1.88	242834	18.83	1.96
6	262657	20.56	1.71	259628	21.37	1.77	266737	18.72	1.73
7	207892	20.67	1.86	202386	21.95	2.11	207619	18.52	2.07
8	209456	21.48	1.41	207762	23.07	1.52	212608	19.91	1.74
9	177479	20.58	1.50	175609	22.00	1.62	178892	18.89	1.87
10	204417	19.54	1.85	202678	21.40	1.92	206647	17.88	2.08
11	199243	19.40	1.80	193287	20.60	2.16	200395	17.72	2.03
12	187332	19.52	1.99	185132	20.67	2.06	187078	17.76	2.20
13	152612	20.02	1.87	151963	21.86	1.78	154977	18.21	1.97
14	120565	20.54	1.59	117442	22.30	1.75	121046	19.33	1.69
15	104384	20.64	1.76	101443	22.70	1.86	104757	19.45	1.82
16	106411	19.78	1.87	103113	21.87	2.26	106229	18.01	2.18
17	86495	20.20	1.89	83996	21.62	2.04	86199	17.96	2.06
18	116802	19.75	1.46	115056	22.22	1.85	117288	17.94	1.97
19	53738	20.15	1.88	52078	22.13	1.88	53675	18.90	2.09
20	117417	15.75	1.41	114764	17.49	1.52	117155	14.69	1.47
21	48635	21.14	1.70	48770	23.10	1.62	50484	20.10	1.85
22	53463	18.44	1.58	54302	19.71	1.50	55206	16.86	1.71
Total/Avg.	**3755469**	**20.01**	**1.72**	**3685205**	**21.71**	**1.86**	**3776850**	**18.30**	**1.93**

TABLE II
PROPERTIES OF THE HAPMAP PHASE II DATASET.

the **Relative Switching Error (RSE)**, defined as the number of switches needed to convert the inferred haplotype pairs into the reference haplotype pairs, expressed as percentage of the total number of ambiguous SNPs. The positions where the SNP genotypes are missing are ignored in the computation of RSE since errors at these positions are separately accounted for by RGE.

B. Comparison with other methods

The first set of experiments was run on the HapMap Phase II datasets, comprising three panels of 90 individuals each, typed at approximately 3.7 million SNPs (see Table II). On these datasets, we compared ENT with two recent phasing methods, 2SNP and ILP, that are capable of (at least partially) handling such large datasets with reasonable time and memory requirements. 2SNP [21] is a phasing method based on genotype statistics collected for pairs of SNPs. ILP [22] employs a window based approach, for each window minimizing the number of distinct haplotypes used for phasing by using an Integer Linear Programming approach. 2SNP handles unrelated genotypes and trio data, while the ILP method is only able to handle trio data.

Table III gives the accuracy measures and the runtime of ENT, 2SNP and ILP on the two trio populations from HapMap Phase II. ENT has the lowest RGE and RSE error rates. Using PHASE haplotypes as ground truth, ENT accurately recovers, on the average, more than 94% of the missing SNP genotypes for the CEU population and more than 90% for the YRI population. On the average the RSE of ENT is 1.51% for the CEU population and 1.94% for the YRI population, compared to over 20% RSE for 2SNP and over 6% RSE for ILP. ENT is orders of magnitude faster than the other two methods, requiring about half an hour for phasing the two trio datasets, compared to over 20 hours for 2SNP and over 16 days for ILP.[1]

Table IV gives the accuracy measures and the runtime of ENT and 2SNP on the unrelated population (JPT+CHB) from HapMap Phase II. The missing entries in the table are due to the fact that the 2SNP method was unable to complete the phasing of larger chromosomes due to memory constraints. In the case of unrelated genotypes, ENT retains the speed advantage over 2SNP, but yields phasings with slightly lower accuracy.

