BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 19 2013, pages 2427–2434
doi:10.1093/bioinformatics/btt418

Genetics and population analysis

Advance Access publication August 13, 2013

Joint haplotype phasing and genotype calling of multiple individuals using haplotype informative reads

Kui Zhang* and Degui Zhi*

Section on Statistical Genetics, Department of Biostatistics, University of Alabama at Birmingham, Birmingham, AL 35294, USA

Associate Editor: Gunnar Ratsch

ABSTRACT

Motivation: Hidden Markov model, based on Li and Stephens model that takes into account chromosome sharing of multiple individuals, results in mainstream haplotype phasing algorithms for genotyping arrays and next-generation sequencing (NGS) data. However, existing methods based on this model assume that the allele count data are independently observed at individual sites and do not consider haplotype informative reads, i.e. reads that cover multiple heterozygous sites, which carry useful haplotype information. In our previous work, we developed a new hidden Markov model to incorporate a two-site joint emission term that captures the haplotype information across two adjacent sites. Although our model improves the accuracy of genotype calling and haplotype phasing, haplotype information in reads covering non-adjacent sites and/or more than two adjacent sites is not used because of the severe computational burden.

Results: We develop a new probabilistic model for genotype calling and haplotype phasing from NGS data that incorporates haplotype information of multiple adjacent and/or non-adjacent sites covered by a read over an arbitrary distance. We develop a new hybrid Markov Chain Monte Carlo algorithm that combines the Gibbs sampling algorithm of HapSeq and Metropolis–Hastings algorithm and is computationally feasible. We show by simulation and real data from the 1000 Genomes Project that our model offers superior performance for haplotype phasing and genotype calling for population NGS data over existing methods.

Availability: HapSeq2 is available at www.ssg.uab.edu/hapseq/.

Contact: dzhi@uab.edu or kzhang@uab.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on January 10, 2013; revised on June 10, 2013; accepted on July 15, 2013

1 INTRODUCTION

Low-coverage sequencing of multiple samples is an efficient strategy to profile genetic variations in a population (Li *et al.*, 2011) because low-sequencing depth makes it affordable to sequence a larger number of samples. The high accuracy of genotype calling and haplotype phasing of such a design strategy is achieved by many innovative developments in the field of bioinformatics and statistical genetics. As demonstrated by the 1000 Genomes Project (Abecasis *et al.*, 2012; Durbin *et al.*, 2010), for common variants, the accuracy of genotype calling from low-coverage sequencing is comparable with that from genotyping arrays.

Although haplotype phasing was not the primary goal of the 1000 Genomes Project, the linkage disequilibrium (LD)-based refinement method, Thunder (Li *et al.*, 2011), used in the project to refine genotype calls from individual subgroups, also phases genotypes into haplotypes.

Thunder, and most LD-based genotype refinement algorithms, is based on the Li and Stephens model (Li and Stephens, 2003). This model takes into account chromosome sharing among multiple individuals, and can be efficiently optimized by hidden Markov model (HMM)-based algorithms. This model was traditionally applied to array-based genotype data in which, at each site, it only observed the unphased genotype. HMM-based methods can phase the haplotypes of multiple individuals simultaneously through a population genetics model that models the chromosome sharing among these individuals. With a modification of single-site emission probability, Thunder extended this approach to phase population next-generation sequencing (NGS) genotype data. By comparing with the Illumina Omni 2.5 M genotyping array data that were phased by additional family samples, Thunder was reported to make one switch error in about every 300–400 kilobytes (KB).

However, Thunder assumes that the allele count data are independently observed at each site and does not consider haplotype informative reads, i.e. reads that cover multiple heterozygous sites, which carry useful haplotype information. In our previous work (Zhi *et al.*, 2012), we developed the HMM based on the Li and Stephens model to incorporate a two-site joint emission probability that can capture the haplotype information across two adjacent sites. Our method, which is implemented in the software package HapSeq, has achieved a 9–12% reduction of error rates compared with Thunder for genotype calling of the sequencing data from the 1000 Genomes Project.

