HAHap: A read-based haplotyping method using hierarchical assembly

Yu-Yu Lin, Ping Chun Wu, Pei-Lung Chen, Yen-Jen Oyang, Chien-Yu Chen

1 Department of Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan
2 Taipei Blood Center, Taiwan Blood Services Foundation
3 Graduate Institute of Medical Genomics and Proteomics, College of Medicine, National Taiwan University, Taipei, Taiwan
4 Department of Bio-Industrial Mechatronics Engineering, National Taiwan University, Taipei, Taiwan

Corresponding Author: Chien-Yu Chen
Email address: chienyuchen@ntu.edu.tw

Background: The need for read-based phasing arises with advances in sequencing technologies. The minimum error correction (MEC) approach is the primary trend to resolve haplotypes by reducing conflicts in a SNP-fragment matrix. However, it is frequently observed that the solution with the optimal MEC might not be the real haplotypes, due to the fact that MEC methods consider all positions together and sometimes the conflicts in noisy regions might mislead the selection of corrections. To tackle this problem, we present a hierarchical assembly-based method designed to progressively resolve local conflicts.

Results: This study presents HAHap, a new phasing algorithm based on hierarchical assembly. HAHap leverages high-confident variant pairs to build haplotypes progressively. The phasing results by HAHap on both real and simulated data, compared to other MEC-based methods, revealed better phasing error rates for constructing haplotypes using short reads from whole-genome sequencing. We compared the number of error corrections on real data with other methods, and it reveals the ability of HAHap to predict haplotypes with a lower number of error corrections. We also used simulated data to investigate the behavior of HAHap under different sequencing conditions, highlighting the applicability of HAHap in certain situations.
HAP: A read-based haplotyping method using hierarchical assembly

Yu-Yu Lin¹, Ping-Chun Wu², Pei-Lung Chen², Yen-Jen Oyang¹, and Chien-Yu Chen³

¹ Department of Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan
² Graduate Institute of Medical Genomics and Proteomics, College of Medicine, National Taiwan University, Taipei, Taiwan
³ Department of Bio-Industrial Mechatronics Engineering, National Taiwan University, Taipei, Taiwan

Corresponding Author:

Chien-Yu Chen

No. 1, Sec. 4, Roosevelt Rd., Taipei, 106, Taiwan

Email address: chienyuchen@ntu.edu.tw

Abstract

Background: The need for read-based phasing arises with advances in sequencing technologies. The minimum error correction (MEC) approach is the primary trend to resolve haplotypes by reducing conflicts in a SNP-fragment matrix. However, it is frequently observed that the solution with the optimal MEC might not be the real haplotypes, due to the fact that MEC methods consider all positions together and sometimes the conflicts in noisy regions might mislead the selection of corrections. To tackle this problem, we present a hierarchical assembly-based method designed to progressively resolve local conflicts.
Results: This study presents HAHap, a new phasing algorithm based on hierarchical assembly. HAHap leverages high-confident variant pairs to build haplotypes progressively. The phasing results by HAHap on both real and simulated data, compared to other MEC-based methods, revealed better phasing error rates for constructing haplotypes using short reads from whole-genome sequencing. We compared the number of error corrections on real data with other methods, and it reveals the ability of HAHap to predict haplotypes with a lower number of error corrections. We also used simulated data to investigate the behavior of HAHap under different sequencing conditions, highlighting the applicability of HAHap in certain situations.

1 Introduction

Haplotype phasing, also known as haplotyping, is the process of resolving precise haplotypes. A haplotype describes serial genetic variants that co-occur on a single chromosome. Characterization of haplotypes is essential in various research problems, including allelic expression (Castel et al. 2016), linkage analysis, association studies (Nalls et al. 2014; Ripke et al. 2013), population genetics (Sankararaman et al. 2014; Schiffels & Durbin 2014) and clinical genetics (Zanger & Schwab 2013). Many studies have proposed experimental or computational phasing approaches to resolve haplotypes (Browning & Browning 2011; Snyder et al. 2015). Advances in sequencing technologies have facilitated faster and cheaper resolution of haplotypes, while increasing the need for efficient and effective computational methods.

Several computational phasing approaches have been proposed in recent years. Genetic phasing makes use of related individuals to achieve better precision but can only be adopted when pedigrees are available (O’Connell et al. 2014). Population phasing uses genotyping data of a large
cohort to infer haplotypes (Glusman et al. 2014). However, this approach is only applicable to well-known variants in a population. Read-based phasing utilizes reads spanning at least two heterozygous variants to infer haplotypes (Edge et al. 2017; Garg et al. 2016; Mazrouee & Wang 2014; Pirola et al. 2016). With the appearance of long-read sequencing technologies, where longer reads could possibly cover more heterozygous variants, read-based phasing has increasingly become more appealing. However, since long-read sequencing still suffers high cost at this moment, it may not deliver as high coverage as short-read sequencing. In this regard, individual phasing based on short reads with a sequencing depth of at least 30x remains competitive nowadays. Even though individual phasing based on short reads cannot deliver as high quality as the long-read or population-based solutions (Choi et al. 2018), whole genome sequencing (WGS) becomes more and more critical and frequently used in medical research and precision medicine (Ellingford et al. 2016; Glusman et al. 2014; Luukkonen et al. 2018; Sousa-Pinto et al. 2016; Stavropoulos et al. 2016)

An important computational model developed for read-based phasing is the minimum error correction (MEC) model (Lancia G 2001). The MEC approach corrects putative errors in the reads by changing the alleles that cause conflicts. Conflicts are the alleles in reads that do not support the predicted haplotypes. In the MEC approach, conflicts are considered as errors, which might be sequencing errors or alignment errors. The MEC problem was proven to be NP-hard (Lippert et al. 2002). In this regard, methods delivering the optimal solutions were usually time-consuming (Chen et al. 2013; He et al. 2010), and are considered impractical in whole genome phasing with a sequencing depth of 30x or more. Here we used an Illumina WGS sample downloaded from the GIAB as an example. This sample (sample ID: NA24143) consisted of 2x250 paired-end reads, with a sequencing depth of about 40x. By using a 48-core machine with Intel Xeon E5-2683 CPUs
and 384 GB of memory, an ILP method, HapAssembly, took more than 150 hours to complete the phasing, in both general and all-heterozygous modes. This is impractical for an application that contains more than hundreds of WGS samples.

In this regard, many computational methods have been proposed to speed up the phasing performance. WhatsHap (Garg et al. 2016) is a well-known phasing tool that down-samples the read sets first and uses a dynamic programming fixed parameter tractability (FPT) algorithm to solve a weighted MEC problem faster. While WhatsHap delivers optimal solutions, several heuristic methods have been proposed to find haplotypes more efficiently (Aguiar & Istrail 2012; Edge et al. 2017; Xie et al. 2015). HapCut2 is a maximum likelihood estimation heuristic method that uses max-cut computation to search a subset of variants such that changing haplotypes on those variants is possible to achieve a greater likelihood. The procedure is repeated until no further improvements are possible, eventually leading to a near-optimal approach of MEC.