Similar results were obtained on the HapMap-based synthetic datasets from [6]. Table V gives phasing accuracy results on these datasets for ENT and the widely-used phasing programs PHASE [20], [23], [24], fastPHASE [25], HAP [26], and HAP2 [27]. These methods are based on a variety of statistical and combinatorial techniques, including Bayesian inference, Expectation Maximization, Hidden Markov Models, Markov-Chain Monte Carlo, and perfect phylogeny. (For a description of how the original methods were extended to handle trio data see [6]). The accuracy on these datasets was measured using three criteria introduced in [6]: switching error, incorrect genotype percentage, and incorrect haplotype percentage. The first measure is similar to RSE, except that it is computed only for SNP loci for which real haplotypes could unambiguously be inferred from the original HapMap data. The incorrect genotype percentage is defined as the percentage of ambiguous single SNP genotypes (heterozygous or missing) that had their phase incorrectly inferred, while the incorrect haplotype percentage measures the percentage of ambiguous individuals whose inferred haplotypes are not completely correct.

For all types of synthetic datasets ENT produces phasings with accuracy that is worse but close to that of the much

[1] For comparison, the PHASE algorithm was reported to take months of CPU time on two clusters with a combined total of 238 nodes when phasing the much smaller Phase I release 16a dataset; no PHASE runtimes have been reported for HapMap Phase II data.

CEU Population									
Chr#	ENT			2SNP			ILP		
	RGE	RSE	Runtime	RGE	RSE	Runtime	RGE	RSE	Runtime
1	4.82	1.63	68.12	13.24	20.76	2599	21.62	6.48	59425
2	5.26	1.24	83.40	13.99	17.86	3340	21.45	5.51	77702
3	4.68	1.41	71.72	13.94	20.72	2616	21.05	5.91	41613
4	4.52	1.48	59.17	13.61	20.08	3020	20.83	6.08	38347
5	4.73	1.36	63.10	13.83	20.23	2175	20.86	5.86	40191
6	4.81	1.40	66.21	13.90	20.56	2418	21.28	5.85	66559
7	4.82	1.52	53.70	14.15	21.12	1785	21.50	6.28	52677
8	4.85	1.20	50.16	13.57	17.69	1888	21.22	5.37	52393
9	5.04	1.35	40.22	13.14	18.25	1453	21.19	5.94	38291
10	4.80	1.47	51.96	13.14	20.39	1707	21.72	6.22	55728
11	4.68	1.51	48.89	13.64	20.58	1647	21.21	6.33	28324
12	4.96	1.61	50.92	13.25	21.42	1568	21.79	6.51	28758
13	4.83	1.47	46.65	13.54	20.85	1187	21.32	6.28	18886
14	4.78	1.43	27.45	14.19	19.89	884	21.49	6.00	12852
15	5.74	1.57	27.90	14.52	19.74	705	23.07	6.23	11466
16	5.45	1.67	25.72	14.19	20.28	700	23.48	6.86	12665
17	5.43	1.70	21.50	13.97	19.99	516	22.21	6.60	11906
18	4.72	1.42	22.16	31.91	35.51	1270	20.97	6.06	19570
19	5.62	1.88	12.66	14.54	21.17	356	22.54	6.78	8910
20	4.97	1.49	23.91	12.24	18.72	977	22.19	6.95	29658
21	6.57	1.65	10.51	13.43	16.79	395	22.53	6.13	4548
22	5.93	1.73	12.17	12.46	17.38	314	22.94	6.97	4142
Avg./Total	**5.09**	**1.51**	**938.20**	**14.47**	**20.45**	**33520**	**21.75**	**6.24**	**714611**