Still, haplotype information in sequencing reads was not fully used in our previous work. Haplotype information in reads that cover more than two adjacent sites is not used because of severe computational burden of higher order HMMs. This throws away valuable haplotype information, especially because newer sequencing technologies can offer longer reads. In addition, paired-end reads can cover non-adjacent sites and thus offer haplotype information over multiple adjacent and/or non-adjacent sites. Paired-end reads are routinely used in sequencing projects in aiding read mapping and assembly (Venter *et al.*, 2001). It would be important to develop advanced statistical methods that can fully use the haplotype information in reads.

*To whom correspondence should be addressed.

Notably, a probabilistic haplotype phasing model, HASH, was proposed by Bansal *et al.* (2008) for a single individual using whole-genome sequencing reads. Their approach was termed 'haplotype assembly' because of its resemblance to the traditional fragment assembly problem. Traditional fragment assembly generates a single consensus sequence out of a set of reads from a single individual, ignoring the diploid nature of the human genome. Bansal *et al.*'s haplotype assembly was to generate a pair of consensus sequences out of a set of reads from a diploid individual. Their model is based on the haplotype likelihood of sequencing reads—the probability of a haplotype pair given the sequencing reads. With the assumption of the uniform prior on the space of haplotypes, this probability is proportional to the probability of reads given the haplotype pair. A Metropolis–Hastings (MH) algorithm was proposed to sample haplotype pairs in which 'moves' are flipping of substrings of the haplotypes. They estimated the switch error rate of haplotypes inferred for a genome (Levy *et al.*, 2007) sequenced by 7.5X Sanger reads was ~1.1%. However, their method assumes that genotypes are readily known, and it requires high-sequencing coverage; thus, it is not applicable to low-coverage sequencing in which read counts are sparse and joint genotype calling and haplotype phasing are essential for high accuracy. In addition, with the assumption of the uniform prior, their method ignores the haplotype information contained by other individuals and/or reference haplotypes.

Another notable work is by He *et al.* (2012). Their Hap-seq (not our program 'HapSeq') method extended the haplotype assembly approach by incorporating population information. Their haplotype likelihood was divided into two independent parts: the probability of sequencing of reads given the haplotype pair, which is similar to that used in HASH (Bansal *et al.*, 2008), and the probability of the haplotype pair given the set of reference haplotypes, which can be calculated using the HMM similar to the HMM in Thunder/HapSeq. He *et al.* (2012) designed a dynamic programming algorithm to find a haplotype pair to maximize the haplotype likelihood. Their simulation results showed that the haplotype inferred from such model had lower switch error rates than those obtained from IMPUTE v1.0 (Marchini *et al.*, 2007). The method of He *et al.* (2012) can be used for low-coverage sequencing but still assumes that the genotypes at each site are already known. Essentially, they extended the Li and Stephens model into higher-order Markov models, and thus their method incurs a computational complexity of $O(4^V)$, for just running a Viterbi pass for phasing one individual, where V is the maximum number of sites spanned by reads. Their approach is impractical if V is large. Unfortunately, paired-end reads are commonly used in real sequencing projects, as such reads generally span a large number of sites. To avoid this potential problem, He *et al.* (2012) had to split a long read to the multiple reads that each span only three heterozygote sites. Obviously, this approach is not optimal, as it does not fully use the haplotype information of reads that cover a large number of sites.

In this work, we develop a fully probabilistic model for joint genotype calling and haplotype phasing that incorporates the joint distribution of two or more sites covered by a read over an arbitrary distance. Our model integrates elements of the population-haplotype likelihood in Thunder/HapSeq and the read-haplotype likelihood in HASH (Bansal *et al.*, 2008), each capturing complementary haplotype information. Because both methods are Markov Chain Monte Carlo (MCMC)-based, we develop a combined MCMC method that embeds a MH procedure into a Gibbs sampling algorithm. Specifically, in each iteration, we first use the Thunder/HapSeq HMM to jointly perform genotype calling and haplotype phasing, and then use the MH algorithm to sample haplotypes of each individual according to the likelihood based on sequencing reads and reference haplotypes. Our method is implemented in the HapSeq2 program, and is evaluated together with Thunder and HapSeq by using simulation and real data.