Although MEC is regarded as the state-of-the-art strategy, there is room for improvement. First, the quality of the reconstructed haplotypes may be severely affected by sequencing and alignment errors. Conceptually, MEC methods search solutions in a global manner, which works well if there are relatively fewer noises than signals. Second, read-based phasing struggles when handling regions with dense variants, which is computationally infeasible for many existing methods. The execution time of a read-based phasing method is highly related to the sequencing coverage and the number of variants. In the future, longer reads from third-generation sequencing technologies will necessitate a more time-efficient method, since longer reads involve more variants in the problem.
This study aims to develop a heuristic read-based haplotyping method based on a hierarchical assembly algorithm. Our method is not designed to solve the MEC problem. Instead, it attempts to eliminate the influence of noises through iteratively considering the most reliable information from variant pairs. We developed an adjusted multinomial probabilistic metric for evaluating the reliability of a variant pair, and the derived scores guide the assembly process. Once only the pairs with low scores remain, we accept a local MEC search method to resolve the haplotypes of this local region, with a much smaller search space than what used to be. The evaluation was performed using the haplotype prediction of the Ashkenazim trio by 10x Genomics, which uses a barcoding technique followed by pooled short read sequencing to resolve haplotypes (Porubsky et al. 2017; Zheng et al. 2016). This study compares the proposed method with an exact method tool, WhatsHap, and a heuristic method, HapCut2, on both real and simulated data. We also generated simulated data in different situations based on the hg19 genome in order to study the applicability of HAHap. While local phasing information becomes more and more important in pathogenicity studies and clinical diagnosis (Cheng et al. 2018; Wu et al. 2018), we provided an example of phasing using short reads, ABO blood type detection, in the end, to illustrate how short-read phasing can help real medical applications.

2 Methods

HAHap is a read-based haplotyping algorithm that adopts hierarchical assembly to progressively build haplotypes, starting from high-confident pairs and working toward low-confident pairs (Figure 1). It takes predicated heterozygous variants from a variant caller as the input. We describe the concept of hierarchical assembly in sections 2.1 and 2.2; the design of probabilistic confidence scores (CSs) in 2.3; and the local MEC search in 2.4 and 2.5.
2.1 Haplotype-informative reads and variant blocks

A locus with the same alleles is homozygous, and a locus with more than one allele is heterozygous. In the phasing problem, we only consider heterozygous variants. The reads spanning multiple heterozygous variants are used in assembling haplotypes. The homozygous variants, which cannot provide information to extend haplotypes, are ignored. In the rest of the article, 'variant' is short for 'heterozygous variant.' Two variants are connected if they are spanned by a read. The connectivity of variant pairs is transitive. For example, suppose there are three variants A, B and C, but no read spans A and C. However, there are reads spanning variants A and B and other reads spanning variants B and C. In this case, A and C can be joined through B, and we describe each pair of variants among these three variants connected. In other words, the reads spanning multiple variants are haplotype-informative. In this regard, the first step of the proposed method is to identify the haplotype-informative reads and determine a variant block in which all the variants inside are connected.

DNA sequencing technology produces reads or read pairs, i.e., fragments of DNA. In the field of computer science, we usually define a read as a string of nucleotides [A, T, C, G]. In the phasing problem, we describe it as a string of [0,1,-] to stand for the allele of [major, minor, others]. A variant block is a set of variants where each one has at least one connection with the others. The informative reads represent reads that span at least two variants.

2.2 Hierarchical assembly

HAHap considers each pair of two loci as pieces of the puzzle and proceeds with the assembly according to the CSs. To resolve the whole haplotypes, we start from phasing small pieces and
assemble them using the concept of hierarchical assembly. Each resultant variant block in section 2.1 is an isolated problem for hierarchical assembly.

Although the MEC method in general capably handles noise, it can still accumulate errors and thus lead to poor phasing when noise overrides the signal in the data. According to this observation, our method endeavors to reduce the influence of noise by leveraging pairs with higher scores. This is achieved by adopting a hierarchical assembly approach, which considers the pair with the highest score first and proceeds to the end. At the beginning of the assembly, the algorithm treats each variant as a vertex and calculates a CS of two vertices for guidance. The CS of a variant pair is described in section 2.3. Then, the single-linkage metric is adopted to evaluate the score between two clusters. The term 'cluster' represents a group of variants and is used in the following discussion. The algorithm continues to unify clusters until all vertices are unified. In total, a block with $n$ variants inside needs $n-1$ merging to group all variants into one unit. This process of assembly could be finished in linear time.

The score $S(X, Y)$ between two clusters $X$ and $Y$ is defined below, where $X$ and $Y$ are any two sets of variants as clusters. The CS takes two variants, $x$ and $y$, as arguments. It ranges from negative infinity to zero. A score closer to zero indicates more confidence.

$$S(X, Y) = \max_{x \in X, y \in Y} CS(x, y) \quad \text{(single-linkage)}$$

HAHap always trusts the heterozygous variants reported by the variant caller. In order words, it only considers the major and alternative alleles detected in variant calling and treats the variants as heterozygous. This is the so-called heterozygous assumption. For example, if the variant caller determines $V_1$ and $V_2$ as two heterozygous loci, and $V_1$ is observed with C/T (0/1) and $V_2$ with
G/C (0/1), the heterozygous assumption tells us that the haplotype candidate [C-G, C-C] is not considered since [C-G, C-C] turns V₁ homozygous. Only two solutions are preferred; one is [C-G (0-0), T-C (1-1)] and the other is [C-C (0-1), T-G (1-0)].

After explaining the rules of the hierarchical assembly, this paragraph introduces a way to resolve one large haplotype from two small phased haplotypes. Our method takes the reads spanning the variant pair of the highest score between two clusters to combine the sub-haplotypes that have been phased in previous steps. Among two preferred solutions, the first solution’s alleles for the pair are [0-0, 1-1] and those of the second solution are [0-1, 1-0]. We calculated the confidence scores, CS₁ and CS₂, for the first and second solutions, respectively, and used the read information of the higher CS to infer a larger haplotype (Figure 2). The hierarchical assembly continually executes until all possible pairs are considered. However, in a heuristic method, it is possible to make a wrong decision on merging with a low CS. Whenever it is likely to happen, HAHap introduces a local MEC search to overcome this problem, which will be explained in section 2.4.

2.3 Multinomial distribution metric

In this section, we introduce the scoring metric that measures the confidence between two variants. The CS is based on the multinomial distribution. Each pair of the two connected variants is treated as an instance for scoring. When appraising the confidence level of variants x and y, we count the total reads that span both x and y, and denote this number as N. Then we consider two preferred solutions separately. First, we define the observed read count of one haplotype as n₁ and the observed read count of the other haplotype as n₂. The sum of n₁ and n₂ is the total observations coming from the solution; therefore, we expect this number to be as close to N as possible. However, the errors from sequencing or alignment bring in unexpected reads that do not fit in any
two haplotypes. To handle these potential errors, we introduce the third number ($n_3$) to take care of the unexpected reads. Let $P_1$ and $P_2$ be the likelihoods of the two haplotypes, and $P_3$ be the likelihood of the unexpected observations. We define the multinomial score, $MS$, as the log$_2$ value of the possibility that the solution satisfies the multinomial distribution when given the observed reads. A normalizing factor, $F$, is used to normalize the effect caused by $N$. Accordingly, the normalized score, $NS$, is defined as the value obtained by subtracting $F$ from $MS$, and the symbol $C$ in formula means the combination (a selection of items from a collection) in mathematics.

$$\begin{align*}
N &= \sum^3_{i=1} n_i \sum^3_{i=1} P_i = 1, \ 0 \leq P_i \leq 1 \\
MS &= \log_2 \left( C^N_{n_1} C^{N-n_1}_{n_2} P_1^{n_1} P_2^{n_2} P_3^{n_3} \right) \\
F &= \log_2 \left( C^N_{N/2} C^{N-N/2}_{N/2} P_1^{N/2} P_2^{N/2} \right) \\
NS &= MS - F
\end{align*}$$

Here, we need to consider the effect of the sequencing coverage. Let the term ‘max coverage’ stand for the largest read count among all pairs in this block and the term ‘local coverage’ stand for the read count of the two variants of interest. We define the ratio of local coverage and maximum coverage as $c$, and use it to adjust the inflection point of the sigmoid function. In the end, we define $CS$ as shown below, which is actually an adjusted value of $NS$.

$$\begin{align*}
c &= \frac{\text{local coverage}}{\text{max coverage}} \\
sigmoid(c) &= \frac{1}{1 + e^{-\left( c - 0.5 \right)}}
\end{align*}$$
The meaning of $CS$ is as follows. In diploid organisms, ideally, reads should only come from the two haplotypes, and the ratio of the observed read counts from the two haplotypes should be close to 1:1. However, in real cases, sequencing and alignment errors cause the cis-allelic appearance to stray from equality. Based on this assumption, HAHap uses a multinomial distribution to estimate how likely the sequencing reads are sampled following this distribution. As a result, this score evaluates the confidence with which the reads on this pair follow the multinomial assumption. We note that there are two preferred solutions for each pair of the two variants. We choose the higher one as the $CS$ for this pair.