YRI Population									
Chr#	ENT			2SNP			ILP		
	RGE	RSE	Runtime	RGE	RSE	Runtime	RGE	RSE	Runtime
1	8.86	2.03	89.32	18.52	23.98	2970	26.47	7.12	61277
2	8.75	1.67	88.34	19.82	22.80	3658	27.11	6.19	68751
3	8.33	1.72	72.36	19.40	23.56	3778	26.90	6.52	39690
4	8.71	2.05	76.35	19.02	24.61	3261	26.23	6.98	35405
5	8.80	1.81	68.01	19.35	23.50	3009	27.14	6.54	37308
6	8.06	1.73	73.51	17.98	23.18	2544	26.31	6.54	67301
7	8.54	1.98	63.66	19.55	24.90	1856	27.44	7.12	49580
8	8.78	1.55	50.34	19.27	21.10	2013	27.59	5.99	49396
9	8.78	1.74	48.49	19.29	21.65	1553	27.25	6.60	36810
10	8.91	1.91	60.74	19.12	23.52	1963	27.33	6.99	55004
11	8.38	2.03	66.54	18.71	24.74	1703	26.69	7.30	26510
12	9.06	2.16	54.44	19.06	24.67	1640	28.04	7.50	27524
13	8.58	1.74	41.02	18.69	22.98	1380	26.89	6.56	18261
14	8.79	1.76	30.69	19.29	22.88	910	27.52	6.53	12229
15	9.60	2.02	27.44	20.24	23.51	757	28.76	7.00	10868
16	10.32	2.37	31.34	20.68	25.39	814	28.85	7.75	12454
17	9.96	2.29	22.56	20.54	24.65	662	28.53	7.56	11226
18	8.79	1.87	29.00	37.13	38.44	1420	25.86	6.61	19568
19	10.48	2.47	14.26	20.15	23.02	449	28.58	7.72	8538
20	9.20	1.98	24.83	34.33	39.02	1069	28.68	7.70	28871
21	8.73	1.75	11.30	18.45	20.89	430	26.78	6.54	4589
22	10.09	2.10	12.39	18.86	19.89	404	28.02	7.47	4212
Avg./Total	**9.02**	**1.94**	**1056.93**	**20.79**	**24.68**	**38243**	**27.41**	**6.95**	**685372**

TABLE III

COMPARISON RESULTS ON HAPMAP PHASE II CEU AND YRI DATASETS.

slower methods included in the comparison. We remark that Table V reflects the latest results available at `http://www.stats.ox.ac.uk/˜marchini/phaseoff.html`. Accuracies reported for some methods and datasets are slightly different from those published in [6] due to inconsistencies discovered by the authors after the publication of the paper.

In Table VI we present accuracy results for PHASE, fastPHASE, 2SNP, HAP, and ENT on the real dataset from [19], consisting of 80 unrelated genotypes for which the real haplotypes have been experimentally determined. For this dataset, we report the same accuracy measures as in Table V, computed using as reference both the real haplotypes and the haplotypes inferred by PHASE. With respect to all three measures, the accuracy of ENT is worse than that of PHASE, fastPHASE, and HAP, but better than that of 2SNP. Although PHASE is not 100% accurate, using the haplotypes inferred by it as a reference does result in the correct relative ranking of the other methods. However, the results in Table VI do suggest that using PHASE haplotypes as ground truth leads to a slight underestimation of true error rates.

C. *Effect of missing data*

In a second set of experiments we assessed the accuracy of the four most scalable methods (ENT, 2SNP, ILP, and HAP) in the presence of varying amounts of missing genotype data. For these experiments we used the trio populations of the HapMap

JPT+CHB Population						
Chr#	ENT			2SNP		
	RGE	RSE	Runtime	RGE	RSE	Runtime
1	8.63	5.26	735.96	-	-	-
2	7.84	4.48	780.27	-	-	-
3	8.11	4.81	642.04	-	-	-
4	8.47	4.97	619.17	-	-	-
5	7.88	4.63	617.75	-	-	-
6	8.59	4.75	656.95	-	-	-
7	8.30	5.12	534.75	-	-	-
8	9.09	4.43	571.12	-	-	-
9	9.47	5.02	464.30	-	-	-
10	8.66	5.17	514.10	4.93	3.13	254960
11	9.77	4.92	491.08	5.50	2.82	227630
12	8.79	6.00	475.08	5.51	3.79	221245
13	8.04	4.94	390.07	4.69	2.90	138481
14	8.39	4.77	290.93	5.18	2.98	46741
15	9.83	5.33	257.82	6.07	3.57	37166
16	9.58	5.89	255.55	6.23	3.99	35300
17	8.98	5.97	208.62	5.64	4.16	20886
18	9.27	5.22	286.31	5.37	3.23	28576
19	9.97	6.75	136.46	6.82	4.96	6886
20	8.40	5.90	222.29	5.17	3.57	22463
21	9.53	4.96	133.49	5.57	3.34	6422
22	10.94	6.09	128.03	6.37	3.95	6681
Avg./Total	**8.93**	**5.24**	**9412.13**	**5.62**	**3.57**	**857495**

TABLE IV

COMPARISON RESULTS ON HAPMAP PHASE II JPT+CHB DATASET.