2 METHODS

2.1 Overview

Our probabilistic model incorporates both the likelihood of the multi-sample chromosome sharing and the likelihood of haplotypes given multisite-spanning reads. Although our model is similar to that of He *et al.* (2012), our algorithm is an MCMC sampling algorithm that is practically efficient. Basically, our MCMC sampling method is a hybrid of the HMM-based Gibbs sampling-like algorithm as in Thunder/HapSeq, and the MH algorithm as in HASH. The overall algorithm is a Gibbs sampling-type algorithm that updates each individual in turn and runs for a number of iterations. For each individual in each iteration, we refine its genotypes and haplotypes in two steps. First, we perform the Thunder/HapSeq HMM sampling to obtain the genotypes and haplotypes. After that, with the assumption of fixed genotypes for that individual, we perform the MH sampling to update the haplotype pair of that individual. The efficiency of our algorithm comes from the idea that we identify three kinds of genotype and haplotype information in reads (as detailed in Section 2.2) and only use parts of the information in reads that are suited in each step. In the first step, we only use reads covering single sites and covering two adjacent sites as in Thunder/HapSeq, and intentionally ignore the haplotype information in multisite-spanning reads. In the second step, the sampling procedure is according to the sequencing reads that cover two or more heterozygote sites (adjacent and/or non-adjacent) and the reference haplotypes. In summary, our procedure can be described as the following.

Initialization: Assign genotypes and haplotypes of each individual according to the sequencing reads;

Outer Iteration: For $t = 1, 2, \cdots$, perform the inner iteration;

Inner Iteration: For each individual $n = 1, 2, \cdots, N$:

(1) Perform the HMM sampling to obtain genotypes and haplotypes,

(2) Perform the MH to update haplotypes;

Finalization: Construct the consensus haplotypes and genotypes of each individual based on haplotypes and genotypes obtained from each outer iteration. The detailed descriptions of each step are given in subsequent sections.

2.2 Genotype and haplotype information in reads

For a typical NGS shotgun sequencing project, we observe reads covering L polymorphic sites for N individuals. Throughout, we assume biallelic sites, with alleles labeled as 0 (the reference allele) and 1 (the alternative alleles). Based on the haplotype information contained, we identify three sets of read information, R_1, R_2 and R_3, according to the following rules: (i) a read that covers a single site belongs to R_1; (ii) a read that covers two

adjacent sites belongs to R_2 and R_3; (iii) for a read that covers three or more adjacent sites, we do the following: first, we put every two adjacent sites to R_2, second, if there is a site left (i.e. the read covers an odd number of adjacent sites), we put the last site to R_1, third, we put the entire read to R_3; (iv) for a read containing non-adjacent sites, e.g. a read pair, we first do step (iii) for each chunk of the consecutive sites covered by the read, and then put the entire read to R_3. In HapSeq, we use two types of reads, R_1 and R_2, where R_3 is broken into R_1 and/or R_2. In this work, we use R_1 and R_2 in the HMM sampling and R_3 in the MH sampling. It is worth noting the classification of a read to R_1, R_2 and R_3 is based on the number of sites, and not the number of heterozygote sites that it covers. Such classification is only performed once with the fixed set of variant sites at the beginning of algorithm; thus it is not changed according to genotypes or haplotypes. We denote $R_{1,\{l,n\}}$ as the set of reads that cover the single site l for the individual n, $R_{2,\{l,n\}}$ as the set of reads that cover two adjacent site l-1 and l for the individual n and $R_{3,\{l,n\}}$ as the set of reads that end at site l and cover two non-adjacent sites and/or three or more sites for individual n.

2.3 The HMM for the whole-genome shotgun sequencing data in Thunder/HapSeq

We denote the set of reference haplotypes as T, and the number of reference haplotypes as $|T|$. For genotype calling and haplotype phasing, both external haplotypes (e.g. haplotypes obtained from the external reference data such as data from the HapMap Project or the 1000 Genomes Project) and/or internal haplotypes (haplotypes estimated from sequenced individuals in the same study) can be used as reference haplotypes (Li *et al.*, 2010; Marchini *et al.*, 2007). For NGS data, external reference haplotypes are often incomplete and/or unavailable. As a result, Thunder and HapSeq often use internal reference haplotypes only. For N individuals, the number of internal reference haplotypes is 2*(N-1) (and the reference haplotypes themselves are different across different individuals). For the rest of manuscript, we will ignore the individual index subscript n for the sake of simplifying notations. This is fine because our overall algorithm runs in a Gibbs sampler fashion and iteratively infers the genotypes and haplotypes for each individual given the reference haplotypes.