The metric takes three factors to eliminate possible biases caused by different coverages. First, the multinomial score is divided by the normalized score, which represents the perfect non-skewed case of the multinomial distribution. Second, we use a sigmoid function to capture the difference between a good region and a bad region. Third, we use the parameter $c$ to adjust the inflection point of the sigmoid function to keep the score with coverage at $c/2$ unchanged. The method takes the $CS$ to prioritize all pairs of variants and uses it to pilot the hierarchical assembly. The distance between two variants is another important clue. We assume that two closer variants are more reliable. Among pairs with the same score, the method chooses the closer one as the next assembly step.

2.4 Local phasing using MEC

The assembly process reduces the search space dramatically. However, it has the drawback that false inferences could cause serious consequences. Therefore, under certain conditions, we perform a local MEC-based search. When merging two clusters, it is possible that the variants
from the two clusters are interleaved. A junction is the boundary between two adjacent variants from different clusters. The method adopts a voting mechanism in which every read spanning the junctions is recruited to evaluate the two preferred solutions. The penalty is defined as how many corrections have to be made on reads in order to be consistent with the solution. In the end, the solution with a lower penalty will be chosen. The local MEC process checks potential corrections for every read in a block. This step is time-consuming, but fortunately it only executes in local regions in HAHap. Next, we explain when the local MEC process will be invoked in the following two subsections.

2.4.1 Embedded merging

We defined embedded merging as the case that merges two clusters with multiple junctions in between adjacent variants. In this situation, HAHap applies a local MEC-based search instead. One reason is that this kind of situation involves numerous phasing decisions, which means the algorithm should proceed more carefully. Normally, closer variants tend to have a higher CS and are usually merged earlier. In this regard, multiple junctions often happen in the area with erroneous information. In the local MEC-based search, all reads spanning the junctions are responsible for inferring haplotypes. We defined the minimum junction number as the threshold for triggering a local MEC-based search. For example, when the number of minimum junctions is set to three, the merging with three junctions or more will invoke a local MEC-based search.

There are only two potential solutions for a particular MEC-based search (mating the two sub-haplotypes from each cluster, respectively). The search only inspects all reads covering the specified region. On average, the time complexity of the local MEC-based search is proportional to the read coverage in this area.
2.4.2 Only ambiguous variant pairs remain

We declare singleton pairs or low-coverage pairs as ambiguous variant pairs. A singleton pair means all reads on this pair only support one haplotype in both preferred solutions. Low-coverage pairs indicate that the count of reads spanning this pair is below the threshold we defined. By default, HAHap chooses the median among all observations as the threshold. In both cases, the CSs usually fail to guide the assembly correctly. Based on the definition of CSs, the score of a singleton is quite small due to the absence of one haplotype. Similarly, low-coverage pairs provide unreliable CSs. Even though the CSs of singleton pairs are extremely low, they are observed to provide correct phasing information with only one haplotype observation. In this regard, when only ambiguous pair remains, we use singleton pairs earlier than low-coverage pairs during assembly.

Sometimes, there may be too many ambiguous pairs in a single block. In this situation, the method would be slow. To faster the method, we combine the embedded mergence with this rule. When encountering ambiguous pairs and the most strict embedded mergence case (minimum junction number is two), we perform a local MEC-based search.

2.5 Examples of choices between heuristic and local MEC-based search

In a single nucleotide polymorphism (SNP)-partitioning graph, the vertex stands for a variant, and an edge links two connected variants. Figure 3 shows a toy demonstration of phasing, which involves hierarchical assembly in the first two steps and a local MEC-based search in the last step. In this process, the weights on edges are the CS scores, and the edge with the highest CS in this step is used to connect the next pair of variants or variant clusters. In summary, HAHap designs a
metric for conducting hierarchical assembly heuristically and switches to MEC-based searches in
certain situations.

Here, we further used an example to demonstrate that HAHap performs better than MEC-based
methods in certain situations. In Figure 4, five types of reads span a block containing three variants.
The example was transformed as a SNP-variant matrix for the phasing problem. In the upper panel
of Figure 4, we described how a MEC method proceeds to find the solution. In the lower panel,
we showed that HAHap determines the phase between $V_1$ and $V_2$ first, and then extended the
haplotype to $V_3$. In the end, two methods predicted different haplotypes. The MEC method
assumed that the read type 4 and read type 5 are incorrect and flipped two entries to make the
matrix with no conflicts in the resultant haplotypes. On the other hand, HAHap suggested read
types 1, 2 and 3 are incorrect and predicted the result based on read type 4 and read type 5. In other
words, the MEC-method considers the exact number of errors, while HAHap cares more about the
ratio of the observed read counts from the two haplotypes. We managed to validate which method
is better in both real data and simulated data in the next section.

3 Experimental Results

We evaluated real and simulated datasets. WhatsHap and HapCut2 are two representative MEC-
based tools for phasing problems (Mark J.P. Chaisson 2018; Sedlazeck et al. 2018). We compared
HAHap with MEC-based tools to reveal that the proposed idea to reduce the search space and
remove noise is beneficial to phasing. In HAHap, the likelihood of $P_1$ and $P_2$ were 0.49, and the
minimum number of junctions for triggering MEC-local search was three as default. The other two
programs were executed using default parameters.
3.1 Evaluation measurement

Here, we introduce two concepts, ‘variant needing to be phased’ and ‘phased variant’. We define a variant needing to be phased as all variants except the left-most variant in its block. For example, when a block contains seven variants, the first variant does not need to be phased, and only six phasing decisions remain. On the other hand, the term ‘phased variants’ is how many variants are indeed phased by a tool. If a locus is too difficult to phase, the algorithm could leave it as undecided. We use \textit{number\_of\_variants} to stand for the first concept (‘variant needing to be phased’) and \textit{number\_phased} for the second one (‘phased variant’). For convenience, we used the term ‘case’ to refer a variant block in the phasing problem. This study used two measurements, phasing error rate and perfect ratio, for evaluating the accuracy of the phasing algorithms. Additionally, we incorporated a measurement, QAN50, to evaluate the average effective length of the properly phased blocks.