Sample	PHASE v2.1	fastPHASE	HAP	HAP2	ENT
	Switch error				
RT-CEU	0.53	-	2.05	2.95	5.88
RT-YRI	2.16	-	4.44	-	9.29
RU	8.41	9.21	10.72	12.56	13.46
	Incorrect genotype percentage				
RT-CEU	0.05	-	0.40	0.33	1.40
RT-YRI	0.16	-	0.33	-	0.93
RU	7.47	-	8.04	8.17	8.31
	Incorrect haplotype percentage				
RT-CEU	6.20	-	20.78	20.42	40.40
RT-YRI	15.7	-	29.25	-	48.92
RU	77.66	83.57	87.96	87.67	91.61

TABLE V

COMPARISON RESULTS ON HAPMAP-BASED SYNTHETIC DATASETS FROM [6].

Reference	PHASE v2.1	fastPHASE	2SNP	HAP	ENT
	Switch error				
True haps	2.60	5.84	13.64	6.49	11.04
PHASE haps	0.00	4.55	11.04	5.19	9.74
	Incorrect genotype percentage				
True haps	0.56	1.25	2.92	1.39	2.36
PHASE haps	0.00	0.83	2.36	0.97	1.94
	Incorrect haplotype percentage				
True haps	5.00	11.25	20.00	11.25	15.00
PHASE haps	0.00	7.50	15.00	7.50	11.25

TABLE VI

COMPARISON RESULTS ON THE REAL DATASET FROM [19].

Phase I release 16a from which we randomly deleted 0-20% of the SNP genotypes. The results obtained for chromosome 22 are summarized in Table VII. For low amounts of missing data, ENT accuracy is similar or better than that of the other three methods. For all methods, the error rates increase with the percentage of missing SNP genotypes. ENT error rate does seem to degrade faster than that of 2SNP and HAP, with HAP being the most accurate for 20% missing genotypes. 2SNP and ILP runtimes seem to be insensitive to the amount of missing data, while ENT and HAP runtimes increase with the percentage of missing SNP genotypes. ENT remains much faster than the other methods even for 20% missing genotypes.

D. *Effect of pedigree information*

In a third set of experiments we assessed improvements in accuracy due to the availability of pedigree information. Two synthetic datasets were created based on the HapMap Phase I CEU and YRI haplotype data for chromosome 22. Families with two parents and two children were created for each trio in these populations by starting from the reference phasing of parent genotypes and then creating two children genotypes by randomly pairing parent haplotypes. The resulting genotypes were used to create three different datasets incorporating varying degrees of knowledge about true inheritance patterns (see Figure 7):

- Children genotypes treated as unrelated individuals;
- Two independent parents-child trios for each family (this allows parent genotypes to be phased differently in the two trios); and
- One pedigree per family describing the full inheritance pattern between the four members.

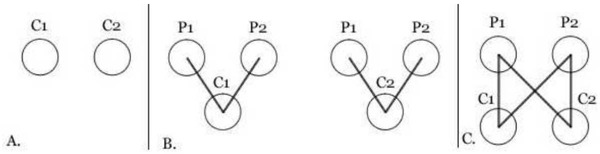


Fig. 7. Full-sibling experiment: (A) children treated as unrelated individuals; (B) independent trio decomposition; and (C) full inheritance pattern.

Table VIII gives child genotype phasing accuracy obtained by running the fastPHASE, 2SNP, HAP, ILP, and ENT algorithms on the three datasets, using each method with default parameters. Since there is no missing data in our MapMap Phase I genotypes, RGE is always equal to 0. To enable a meaningful comparison across the three scenarios, which result in different numbers of ambiguous SNP genotypes, in addition to RSE we also report the average number of switches required to transform the inferred haplotypes of a child into the reference ones. The performance of ENT compared to that of the other methods is consistent with the results presented in Section IV-B. As expected, for all methods that can be run on multiple datasets (ENT, 2SNP, and HAP) the absolute accuracy (as measured by the number of switches per child) is improving with the amount of pedigree information. Interestingly, the relative accuracy measured by RSE is also improving with the amount of pedigree information for ENT and HAP, but