The HMM is the same as the HMM in HapSeq (Zhi *et al.*, 2012), which can be described as following:

$$P(R_1, R_2, S) = P(S_1)\prod_{l=2}^{L} P(S_l \mid S_{l-1})\prod_{l=1}^{L} P(R_{1,l} \mid S_l)\prod_{l=2}^{L} P(R_{2,l} \mid S_{l-1}, S_l) \tag{1}$$

In formula (1), we use a series of indicator variables $S_l (l = 1, \cdots, L)$ to represent a hypothetical (and unobserved) state sequence for that individual, indicating to which reference haplotypes that individual is closest at the site l. At a specific site l, a diploid state $S_l = (x_l, y_l)(l = 1, \cdots, L)$ indicates that the two haplotypes of the individual are x_l and y_l out of the $|T|$ reference haplotypes, respectively. In addition, $P(S_1)$ denotes the prior probability of the initial mosaic state and is usually assumed to be equal for all possible compatible haplotype configurations of each individual; $P(S_l \mid S_{l-1})$ denotes the transition probability between two mosaic states and reflects the likelihood of historical recombination events between the sites l and l-1; $P(R_{1,l} \mid S_l)$ denotes the emission probability, which is the probability of observed R_1 reads that cover the site l conditioning on the underlying mosaic state at the site l; $P(R_{2,l} \mid S_{l-1}, S_l)$ denotes the emission probability, which is the probability of R_2 reads that cover two adjacent sites (l-1 and l) conditioning on the underlying mosaic state at sites l-1and l. Note that the emission probability $P(R_{2,l} \mid S_{l-1}, S_l)$ at site l not only depends on S_l but also depends on S_{l+1} because $R_{2,l}$ actually reflects the haplotype information between the sites l-1 and l.

The emission probability, $P(R_{2,l} \mid S_{l-1}, S_l)$, is based on two haplotypes h_{1l} and h_{2l} defined by S_{l-1} and S_l across the sites $l-1$ and l, and the error parameter δ, which is the per base sequencing error rate. We further denote $R_{2,l} = (n_{00,l}, n_{01,l}, n_{10,l}, n_{11,l})$ as the number of combinations of alleles 0 and 1 across the sites $l-1$ and l that are simultaneously observed in the R_2 reads. Then $P(R_{2,l} \mid S_{l-1}, S_l)$ is defined to follow a multinominal distribution (Zhi *et al.*, 2012):

$$\begin{aligned} P(R_{2,l} &= (n_{00,l}, n_{01,l}, n_{10,l}, n_{11,l}) \mid S_{l-1}, S_l)) \\ &= P(R_{2,l} = (n_{00,l}, n_{01,l}, n_{10,l}, n_{11,l}) \mid (h_{1l}, h_{2l})) \\ &\propto P(00 \mid (h_{1l}, h_{2l}))^{n_{00,l}} P(01 \mid (h_{1l}, h_{2l}))^{n_{01,l}} \\ &{}^{*}P(10 \mid (h_{1l}, h_{2l}))^{n_{10,l}} P(11 \mid (h_{1l}, h_{2l}))^{n_{11,l}} \end{aligned} \tag{2}$$

where

$$P(00 \mid (h_{1l}, h_{2l})) = 0.5 * P(00 \mid h_{1l}) + 0.5 * P(00 \mid h_{2l}) \tag{3}$$

and

$$P(00 \mid h) = \begin{cases} (1-\delta)^2, & \text{if two hapolotypes are identical } (h = 00), \\ \delta(1-\delta), & \text{if two haplotypes differs at one site } (h = 01), \\ \delta^2, & \text{if two haplotypes differes at both sites } (h = 11). \end{cases} \tag{4}$$

$P(01 \mid (h_{1l}, h_{2l}))$, $P(10 \mid (h_{1l}, h_{2l}))$ and $P(11 \mid (h_{1l}, h_{2l}))$ can be defined similarly.

Once the prior probability $[P(S_1)]$, the transition probability $[P(S_l \mid S_{l-1})]$ and the emission probability $[P(R_{1,l} \mid S_l)$ and $P(R_{2,l} \mid S_{l-1}, S_l)]$ in formula (1) are defined, we can use the standard HMM Monte-Carlo procedure to sample $S_1, \cdots, S_L$, impute the genotype and determine the haplotype pair of each individual over a number of iterations. The detailed description of all terms in the HMM and the sampling procedure can be found in Zhi *et al.* (2012).