For each block, the first predicted haplotype is always a mosaic of the two true haplotypes, and the second predicted haplotype is exactly the complement of the first one, due to only considering heterozygous sites. In this regard, the switch error is defined as the case where a swap event between two haplotypes happens. However, when two switch errors are adjacent, it is treated as one flip error instead. We used \textit{flip\_error} and \textit{switch\_error} to respectively represent the total number of flip errors and switch errors in a block. The \textbf{phasing error rate} is defined as the sum of the switch and flip errors divided by the number of phased variants.

\[
\text{phasing error rate} = \frac{\text{flip\_error} + \text{switch\_error}}{\text{number\_of\_phased}}
\]  

(9)
The **perfect ratio** is the ratio of the error-free phasing cases over the total phasing cases. We considered it as the true positive rate in this study:

\[
\text{perfect ratio} = \frac{\text{number of error-free case}}{\text{number of case}}
\]  

(10)

Additionally, we used **quality adjusted N50 (QAN50)** to evaluate the completeness and quality of the predicted haplotype. This measure is calculated as follows: (1) breaking each haplotype block into the longest possible sub-blocks where no switch error inside; (2) calculating the distance from the first phased variant to the last phased variant for each sub-block; (3) multiplying the distance of each sub-block by the proportion of phased variants inside the sub-block; and (4) calculating the N50 of the distance set revised in step (3).

### 3.2 Evaluation using real data

The Genome in a Bottle Consortium (GIAB) hosted by the National Institute of Standards and Technology has characterized seven individuals using eleven different technologies (Zook et al. 2014). This study focused on the Ashkenazim trio, consisting of three related individuals: NA24143 (mother), NA24119 (father) and NA24385 (son). The NovoAlign BAM files of 2x250 paired-end reads with coverage around 40x to 50x produced by Illumina HiSeq were downloaded from GIAB and used as input for phasing. GIAB also provides the haplotypes of the Ashkenazim trio predicted by 10x Genomics. The 10x GemCode Technology creates a unique reagent delivery system that partitions long DNA molecules (including >100 kb) and prepares sequencing libraries in parallel such that all fragments produced within a partition share a common barcode. By this barcoding technique and combined with a proprietary data analysis tool called Long Ranger software (v2.1), the prediction could be more reliable and large-scale than other experimental
phasing. This study adopted the 10x Genomics prediction as the answer for haplotyping. This study considered all bi-allelic SNPs on chromosomes 1 to 22, resulting in 5,342,998 SNPs in the mother, 5,220,679 in the father and 4,931,224 in the son. We only considered heterozygous variants, resulting in 3,465,217 heterozygous variants in the mother, 3,406,189 in the father and 3,108,937 in the son. Following the standard pipeline, phasing after variant calling, we used GATK HaplotypeCaller (v3.6) as the variant caller to forecast the variants, which identified the heterozygous loci for phasing tools. We took the intersection of the predictions from 10x Genomics and the variant calling results from GATK as the ground truth to make sure the interested variants are indeed covered by the raw reads. Finally, in the answer set of real data experiment, there are 2,240,300 heterozygous variants in the mother, 2,174,189 in the father and 2,225,891 in the son, which are considered detectable in the Illumina BAM files. Each tool has unique features for identifying variant blocks, for example, different mapping quality cutoffs and re-alignment procedures, so that each tool classifies distinct variant blocks with different numbers of variants inside. To have a fair comparison, we used all reads without filtering to recognize the original distribution of the block sizes (defined as the number of heterozygous variants in a block). The blocks with sizes less than 20 account for 99% (Table-S1) of the cases. Since the challenge of phasing is on the larger blocks, we emphasized our investigation of cases containing variants more than 20.

We discussed the perfect ratios first. Among the blocks with sizes more than 20, we identified 12,784 cases in the BAM files before read filtering and took this number as the original case number. In this condition, HAHap successfully identified more error-free phasing cases (12,191 perfect cases; true positives) than WhatsHap (11,712 cases) and HapCut2 (12,125 cases). Table 1 and Table S1 listed the results for all cases, and Table 2 highlights the results for case sizes larger...
than 20. To investigate the performance relative to block sizes, the true positive rate across different block size is shown in Figure 5. This reveals that HAHap outperformed the others in all categories of block sizes.

Next, we compared the phasing error rates between tools. HAHap performed better (0.228%) in this measurement than WhatsHap (0.511%) and HapCut2 (0.299%). Regarding the block size, HAHap consistently outperformed the others across categories. The results are shown in Figure 6.

We used scatter plots to visualize the comparison and verified the significance of differences. In Figure 7 and Figure 8, we only considered the blocks with sizes larger or equal to 20. Because the three tools predicted perfectly on most of the cases, we excluded those cases in the following statistic testing. After filtering out the cases with ‘phasing error rate’ = 0 on both tools in comparison, only 447 cases remained in the comparison of HAHap and WhatsHap and 485 cases remained in the comparison of HAHap and HapCut2. The p-values of Wilcoxon rank sum test are 8.969E-06 when comparing HAHap versus WhatsHap and 0.0181 when comparing HAHap versus HapCut2. Besides, we observed that the dots below the diagonal lines are much more than the ones above the diagonal lines in both figures. This revealed that WhatsHap and HapCut2 predicted more problematic phased variants (having higher error rates) than HAHap.

The last measure, in terms of completeness and quality, QAN50, showed closer phasing quality between these three programs (Table 3 and Table-S1). The QAN50 for blocks with a size larger than 20 is 4,295 for HAHap and are 4,367 and 4,275 for WhatsHap and HapCut2, respectively. WhatsHap delivered slightly longer correct haplotypes than HAHap and HapCut2.
3.3 Evaluation using simulated data

We conducted 50 runs of simulation with different parameter settings to investigate the performance changes under different situations. First, in each simulation, we used in-house scripts to create simulated SNPs. A genome sequence on chr22, from 16,070,000 to 16,790,000, were chosen as the experimental region, and we created ten variant blocks for each of the five block sizes (containing 30, 50, 100, 200, and 500 variants) on either one of the haplotypes. Second, based on the artificial haplotypes, we simulated reads in each of the nine conditions to create a unique experiment. The nine conditions includes three levels of sequencing error rates (0.002, 0.01, and 0.03) and three levels of read coverages (20, 30, and 40). In total, the evaluation included 450 experiments, i.e., 50 replicates for each of the nine sequencing conditions.

Simulated reads were produced by wgsim. The read length of a paired-end read is 250 bps, and the fragment length follows a normal distribution with a mean of 850 and a standard deviation of 50. We adopted BWA-MEM as the mapper and all the chromosomes of hg19 as the reference, and then took the mapped files as the input for phasing.

Here, we discussed the phasing error rates first. Among the nine sequencing conditions, shown in Figure 9 and Table S2, HAHap performed better than WhatsHap in all conditions and outperformed HapCut2 in most conditions (better in seven conditions and worse in two). The results are consistent with the results in the real data. On the other hand, when evaluating the performance according to block sizes, as shown in Figure 10 and Table S2, HAHap outperformed the other two tools in most of the block sizes (except the block size 200-500 when compared with HapCut2). Second, the perfect ratios of three tools are pretty close to each other (all above 98%), and the detailed results are shown in Table S2. The number of blocks is small in the simulation,
causing the QAN50 less meaningful. The details of QAN50 comparison were provided in Table S2.

3.4 Evaluation HAHap on sequencing skewness

We investigated the performance of HAHap in one more sequencing condition, sequencing skewness. Skewness means the sequencing read unbalance on two haplotypes. After integrating five levels of skewness into the previous simulation design, where nine conditions were included when considering two factors, we had 45 conditions in this examination. We did 25 times of simulation and inspected the effect of the three factors, respectively. Results of evaluating 45 distinct configurations were exhibited in Table 4 and Table S3.

Under the same conditions, higher coverage and lower sequencing error rates usually facilitate better phasing performance. Only a few exceptions exist, but the difference is tiny (e.g., the combination of ‘coverage=40’, ‘error rate=0.002’ and ‘skewness=50/50’ is 0.00041% worse than the combination of coverage 30 with the other factors the same).