Deleted		ENT		2SNP		ILP		HAP	
		CEU	YRI	CEU	YRI	CEU	YRI	CEU	YRI
0%	RGE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RSE	1.23	1.66	4.98	8.97	3.85	4.77	1.35	1.58
	CPU	1.94	2.01	1248	1380	855	887	942.43	1168.76
1%	RGE	4.89	7.05	6.01	10.51	18.46	23.56	5.25	6.28
	RSE	1.51	2.05	5.06	9.06	4.50	5.56	1.39	1.66
	CPU	2.89	3.03	1298	1445	863	895	991.22	1255.70
2%	RGE	5.18	7.69	6.02	10.58	18.75	23.86	5.36	6.43
	RSE	1.82	2.48	5.12	9.15	5.04	6.28	1.40	1.79
	CPU	3.97	4.16	1306	1397	860	912	1116.14	1293.91
5%	RGE	5.97	8.95	6.54	11.28	18.58	24.12	5.87	7.00
	RSE	2.76	3.72	5.33	9.39	6.72	8.44	1.67	2.17
	CPU	7.95	8.28	1318	1423	828	906	1211.81	1431.53
10%	RGE	7.43	11.11	7.26	12.70	19.48	25.61	6.76	8.18
	RSE	4.32	6.05	5.62	9.90	9.25	12.04	2.21	3.06
	CPU	16.77	17.40	1322	1425	824	919	1394.27	1648.70
20%	RGE	10.65	15.51	9.66	15.99	22.66	29.53	8.42	10.66
	RSE	8.13	11.66	6.39	10.91	14.58	19.27	3.38	5.29
	CPU	44.47	47.03	1294	1460	832	995	1800.33	2289.53

TABLE VII

Comparison results for HapMap Phase I Chromosome 22 (15,548 SNPs for CEU and 16,386 SNPs for YRI) with 0-20% deleted SNPs.

not for 2SNP. The ENT version that uses the full pedigree information outperforms all other methods. Including the full pedigree information also speeds up the ENT algorithm, as it reduces the number of zero-recombination phasings that need to be enumerated in each local improvement iteration.

V. Conclusions

In this paper we have presented a highly scalable algorithm for genotype phasing based on entropy minimization. Experimental results on large datasets extracted from the HapMap repository show that our algorithm is several orders of magnitude faster than existing phasing methods while achieving a phasing accuracy close to that of best existing methods. A unique feature of our algorithm is that it can handle related genotypes coming from complex pedigrees, which can lead to significant improvements in phasing accuracy over methods that do not take into account pedigree information. The open source code implementation of our algorithm and a web interface are publicly available at `http://dna.engr.uconn.edu/˜software/ent/`.

In ongoing work we are integrating the ENT algorithm with Hidden Markov Models of haplotype diversity to obtain scalable methods for genotype error detection, haplotype frequency estimation, and haplotype-based whole-genome disease association.

Acknowledgments

We would like to thank Dumitru Brinza and Alex Zelikovsky for their help with 2SNP and ILP experiments, Eran Halperin and Eleazar Eskin for kindly providing us a HAP executable, and Jonathan Marchini for providing us with ENT accuracy measures on the synthetic datasets from [6].

References

[1] B. Paşaniuc and I. Măndoiu, "Highly scalable genotype phasing by entropy minimization," in *Proc. 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 3482–3486.

[2] D. Gusfield, "An overview of combinatorial methods for haplotype inference," in *Proc. DIMACS/RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotype Inference*, 2004, pp. 9–25.

[3] B. Halldorsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail, "A survey of computational methods for determining haplotypes," in *Proc. DIMACS/RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotype Inference*, 2004, pp. 26–47.

[4] T. Niu, "Algorithms for inferring haplotypes," *Genet. Epid.*, vol. 27, pp. 334–347, 2004.

[5] R. Salem, J. Wessel, and N. Schork, "A comprehensive literature review of haplotyping software and methods for use with unrelated individuals," *Human Genomics*, vol. 2, pp. 39–66, 2005.