Theoretically, we can incorporate the R_3 reads into the HMM:

$$\begin{aligned} P(R_1, R_2, R_3, S) = P(S_1)\prod_{l=2}^{L} P(S_l \mid S_{l-1})\prod_{l=1}^{L} P(R_{1,l} \mid S_l) \\ \prod_{l=2}^{L} P(R_{2,l} \mid S_{l-1}, S_l)\prod_{l=3}^{L} P(R_{3,l} \mid S_1, \cdots, S_{l-1}, S_l) \end{aligned} \tag{5}$$

It can be seen that the emission probability $P(R_{3,l} \mid S_1, \cdots, S_{l-1}, S_l)$ depends on not only S_l and S_{l-1} if $R_{3,l}$ covers sites l and l-1 and some sites from 1 to l-2. This greatly increases the computational complexity when we perform the forward probability calculation in Monte-Carlo sampling. The computation increases rapidly with the inclusion of reads that cover more sites. Note, when we only consider the reads that cover a single site and two adjacent sites, the complexity of calculation of the forward probability is still $O(|T|^2)$. Therefore, the above pure HMM is not practical to handle R_3 reads because of the high computational complexity.

2.4 Metropolis–Hastings procedure

We denote H as a haplotype pair and $\mathrm{P}(H \mid R_3, T)$ as the haplotype likelihood H of the given sequencing reads and the set of reference haplotypes (T). We further denote $\Pr(H \to H^*)$ as the proposed probability from a haplotype pair H to a new haplotype pair H^*. Our MCMC procedure is a standard MH procedure and can be described as follows:

Initialization: Obtain H^0, the haplotype pair of that individual from the HMM procedure;

Iteration: For $t = 1, 2, \cdots$, obtain H^{t+1} from H^t as follows:

(1) Propose H^* according to H^t and the sequencing reads with the probability $\Pr(H^t \to H^*)$,

(2) With the probability $\min[1, \frac{\Pr(H^* \mid R_3, T)}{\Pr(H^t \mid R_3, T)} * \frac{\Pr(H^* \to H^t)}{\Pr(H^t \to H^*)}]$ to set $H^{t+1} = H^*$, otherwise, set $H^{t+1} = H^t$;

Finalization: Construct the consensus haplotype pair from $H^0, H^1, \cdots, H^t, \cdots$ and use it in the next step of HMM.

We would like to make the following notes: (i) For this MH procedure, we assume that the genotypes are fixed, as they are already determined from the HMM procedure so only haplotypes of that individual are updated according to the sequencing reads and haplotypes from the reference haplotypes (internal and/or external reference haplotypes). The rationale behind this is that the HMM step already can generate highly accurate genotypes. (ii) Because we assume that the genotypes are fixed, the homozygote sites covered by the R_3 reads will not affect the proposed probability [$\Pr(H^* \to H^t)$ or $\Pr(H^t \to H^*)$] or the ratio of likelihood of two haplotype pairs [$\Pr(H^* \mid R_3, T)/\Pr(H^t \mid R_3, T)$], and only the heterozygote sites need to be considered in the calculation of $\Pr(H^t \mid R_3, T)$. Here R_3 are the reads spanning two or more sites, and T is the set of reference haplotypes. It can be seen that we use both the information of haplotypes from the sequencing reads (R_3) and the LD information (T) from this set of samples or the reference samples to update the haplotypes of that individual. (iii) The updated haplotypes will be used as the reference haplotypes for other individuals in the next iteration of HMM. So we expect that the updated haplotypes (more accurate) will then improve the overall accuracy of HMM for genotype calling and haplotype phasing.

To describe our detailed algorithm of making proposal H^*, we first introduce the following notations. As we have mentioned, we only need to consider the heterozygote sites of that individual. We assume there are K heterozygote sites: $l_1, l_2, \cdots, l_K$. For the sequencing read i spanning two or more heterozygote sites (adjacent or non-adjacent), we use $Z_i = \{z_{ik}, k = l_1, \cdots, l_K\}$ to represent the observed allele of the read i at the site l_k as the reference allele (0), the alternative allele (1) or no observation (−1). Similarly, we use $H = \{h, \bar{h}\} = \{\{h_k\}, \{\bar{h}_k\}, k = l_1, l_2, \cdots, l_K\}$ to represent the haplotype pair of an individual, where $h_j = 0$ or 1 ($\bar{h}_j = 0$ or 1) represents the observed allele of that haplotype.