Skewness is a factor that attracts attention because the proposed CS is based on a multinomial distribution where the likelihoods of the first two outcomes are presumed to be equal. In this regard, a skewness that violates the assumption would confuse the assembly process. Table 4 reveals that with a skewness of 10/90, HAHap performed much worse than 50/50. We observe that extreme skewness leads to more singletons and pairs with much lower scores, which eventually damages the method. Fortunately, HAHap still performs well with skewness of 30/70 (with a phasing error rate close to that of 50/50), and 10/90 is an extreme case that rarely happens in real sequencing. In summary, we recommend not using HAHap under conditions of extreme skewness
Generally speaking, HAHap is a reliable phasing method under various conditions.

### 3.5 Comparison of number of error corrections

To evaluate the performance in terms of the number of error corrections (ECs) as in MEC-based approaches, we compared the number of ECs between HAHap and the other methods using real data. The EC rate is calculated as the percentage of ECs over the total characters covering reads.

We drew each case on a scatter plot, where the y-axis stands for the EC rate of HAHap, and the x-axis for the tool to be compared. Both Figures 11 and 12 use distinct symbols to label which method achieved better phasing error rates. In comparison between HAHap and WhatsHap (Figure 12), we investigated 700 cases for which two tools have different predictions. HAHap outperformed WhatsHap in 547 cases, where the weighted average of the EC rates of HAHap is 0.0628, and that of WhatsHap is 0.0633. In contrast, WhatsHap surpassed HAHap in 153 cases, where the weighted average EC rate of HAHap is 0.0460, and that of WhatsHap is 0.0451 (Table 5). In the comparison to HapCut2 (Figure 13), HAHap outperformed HapCut2 in 300 cases, where the weighted average EC rate is 0.0761, and that of HapCut2 is 0.0680. HapCut2 surpassed HAHap in 175 cases, where the weighted average EC rate of HAHap is 0.0636, and that of HapCut2 is 0.0592 (Table 6).

In conclusion, HAHap achieves a competitive level of ECs compared to the other methods we evaluated, even though it adopted a different approach rather than minimizing the number of ECs. Although HapCut2 performed better regarding ECs, HAHap was superior in most cases, which indicates that a lower rate of EC does not guarantee better performance in terms of phasing error rates.
3.6 Application on ABO blood type detection.

We cooperated with Taipei Blood Center, Taiwan Blood Services Foundation to demonstrate the possibility of using genotyping and haplotyping to assist traditional blood typing with serology. The ABO blood group system is used to denote the presence of the A and B antigens on erythrocytes. The ABO blood group can be characterized into four main types: (1) only A antigen presents, (2) only B antigen present, (3) both presents, and (4) both not present. These four types are classified as group A, group B, group AB, and group O. However, the blood subtypes are much more complicated. Until today, there are more than three hundreds of distinct ABO blood subtypes discovered. In human blood transfusions, a mismatch on the ABO blood type between the donor and the recipient could cause a serious adverse reaction and may lead to fatality. Traditionally, the serology, through blood testing, is the standard method to determine the ABO blood group, but it has limitations with sensitivity, and manual testing may be biased to each personnel's judgment.

We chose 12 samples from a previous study (Wu et al. 2018) to demonstrate how variant phasing can help. Five of the 12 samples belonged to the normal ABO type and served as controls here, and the remaining seven samples are unknown blood types. For these unknown samples, medical technologists found discrepancy using ABO serology and suspected them to be ABO subtypes. Experts in the Taipei Blood Center manually determined the subtypes for these seven samples. With HAHap, we were able to find the variations, defining ABO subtypes, and to determine the cis/trans association to the $A/B/O$ alleles without the help of the experts. Through the well-studied relations between ABO subtypes and the $ABO$ genomic sequences, accurate haplotypes provided great enhancement when compared to the original blood type testing accuracy and therefore improved the safety of blood transfusion. Among these 12 samples, the subtypes can be determined by using the haplotypes of the 15 variants plus one additional variant for the specific subtype.
These variants are located at exon 6 and exon 7 of the ABO gene. In the end, HAHap predicted both normal and subtype samples correctly. We provided the predictions in Table S4. Although only a small number of variants were included in determining the ABO subtypes, these examples revealed the value of conducting haplotyping using short reads in many precision medicine applications in the future.

3.7 Running time

We measured the running time of the three tools on the NovoAlign-generated BAM files of the Ashkenazim trio, which were used in the real data evaluation. All experiments were performed on a 48-core machine with Intel Xeon E5-2683 CPUs and 384 GB of memory. HAHap took about three hours to complete whole-genome phasing, which was slower than HapCut2, but much faster than WhatsHap (Table 7).

4 Discussion

This study presents a read-based heuristic method using hierarchical assembly to build haplotypes. HAHap uses a CS to rank the pairs of heterozygous variants. The idea is that, for diploid organisms, all reads come from two chromosomes, and the counts of observations on the two haplotypes tend to be close to each other. The CS follows the diploid assumption using a multinomial distribution and takes a relatively low likelihood outcome to represent sequencing and mapping errors. Unlike MEC-based methods that try to consider all observations, including noise, HAHap attempts to exclude the influence of the noise via hierarchical assembly.

We compared HAHap with two state-of-the-art tools: an exact MEC-based method, WhatsHap, and a heuristic MEC-based method, HapCut2. The results demonstrated that our method achieves
better accuracy in all categories of variant blocks on the real dataset of Illumina sequencing reads and simulated data. The results supported the idea that correctly removing noise has the potential to deliver better results. On a simulated dataset, we tested HAHap under different sequencing conditions. We considered whether the CSs based on the diploid assumption were stable in some extreme cases. The results revealed that our method handled high sequencing errors and was stable for a read set with coverage more than 20x. The degree of sequencing skewness encumbered HAHap the most. However, it still performed well at skewness of 30/70. A level of skewness worse than 30/70 rarely occurs. As a result, HAHap is a practical solution in most cases.

5 Conclusion

This study proposed a hierarchical assembly-based haplotyping method, HAHap, for short-read sequencing technologies. Both real and simulated data revealed the value of HAHap in handling haplotyping for large variant blocks. In the future, Moreover, it might be possible to incorporate HAHap with some other knowledge-based phasing methods like genetic and statistic phasing to obtain higher accuracy by using valuable information of related individuals.

Acknowledgements

Thank Dr. Chung-Tsai Su, Dr. Mei-Ju Chen and Mr. Yin-Hung Lin for contributing ideas in method development.
References

Aguiar D, and Istrail S. 2012. HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data. *J Comput Biol* 19:577-590.

Browning SR, and Browning BL. 2011. Haplotype phasing: existing methods and new developments. *Nat Rev Genet* 12:703-714.

Castel SE, Mohammadi P, Chung WK, Shen Y, and Lappalainen T. 2016. Rare variant phasing and haplotypic expression from RNA sequencing with phASER. *Nat Commun* 7:12817.

Chen ZZ, Deng F, and Wang L. 2013. Exact algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* 29:1938-1945.

Cheng SJ, Jiang S, Shi FY, Ding Y, and Gao G. 2018. Systematic identification and annotation of multiple-variant compound effects at transcription factor binding sites in human genome. *Journal of Genetics and Genomics* 45:373-379.

Choi Y, Chan AP, Kirkness E, Telenti A, and Schork NJ. 2018. Comparison of phasing strategies for whole human genomes. *PLoS Genet* 14:e1007308.

Edge P, Bafna V, and Bansal V. 2017. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome Res* 27:801-812.

Ellingford JM, Barton S, Bhaskar S, Williams SG, Sergouniotis PI, O'Sullivan J, Lamb JA, Perveen R, Hall G, Newman WG, Bishop PN, Roberts SA, Leach R, Tearle R, Bayliss S, Ramsden SC, Nemeth AH, and Black GC. 2016. Whole Genome Sequencing Increases Molecular Diagnostic Yield Compared with Current Diagnostic Testing for Inherited Retinal Disease. *Ophthalmology* 123:1143-1150.