[6] J. Marchini, D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. Qin, H. Munro, G. Abecasis, P. Donnelly, and International HapMap Consortium, "A comparison of phasing algorithms for trios and unrelated individuals," *American Journal of Human Genetics*, vol. 78, pp. 437–450, 2006.

[7] E. Halperin and R. Karp, "The minimum-entropy set cover problem," in *Proc. Annual International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[8] http://www.hapmap.org/.

[9] H. Ackerman, S. Usen, R. Mott, A. Richardson, F. Sisay-Joof, P. Katundu, T.Taylor, R. Ward, M. Molyneux, M. Pinder, and D. P. Kwiatkowski, "Haplotypic analysis of the tnf locus by association efficiency and entropy," *Genome Biology*, vol. 4, pp. R24.1–R24.13, 2003.

[10] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander, "High-resolution haplotype structure in the human genome." *Nature Genetics*, vol. 29, no. 2, pp. 229–232, 2001.

[11] I. Măndoiu and B. Paşaniuc, "Haplotype inference by entropy minimization," in *9th Annual International Conference on Research in Computational Molecular Biology (RECOMB) Poster Book*, 2005, pp. 221–222.

[12] Z. Qin, T. Niu, and J. Liu, "Partition-ligation – expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms," *Am. J. Hum. Genet.*, vol. 71, pp. 1242–1247, 2002.

[13] E. Eskin, E. Halperin, and R. Sharan, "Optimally phasing long genomic regions using local haplotype predictions," in *Proc. Second RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotypes*, 2004, pp. 13–16.

[14] D. Gusfield, "Haplotyping by pure parsimony," in *Proc. 14th Annual Symp. on Combinatorial Pattern Matching (CPM)*, 2003, pp. 144–155.

[15] L. Wang and Y. Xu, "Haplotype inference by maximum parsimony," *Bioinformatics*, vol. 19, pp. 1773–1780, 2003.

[16] D. Brown and I. Harrower, "A New Integer Programming Formulation for the Pure Parsimony Problem in Haplotype Analysis," in *Algorithms in Bioinformatics, 4th International Workshop (WABI)*, ser. Lecture Notes in Bioinformatics, I. Jonassen and J. Kim, Eds., vol. 3240, 2004, pp. 254–265.

[17] G. Lancia, M. Pinotti, and R. Rizzi, "Haplotyping populations by pure

		Unrelated				2 Trios				Full
		ENT	fastPHASE	2SNP	HAP	ENT	2SNP	HAP	ILP	ENT
CEU	RSE	6.94	3.54	5.23	4.82	3.97	7.04	3.17	14.01	1.97
(15,548 SNPs)	#switches/child	361.52	184.43	272.40	250.78	40.76	72.18	32.51	143.51	20.18
	CPU	27.56	12960	588.70	1756.39	5.83	328.8	2069.59	15051.33	2.24
YRI	RSE	12.20	4.97	9.53	8.59	5.11	13.25	3.07	16.89	2.75
(16,386 SNPs)	#switches/child	638.42	247.11	499.06	449.64	52.55	136.28	31.59	173.69	28.34
	CPU	26.01	23016	648.60	2268.70	5.76	325.3	2510.65	15612.56	1.42

TABLE VIII

RESULTS FOR HAPMAP PHASE I CHROMOSOME 22 FULL-SIBLINGS EXPERIMENT.

parsimony: Complexity of exact and approximation algorithms," *INFORMS Journal on Computing*, vol. 16, pp. 348–359, 2004.

[18] J. Xiao, L. Liu, L. Xia, and T. Jiang, "Fast elimination of redundant linear equations and reconstruction of recombination-free mendelian inheritance on a pedigree," in *Accepted by ACM-SIAM Symposium on Discrete Algorithms(SODA'2007)*, 2007.

[19] S. H. Orzack, D. Gusfield, J. Olson, S. Nesbitt, L. Subrahmanyan, and V. P. S. Jr., "Analysis and Exploration of the Use of Rule-Based Algorithms and Consensus Methods for the Inferral of Haplotypes," *Genetics*, vol. 165, no. 2, pp. 915–928, 2003.