Given the current haplotypes of that individual, H^t, we only consider the proposal H^* that is a single crossover away from H^t. A single crossover of haplotype pair at a recombinant point refers to that two haplotypes beyond that point are swapped to form a new haplotype pair. We choose the recombination point (between two adjacent heterozygote sites) with the probability $\Pr(H^t \to H^*)$ that is proportional to a weight. We define the weight of two adjacent heterozygote sites, (l_{k-1}, l_k) as $W(l_{k-1}, l_k), k = 2, \cdots, K$ for the recombination point between heterozygote sites l_{k-1} and l_k. The weight is calculated according to H^t and H^* and the sequencing reads. Specifically, we first calculate the total number of sequencing reads that are in conflict with the current haplotype pair H^t and the proposed haplotype pair H^* and denote them as $C^t$and C^*, respectively. A conflict is claimed for a read if (i) the read spans the recombination point, i.e. covers at least one site from l_1 to l_{k-1} and one site from l_k to l_K; and (ii) the read is not compatible (identical) with either of haplotype of H^t at the heterozygote sites covered by this read. If $C^t \geq C^*$, the weight is 1 plus $C^t - C^*$, otherwise the weight is 1 (to avoid weight of 0). Once a recombinant point is chosen, the new haplotype pair H^* is just the recombinant haplotypes of H^t at the recombination point. Our procedure captures the essential element of HASH (Bansal *et al.*, 2008) that concentrates on 'flipping' a subset of the entire haplotype pair. In the same time, we restrict the allowed flipping operation to just the single crossovers so the likelihood ratio in the MH procedure can be efficiently calculated, as described below.

In the second step of the MH algorithm, we accept or reject the proposal H^* according to $\Pr(H^t \mid R_3, S)$ and $\Pr(H^* \mid R_3, S)$. To calculate $\Pr(H \mid R_3, S)$, we have:

$$\Pr(H \mid R_3, T) \propto P(H, R_3, T) = P(R_3 \mid H)P(H \mid T) \tag{6}$$

where $P(R_3 \mid H) = \prod_i P(Z_i \mid H) = \prod_i \frac{1}{2}(P(Z_i \mid h) + P(Z_i \mid \bar{h})$ is the haplotype likelihood of the sequencing reads (Bansal *et al.*, 2008; He *et al.*, 2012) and is a function of per base sequencing error and $P(H \mid T)$ is the conditional probability of the haplotype pair H given the reference haplotypes and reflects the LD information of sites. $P(H \mid T)$ can be considered as a prior distribution of a haplotype pair. In the article of Bansal *et al.* (Bansal *et al.*, 2008), the uniform prior was used. Here, $P(H \mid T)$ incorporates essential chromosomal sharing information and the HMM can be used to calculate $P(H \mid T) = P(h \mid T)P(\bar{h} \mid T)$. Again, the haplotype h is considered as the recombination events of reference haplotypes and each reference haplotype is considered as a hidden state. Therefore, the probability $P(h \mid T)$ is calculated across all possible hidden states:

$$P(h \mid T) = \sum_{S_{l_1}} \cdots \sum_{S_{l_K}} P(S_{l_1}) \prod_{k=2}^{K} P(h_{l_k} \mid S_{l_k}) P(S_{l_k} \mid S_{l_{k-1}}) \tag{7}$$

In formula (7), the prior probability $[P(S_{l_1})]$, the transition probability $[P(S_{l_k} \mid S_{l_{k-1}})]$ and the emission probability $[P(h_{l_k} \mid S_{l_k})]$ can be defined similarly as the HMM in HapSeq. Then $P(h \mid T)$ can be easily calculated by the Baum's forward algorithm. Specifically, we define the forward probability:

$$\alpha_k(i) = P(h_{l_1}, \cdots, h_{l_k}, S_{l_k} = i)(k = 1, \cdots, K; i = 1, \cdots, |T|).$$