Garg S, Martin M, and Marschall T. 2016. Read-based phasing of related individuals. *Bioinformatics* 32:i234-i242.

Glusman G, Cox HC, and Roach JC. 2014. Whole-genome haplotyping approaches and genomic medicine. *Genome Med* 6:73.

He D, Choi A, Pipatsrisawat K, Darwiche A, and Eskin E. 2010. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* 26:i183-190.

Lancia G ea. 2001. SNPs problems, complexity, and algorithms. The 9th Annual European Symposium on Algorithms. Berlin: Springer. p pp. 182–193.

Lippert R, Schwartz R, Lancia G, and Istrail S. 2002. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Brief Bioinform* 3:23-31.

Luukkonen TM, Mehrjouy MM, Poyhonen M, Anttonen AK, Lahermo P, Ellonen P, Paulin L, Tommerup N, Palotie A, and Varilo T. 2018. Breakpoint mapping and haplotype analysis of translocation t(1;12)(q43;q21.1) in two apparently independent families with vascular phenotypes. *Mol Genet Genomic Med* 6:56-68.
Mark J.P. Chaisson ADS, Xuefang Zhao, Ankit Malhotra, David Porubsky, Tobias Rausch, Eugene J.

Gardner, Oscar Rodriguez, Li Guo, Ryan L. Collins, Xian Fan, Jia Wen, Robert E. Handsaker, Susan

Fairley, Zev N. Kronenberg, Xiangmeng Kong, Feraydoun Hormozdiari, Dillon Lee, Aaron M.

Wenger, Alex Hastie, Danny Antaki, Peter Audano, Harrison Brand, Stuart Cantsilieris, Han Cao,

Eliza Cervera, Chong Chen, Xintong Chen, Chen-Shan Chin, Zechen Chong, Nelson T. Chuang,

Christine C. Lambert, Deanna M. Church, Laura Clarke, Andrew Farrell, Joey Flores, Timur

Galeev, David Gorkin, Madhusudan Gujral, Victor Gyurev, William Haynes Heaton, Jonas Koralch,

Sushant Kumar, Jee Young Kwon, Jong Eun Lee, Joyce Lee, Wan-Ping Lee, Sau Peng Lee, Shantao

Li, Patrick Marks, Karine Viaud-Martinez, Sascha Meiers, Katherine M. Munson, Fabio Navarro,

Bradley J. Nelson, Conor Nodzak, Amina Noor, Sofia Kyriazopoulou-Panagiotopoulou, Andy Pang,

Yunjiang Qiu, Gabriel Rosanio, Mallory Ryan, Adrian Stutz, Diana C.J. Spierings, Alistair Ward,

AnneMarie E. Welch, Ming Xiao, Wei Xu, Chengsheng Zhang, Qihui Zhu, Xiangqun Zheng-

Bradley, Ernesto Lowy, Sergei Yakneen, Steven McCarroll, Goo Jun, Li Ding, Chong Lek Koh, Bing

Ren, Paul Flicek, Ken Chen, Mark B. Gerstein, Pui-Yan Kwok, Peter M. Lansdorp, Gabor Marth,

Jonathan Sebat, Xinghua Shi, Ali Bashir, Kai Ye, Scott E. Devine, Michael Talkowski, Ryan E. Mills,

Tobias Marschall, Jan O. Korbel, Evan E. Eichler, Charles Lee. 2018. Multi-platform discovery of

haplotype-resolved structural variation in human genomes. bioRxiv.

Mazrouee S, and Wang W. 2014. FastHap: fast and accurate single individual haplotype reconstruction

using fuzzy conflict graphs. Bioinformatics 30:i371-378.

Nalls MA, Pankratz N, Lill CM, Do CB, Hernandez DG, Saad M, DeStefano AL, Kara E, Bras J, Sharma M,

Schulte C, Keller MF, Arepalli S, Letson C, Edsall C, Stefansson H, Liu X, Pliner H, Lee JH, Cheng R,

International Parkinson's Disease Genomics C, Parkinson's Study Group Parkinson's Research: The Organized GI, andMe, GenePd, NeuroGenetics Research C, Hussman Institute of Human G,

Ashkenazi Jewish Dataset I, Cohorts for H, Aging Research in Genetic E, North American Brain

Expression C, United Kingdom Brain Expression C, Greek Parkinson's Disease C, Alzheimer

Genetic Analysis G, Ikram MA, Ioannidis JP, Hadjigeorgiou GM, Bis JC, Martinez M, Perlmutter JS,

Goate A, Marder K, Fiske B, Sutherland M, Xiromerisiou G, Myers RH, Clark LN, Stefansson K,

Hardy JA, Heutink P, Chen H, Wood NW, Houlden H, Payami H, Brice A, Scott WK, Gasser T,

Bertram L, Eriksson N, Foroud T, and Singleton AB. 2014. Large-scale meta-analysis of genome-

wide association data identifies six new risk loci for Parkinson's disease. Nat Genet 46:989-993.

O'Connell J, Gurdasani D, Delaneau O, Pirastu N, Ulivi S, Cocca M, Traglia M, Huang J, Huffman JE, Rudan

I, McQuillan R, Fraser RM, Campbell H, Polasek O, Asiki G, Ekoru K, Hayward C, Wright AF, Vitart

V, Navarro P, Zagury JF, Wilson JF, Toniolo D, Gasparini P, Soranzo N, Sandhu MS, and Marchini

J. 2014. A general approach for haplotype phasing across the full spectrum of relatedness. PLoS

Genet 10:e1004234.
Pirola Y, Zaccaria S, Dondi R, Klau GW, Pisanti N, and Bonizzoni P. 2016. HapCol: accurate and memory-efficient haplotype assembly from long reads. Bioinformatics 32:1610-1617.

Porubsky D, Garg S, Sanders AD, Korbel JO, Guryev V, Lansdorp PM, and Marschall T. 2017. Dense and accurate whole-chromosome haplotyping of individual genomes. Nat Commun 8:1293.