[20] M. Stephens and N. J. Smith and Peter Donnelly, "A new statistical method for haplotype reconstruction from population data," *American Journal of Human Genetics*, vol. 68, pp. 978–989, 2001.

[21] D. Branza and A. Zelikovsky, "2snp: scalable phasing based on 2-snp haplotypes," *Bioinformatics*, vol. 22, no. 3, pp. 371–373, 2006.

[22] D. Branza, J. He, W. Mao, and A. Zelikovsky, "Phasing and missing data recovery in family trios," *Lecture Notes in Computer Science*, vol. 3515, pp. 1011–1019, 2005.

[23] M. Stephens and P. Donnelly, "A comparison of bayesian methods for haplotype reconstruction from population genotype data." *American Journal of Human Genetics*, vol. 73, pp. 1162–1169, 2003.

[24] M. Stephens and P. Scheet, "Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation." *American Journal of Human Genetics*, vol. 76, pp. 449–462, 2005.

[25] P. Scheet and M. Stephens, "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase." *American Journal of Human Genetics(to appear)*, 2006.

[26] E. Halperin and E. Eskin, "Haplotype reconstruction from genotype data using imperfect phylogeny." *Bioinformatics*, vol. 20, pp. 1842–1849, 2004.

[27] S. Lin, A. Chakravarti, and D. Cutler, "Haplotype and Missing Data Inference in Nuclear Families," *Genome Res.*, vol. 14, no. 8, pp. 1624–1632, 2004.

Alexander Gusev received his B.S. degree in Computer Science from the University of Connecticut in 2007. During his time as an undergraduate, he worked as an undergraduate researcher in the bioinformatics lab directed by Dr. Măndoiu. Most recently, he was awarded honorable mention by the NSF Graduate Research Fellowship. In the Fall of 2007, he will begin working on his Ph.D. in the field of Computational Biology and Bioinformatics at Columbia University. His research interests include bioinformatics, machine learning, and population genetics. In his free time he enjoys programming contests, political debate and camping.

Ion I. Măndoiu received the M.S. degree from Bucharest University in 1992 and the Ph.D. degree from Georgia Institute of Technology in 2000, both in Computer Science.

Between 2000 and 2003, Dr. Măndoiu was a postdoctoral researcher and then Research Scientist at the University of California at Los Angeles and at San Diego. Currently, he is an Assistant Professor with the Computer Science and Engineering Department at the University of Connecticut, Storrs. He is the author of over 60 refereed journal and conference articles. His main research interests are in the design and analysis of approximation algorithms for NP-hard optimization problems, particularly in the areas of bioinformatics, design automation, and ad-hoc wireless networks.

Dr. Măndoiu is founding Co-Chair of the *ACIS International Workshop on Self-Assembling Wireless Networks (SAWN)* and has served as Program Committee Chair for several conferences, including the *2007 ACM/IEEE System Level Interconnect Prediction Workshop (SLIP)*, the *2007 International Symposium on Bioinformatics Research and Applications (ISBRA)*, and the *2007 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. He also serves on the editorial board of the *International Journal of Bioinformatics Research and Applications*, and is a guest editor for *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *International Journal of Wireless and Mobile Computing*, and the *Journal of Universal Computer Science*. He received the NSF Faculty Early Career Development Award in 2006.

Bogdan Paşaniuc received his B.Sc. degree in Computer Science from the "A. I. Cuza" University of Iaşi in 2003. While completing his B.Sc. degree, he received an ERASMUS fellowship to study for one semester at the University of Granada, Spain. Currently he is working towards his Ph.D. within the area of Bioinformatics at the Department of Computer Science and Engineering, University of Connecticut. He received several travel awards for courses and conferences, including the 2006 *Genetics of Complex Human Diseases* course at Cold Spring Harbor Laboratory, the workshop on *Developing Tools for a New Generation of Biodiversity Data* held in July 2006 at The National Museum of Natural History, Paris, and the 2007 *Summer Institute in Statistical Genetics* at the Department of Biostatistics, University of Washington. His research interests include bioinformatics, computational molecular biology and microfluidics biochips.