First we calculate $\alpha_1(i)$ as: $\alpha_1(i) = P(h_{l_1}, S_{l_1} = i) = P(h_{l_1} \mid S_{l_1} = i) P(S_{l_1} = i)$. Then the following recursion is used to calculate $\alpha_{k+1}(i)$:

$$\begin{aligned}
\alpha_{k+1}(i) &= P(h_{l_1}, \cdots, h_{l_k}, h_{l_{k+1}}, S_{l_{k+1}} = i) \\
&= \sum_j P(h_{l_1}, \cdots, h_{l_k}, h_{l_{k+1}}, S_{l_k} = j, S_{l_{k+1}} = i) \\
&= \sum_j P(h_{l_{k+1}} \mid S_{l_{k+1}} = i) P(S_{l_{k+1}} = i \mid S_{l_k} = j) P(h_{l_1}, \cdots, h_{l_k}, S_{l_k} = j) \\
&= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i) \sum_j P(S_{l_{k+1}} = i \mid S_{l_k} = j) \alpha_k(j)
\end{aligned} \tag{8}$$

Finally, we can calculate $P(h \mid T) = \sum_i \alpha_K(i)$.

We would like to point out that the above computation can be greatly reduced. It can be seen that the complexity of the naïve implementation of the calculation of the forward probability is $O(K^* \mid T \mid^2)$, as we need $\mid T \mid$ (the number of reference haplotypes) additions for $\alpha_k(i)$. However, the complexity can be reduced because the transition probability, $P(S_{l_k} = i \mid S_{l_{k-1}} = j)$, only depends on if i is same with j. Specifically, $P(S_{l_k} = i \mid S_{l_{k-1}} = j) = 1 - \theta_k + \theta_k / \mid T \mid$ if $i = j$ and $P(S_{l_k} = i \mid S_{l_{k-1}} = j) = \theta_k / \mid T \mid$ if $i \neq j$, where θ_k is the recombination rate between the sites l_{k-1} and l_k. Thus, we define $\beta_k = \sum_j \alpha_k(j)$, then

$$\begin{aligned}
\alpha_{k+1}(i) &= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i) \sum_j P(S_{l_{k+1}} = i \mid S_{l_k} = i) \alpha_k(j) \\
&= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i)[P(S_{l_{k+1}} = i \mid S_{l_k} = i) \alpha_k(i) \\
&\quad + \sum_{j \neq i} P(S_{l_{k+1}} = i \mid S_{l_k} = j) \alpha_k(j)] \\
&= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i)[(1 - \theta_k + \theta_k / \mid T \mid) \alpha_k(i) + \theta_k / \mid T \mid \sum_{j \neq i} \alpha_k(j)] \\
&= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i)[(1 - \theta_k) \alpha_k(i) + \theta_k / \mid T \mid \sum_j \alpha_k(j)] \\
&= P(h_{l_{k+1}} \mid S_{l_{k+1}} = i)[(1 - \theta_k) \alpha_k(i) + \theta_k / \mid T \mid {}^* \beta_k]
\end{aligned} \tag{9}$$

The computational complexity is reduced to $O(K^* \mid T \mid)$. As the probability $P(h \mid T)$ for each haplotype is calculated independently, the computational burden of this MH procedure is actually small compared with the HMM step.

2.5 Evaluation using simulation

We used simulated sequencing data to compare the performance of Thunder, HapSeq and HapSeq2. We generated chromosomes for two populations using the cosi program (Schaffner *et al.*, 2005). We adopted the 'bestfit' model distributed with the cosi package, which takes into account the HapMap LD patterns, local recombination rates and recent human population demography. We generated 3000 chromosomes from the 'European population' (EUR) and 3000 chromosomes from the 'African population' (AFR). Each chromosome is of length 100 KB. For each population, we randomly sampled with replacement 60 sets of unrelated diploid individuals, each of $N = 60$ (120 chromosomes), to form a 'population panel'. These 60 sets are simulated with 6 scenarios, each with 10 repetitions: (i) 36 bp reads, unpaired (coded as 36-0); (ii) 100 bp reads,