Ripke S, O’Dushlaine C, Chamberl K, Moran JL, Kahler AK, Akterin S, Bergen SE, Collins AL, Crowley JJ, Fromer M, Kim Y, Lee SH, Magnusson PK, Sanchez N, Stahl EA, Williams S, Wray NR, Xia K, Bettella F, Borglum AD, Bulik-Sullivan BK, Cormican P, Craddock N, de Leeuw C, Durmishi N, Gill M, Golimbet V, Hamshere ML, Holmans P, Hougaard DM, Kendler KS, Lin K, Morris DW, Mors O, Mortensen PB, Neale BM, O’Neill FA, Owen MJ, Milovanovic MP, Posthuma D, Powell J, Richards AL, Riley BP, Ruderfer D, Rujescu D, Sigurdsson E, Silagadze T, Smit AB, Stefansson H, Steinberg S, Suvisaari J, Tosato S, Verhage M, Walters JT, Multicenter Genetic Studies of Schizophrenia DF, Levinson DF, Gejman PV, Kendler KS, Laurent C, Mowry BJ, O’Donovan MC, Owen MJ, Pulver AE, Riley BP, Schwab SG, Wildenauer DB, Dudbridge F, Holmans P, Shi J, Albus M, Alexander M, Campion D, Cohen D, Dikeos D, Duan J, Eichhammer P, Godard S, Hansen M, Lerer FB, Liang KY, Maier W, Mallet J, Nertney DA, Nestadt G, Norton N, O’Neill FA, Papadimitriou GN, Ribble R, Sanders AR, Silverman JM, Walsh D, Williams NM, Wormley B, Psychosis Endophenotypes International C, Arranz MJ, Bakker S, Bender S, Bramon E, Collier D, Crespo-Facorro B, Hall J, Iyegbe C, Jablensky A, Kahn RS, Kalaydjieva L, Lawrie S, Lewis CM, Lin K, Linszen DH, Mata I, McIntosh A, Murray RM, Ophoff RA, Powell J, Rujescu D, Van Os J, Walshe M, Weisbrod M, Wiersma D, Wellcome Trust Case Control C, Donnelly P, Barroso I, Blackwell JM, Bramon E, Brown MA, Casas JP, Corvin AP, Deloukas P, Duncanson A, Jankowski J, Markus HS, Mathew CG, Palmer CN, Plomin R, Rautanen A, Sawcer SJ, Trembath RC, Viswanathan AC, Wood NW, Spencer CC, Band G, Bellenguez C, Freeman C, Hellenthal G, Giannoulatou E, Pirinen M, Pearson RD, Strange A, Su Z, Vukcevic D, Donnelly P, Langford C, Hunt SE, Edkins S, Gwilliam R, Blackburn H, Bumpstead SJ, Dronov S, Gillman M, Gray E, Hammond N, Jayakumar A, McCann OT, Liddle J, Potter SC, Ravindrarajah R, Ricketts M, Tashakkori-Ghanbaria A, Waller MJ, Weston P, Widaa S, Whittaker P, Barroso I, Deloukas P, Mathew CG, Blackwell JM, Brown MA, Corvin AP, McCarthy MJ, Spencer CC, Bramon E, Corvin AP, O’Donovan MC, Stefansson K, Scolnick E, Purcell S, McCarthy SA, Sklar P, Hultman CM, and Sullivan PF. 2013. Genome-wide association analysis identifies 13 new risk loci for schizophrenia. Nat Genet 45:1150-1159.

Sankararaman S, Mallick S, Dannemann M, Prufer K, Kelso J, Paabo S, Patterson N, and Reich D. 2014. The genomic landscape of Neanderthal ancestry in present-day humans. Nature 507:354-357.

Schiffels S, and Durbin R. 2014. Inferring human population size and separation history from multiple genome sequences. Nat Genet 46:919-925.

Sedlazeck FJ, Lee H, Darby CA, and Schatz MC. 2018. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. Nat Rev Genet 19:329-346.
Snyder MW, Adey A, Kitzman JO, and Shendure J. 2015. Haplotype-resolved genome sequencing: experimental methods and applications. *Nat Rev Genet* 16:344-358.

Sousa-Pinto B, Correia C, Gomes L, Gil-Mata S, Araujo L, Correia O, and Delgado L. 2016. HLA and Delayed Drug-Induced Hypersensitivity. *International Archives of Allergy and Immunology* 170:163-179.

Stavropoulos DJ, Merico D, Jobling R, Bowdin S, Monfared N, Thiruvahindrapuram B, Nalpathamkal T, Pellecchia G, Yuen RKC, Szego MJ, Hayeems RZ, Shaul RZ, Budno M, Girdea M, Frey B, Alipanahi B, Ahmed S, Babul-Hirji R, Porras RB, Carter MT, Chad L, Chaudhry A, Chitayat D, Doust SJ, Cytrynbaum C, Dupuis L, Ejaz R, Fishman L, Guerin A, Hashemi B, Helal M, Hewson S, Inbar-Feigenberg M, Kannu P, Karp N, Kim R, Kronick J, Liston E, MacDonald H, Mercimek-Mahmutoglu S, Mendoza-Londono R, Nasr E, Nimmo G, Parkinson N, Quercia N, Raiman J, Roifman M, Schulze A, Shugar A, Shuman C, Sinajon P, Siriwardena K, Weksberg R, Yoon G, Carew C, Erickson R, Leach RA, Klein R, Ray PN, Meyn MS, Scherer SW, Cohn RD, and Marshall CR. 2016. Whole Genome Sequencing Expands Diagnostic Utility and Improves Clinical Management in Pediatric Medicine. *NPJ Genom Med* 1.

Wu PC, Lin YH, Tsai LF, Chen MH, Chen PL, and Pai SC. 2018. ABO genotyping with next-generation sequencing to resolve heterogeneity in donors with serology discrepancies. *Transfusion*.

Xie M, Wang J, and Chen X. 2015. LGH: A Fast and Accurate Algorithm for Single Individual Haplotyping Based on a Two-Locus Linkage Graph. *IEEE/ACM Trans Comput Biol Bioinform* 12:1255-1266.

Zanger UM, and Schwab M. 2013. Cytochrome P450 enzymes in drug metabolism: regulation of gene expression, enzyme activities, and impact of genetic variation. *Pharmacol Ther* 138:103-141.

Zheng GX, Lau BT, Schnall-Levin M, Jarosz M, Bell JM, Hindson CM, Kyriazopoulou-Panagiotopoulou S, Masquelier DA, Merrill L, Terry JM, Muvdivarti PA, Wyatt PW, Bharadwaj R, Makarewicz AJ, Li Y, Belgrader P, Price AD, Lowe AJ, Marks P, Vurens GM, Hardenbol P, Montesclaros L, Luo M, Greenfield L, Wong A, Birch DE, Short SW, Bjornson KP, Patel P, Hopmans ES, Wood C, Kaur S, Lockwood GK, Stafford D, Delaney JP, Wu I, Ordenez HS, Grimes SM, Greer S, Lee JY, Belhocine K, Giorda KM, Heaton WH, McDermott GP, Bent ZW, Meschi F, Kondov NO, Wilson R, Bernate JA, Gauby S, Kindwall A, Bermejo C, Fehr AN, Chan A, Saxonov S, Ness KD, Hindson BJ, and Ji HP. 2016. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol* 34:303-311.

Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, and Salit M. 2014. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat Biotechnol* 32:246-251.
Table 1 (on next page)

Evaluation results of all the cases in real datasets
Table 1. Evaluation results of all the cases in real datasets

| Method    | Perfect Ratio (%) | Phasing error rate (%) |
|-----------|-------------------|------------------------|
| HAHap     | 99.39 (1,277,260/1,285,153) | 0.2293 (8,646 / 3,771,159) |
| WhatsHap  | 99.15 (1,272,347/1,283,302) | 0.3169 (10,955 / 3,774,198) |
| HapCut2   | 99.35 (1,276,857/1,285,092) | 0.2503 (9,440 / 3,771,022) |
Table 2 (on next page)

Evaluation results of the cases with size > 20 in real datasets
### Table 2. Evaluation results of the cases with size > 20 in real datasets

| Method   | Perfect Ratio (%) | Phasing error rate (%) |
|----------|-------------------|------------------------|
| HAHap    | 95.36 (12,191/12,784) | 0.2283 (1,056 / 462,542) |
| WhatsHap | 91.14 (11,712/12,851) | 0.5105 (2,405 / 471,133) |
| HapCut2  | 94.87 (12,125/12,780) | 0.2992 (1,384 / 462,564) |
Table 3 (on next page)

Evaluation on QAN50 in real datasets
Table 3. Evaluation on QAN50 in real datasets

| Method   | All cases | The cases with size > 20 |
|----------|-----------|--------------------------|
| HAHap    | 858.86    | 4,295                    |
| WhatsHap | 864       | 4,367                    |
| HapCut2  | 858       | 4,275                    |
Table 4 (on next page)

Evaluation of 45 distinct configurations
Table 4. Evaluation of 45 distinct configurations

| Coverage | S-e | S-skew | P-e (%) | S-skew | P-e (%) |
|----------|-----|--------|---------|--------|---------|
| 20       | 0.002 | 50/50   | 1.2798  | 20/80  | 1.0249  |
|          |       | 40/60   | 1.2289  | 10/90  | 1.8478  |
|          |       | 30/70   | 1.6000  |        |         |
| 0.01     |       | 50/50   | 1.2414  | 20/80  | 3.2312  |
|          |       | 40/60   | 1.2223  | 10/90  | 14.7022 |
|          |       | 30/70   | 0.8636  |        |         |
| 0.03     |       | 50/50   | 1.2887  | 20/80  | 12.1024 |
|          |       | 40/60   | 2.3514  | 10/90  | 92.9313 |
|          |       | 30/70   | 2.6748  |        |         |
| 30       | 0.002 | 50/50   | 0.9466  | 20/80  | 1.4463  |
|          |       | 40/60   | 1.1406  | 10/90  | 1.5991  |
|          |       | 30/70   | 1.1039  |        |         |
| 0.01     |       | 50/50   | 1.0248  | 20/80  | 1.7626  |
|          |       | 40/60   | 1.2779  | 10/90  | 12.2350 |
|          |       | 30/70   | 0.8631  |        |         |
| 0.03     |       | 50/50   | 1.1988  | 20/80  | 4.1918  |
|          |       | 40/60   | 0.9830  | 10/90  | 62.1470 |
|          |       | 30/70   | 2.3778  |        |         |
| 40       | 0.002 | 50/50   | 0.9885  | 20/80  | 1.1197  |
|          |       | 40/60   | 1.4362  | 10/90  | 1.2762  |
|          |       | 30/70   | 0.9022  |        |         |
| 0.01     |       | 50/50   | 1.3188  | 20/80  | 1.3169  |
|          |       | 40/60   | 0.6999  | 10/90  | 8.2558  |
|          |       | 30/70   | 0.8605  |        |         |
| 0.03     |       | 50/50   | 1.4048  | 20/80  | 2.6704  |
|          |       | 40/60   | 1.4719  | 10/90  | 36.6091 |
|          |       | 30/70   | 1.8134  |        |         |

S-e: sequencing error; S-skew: sequencing skewness; P-e: phasing error rate (%% = 1/10000)
Table 5 (on next page)

Comparison of weighted ECs rate with WhatsHap
Table 5. Comparison of weighted ECs rate with WhatsHap

|                        | HAHap performs better (547 cases) | WhatsHap performs better (153 cases) |
|------------------------|-----------------------------------|-------------------------------------|
| HAHap                  | 0.0628                            | 0.0460                              |
| WhatsHap               | 0.0633                            | 0.0451                              |
Table 6 (on next page)

Comparison of weighted ECs rate with HapCut2
**Table 6. Comparison of weighted ECs rate with HapCut2**

|                  | HAHap performs better (300 cases) | HapCut2 performs better (175 cases) |
|------------------|-----------------------------------|-------------------------------------|
| HAHap            | 0.0761                            | 0.0636                              |
| HapCut2          | 0.0680                            | 0.0592                              |
Table 7 (on next page)

Comparison of running time
## Table 7. Comparison of running time

| App     | NA24385   | NA24149   | NA24143   |
|---------|-----------|-----------|-----------|
| HAHap   | 160m13s   | 182m54s   | 195m58s   |
| WhatsHap| 463m58s   | 481m50s   | 531m20s   |
| HapCut2 | 52m50s    | 57m50s    | 67m48s    |
Figure 1

Haplotyping using hierarchical assembly versus dynamic programming

(a) Indicates the proposed heuristic-based process using hierarchical assembly, and (b) shows the MEC-based search process using dynamic programming. \( F_j \) is a fragment (read pair), and \( V_i \) is a heterozygous variant. Given that the gray cell is a sequencing error, the proposed hierarchical approach will not be affected when phasing other positions, but the MEC approach will consistently consider it during the whole process, and this might damage the solution if too many noises exist.
Full phased haplotype

Partial phased

3 variants

2 variants

(a) Heuristic-based (Hierarchical Assembly)

(b) MEC-based (Dynamic Programming)

... \( v_{i-1} \) | \( v_i \) | \( v_{i+1} \) | \( v_{i+2} \) | \( v_{i+3} \) ...

| \( F_j \) | 1 | 1 | 0 | 0 |
| \( F_{j+1} \) | 0 | 0 | 1 |
| \( F_{j+2} \) | **1** | **0** | 1 |
| \( F_{j+3} \) | 1 | 1 | 0 |
| \( F_{j+4} \) | 1 | 1 | 0 |
| \( F_{j+5} \) | 0 | 0 | 1 |
Figure 2

HAHap uses the proposed CS score to guide the haplotyping process

(a). Merging Cluster1 and Cluster2 by variants V\textsubscript{3} and V\textsubscript{4}: There are two variants in cluster1 and three in cluster2. HAHap chooses the pair with the highest CS score to lead the process (the 3-4 pair in this example). HAHap records the read counts (RC) of four possible combinations into the read table; (b). Calculating the scores for the two preferred solutions (S1 and S2) for variants V\textsubscript{3} and V\textsubscript{4}: We used the notation [0-0, 1-1] to denote the first prediction (S1), which is [Cluster1\_H1-Cluster2\_H1, Cluster1\_H2-Cluster2\_H2]. Similarly, we used the notation [0-1, 1-0] to denote the second prediction (S2), which is [Cluster1\_H1-Cluster2\_H2, Cluster1\_H2-Cluster2\_H1]. In the end, the one with the higher score (CS\textsubscript{1}) will be chosen as the prediction.
Figure 3

An example of the proposed hierarchical assembly

$F_j$ is a fragment (read pair), $V_i$ is a heterozygous variant and CS for confidence score. (a) Initial state; (b) Merging variants 1 and 3 and updating the score matrix by single-linkage; (c) Merging cluster [1-3] and variant 4; (d) Local MEC-search: calculating the number of error corrections for the two preferred solutions. For each sub figures, the lower panel shows the resolved haplotypes.
Figure 4

An example to demonstrate the difference between HAHap and MEC-based methods

This example contains three variants spanned by fifty reads, including five types of reads ($f_1$~$f_5$). (a) The MEC method finds a solution with twenty corrections (purple cells) to produce a conflict-free matrix. (b) HAHap selects to phase V1 and V2 first according to the proposed CS scores. There are two preferred solutions: red and black rectangles. Because the sampling ratio is close to 1:1 in the red solution, the CS of the red solution is higher than CS of the black solution. Therefore, it phases the segment as (G-A, A-T). Next, HAHap extends the phasing process to V3 by the same rule. In the end, two methods predict different haplotypes. The MEC method assumes reads in categories 4 and 5 are errors, but HAHap suggests reads in categories 1, 2 and 3 are incorrect.
Figure 5

Comparison on perfect ratios
Figure 6

Comparison on phasing error rates
Figure 7

Comparison of HAHap with WhatsHap in blocks with a size $\geq 20$
Figure 8

Comparison of HAHap with HapCut2 in blocks with a size $\geq 20$
Figure 9

Comparison on phasing error rates using the simulated data under different conditions

(condition notation: ‘sequencing error rate’ _‘read coverage’)
Figure 10

Comparison on phasing error rates using the simulated data for different block sizes
Figure 11

Comparison of phasing error rates under different settings of skewness

(Condition notation: ‘sequencing error rate’ ‘read coverage’)
Figure 12

Error correction (EC) rates of HAHap versus that of WhatsHap on 700 cases of block size \(\geq 20\) where two tools have different predictions.
Figure 13

Error correction (EC) rates of HAHap versus that of HapCut2 on 475 cases of block size $\geq 20$ where two tools have different predictions.