unpaired (100-0); (iii) 36 bp reads, paired with 250 bp insert (36-250); (iv) 100 bp reads, paired with 250 bp insert (100-250); (v) 100 bp reads, paired with 500 bp insert (100-500); and (vi) 100 bp reads, paired with 1000 bp insert (100-1000). These settings will allow us to test the performance of these methods in different sequencing settings with various read length and insert length for paired-end reads. We fixed the rest of the simulation parameters: sequencing error rate to 0.5%, sequencing depth of coverage to 4X, as we have shown that the trends observed from different settings of these parameters tend to be similar (Zhi *et al.*, 2012). Generation of reads and site promotion follow that of HapSeq (Zhi *et al.*, 2012). Briefly, read starting positions were placed uniformly and randomly along the chromosome, and sequencing errors were generated uniformly and randomly along the length of the reads as well. For paired-end settings, the starting positions of insert fragments were placed uniformly and randomly along the chromosome, and a pair of reads from each end of the insert fragment was generated. We followed Li *et al.* (Li *et al.*, 2010) and calculated the score $w = \sum_{n=1}^{N} d_n(d_n + 1)/2$, where d_n is the minor allele count of individual n at each site. We promoted sites with $w \geq 5$ as potential polymorphic sites.

For each dataset, we run four versions of programs with 100 iterations of Gibbs sampling: (i) the original Thunder; (ii) Thunder + MH; (iii) the original HapSeq; (iv) HapSeq + MH, i.e. HapSeq2. We ran $5c$ iterations for the MH-flipping after each HMM iteration where c is the total number of heterozygote sites of the haplotypes obtained from the HMM (same as below). For Thunder, we also ran 200 and 300 iterations of Gibbs sampling. These may also improve the performance of Thunder with comparable running time as options (ii)–(iv) mentioned above. For each simulated dataset, we computed the switch error rate, genotype calling concordance rates and r^2 between the estimated and true genotypes. The averages over 10 repetitions are presented in the Results section.

2.6 Evaluation using the 1000 Genomes Project phase 1 data

To capture complexities in real sequencing data, we also evaluated these methods using the 1000 Genomes Project phase 1 data. The low-coverage read alignment data and the integrated variant calls of chromosome 20 were downloaded for 98 individuals of Utah residents with Northern and Western European ancestry from the CEPH collection (CEU) and 99 individuals of Yoruba in Ibadan of Nigeria (YRI) from http://www.1000genomes.org/. We used VCFTools v0.1.9.0 (Danecek *et al.*, 2011) to extract the polymorphic sites (but not the genotypes) in the integrated variant call sets for chromosome 20, removing any non-SNP (single nucleotide polymorphism) variant sites. We used SAMTools version 0.1.18 (Li, 2011) to merge the BAM files of the same individual, and then used an internal script to parse the BAM file of each individual to obtain the read counts (R_1), jumping reads (R_2) and read site spanning information (R_3) over these sites.

Following the 1000 Genomes Project phase 1 evaluation (Abecasis *et al.*, 2012), we used the Omni 2.5 genotype data phased by SHAPEIT (Delaneau *et al.*, 2012) as released from the 1000 Genomes Project Web site as the gold standard. Most CEU and YRI phase 1 samples are parents of father–mother–child trios, and thus the Omni data, available for all members of trios, can be phased with high accuracy. We calculated the genotype concordance and haplotype switch error results against the Omni data by using VCFTools v0.1.9.0 (Danecek *et al.*, 2011).

3 RESULTS

3.1 Simulation results

As all calculation is done over the promoted potential polymorphic sites, the following measures are relevant to LD-refinement algorithms based on haplotype-informative reads: (i) read-cover: the number of sites covered by a read; (ii) read-span: the number of sites between the first site and the last site covered by a read. The only difference between these two measures is that sites skipped by the gap between the paired-end reads are counted in read-span, but not in read-cover. For non–paired-end reads, the two statistics are the same.

Figure 1 shows, as expected, that whereas the read-cover of paired-end reads is twice as that of single reads, the read-span of paired-end reads is much longer. Noticeably, paired-end reads that span longer distance (100–500 and 100–1000) have a clear bimodal distribution of their read-span. The difference between read-span and read-cover demonstrates the information can be used by the MH haplotype flipping (MH-flipping), but not Thunder/HapSeq.

As shown in Figure 2, Thunder and HapSeq with interlaced MH-flipping result in longer switch error-free (SEF) haplotype blocks in both European and African samples. The average improvement with MH-flipping is 46.1%. Therefore, the ability of MH-flipping in capturing haplotype information over multiple sites in reads improves the haplotype phasing. If comparing HapSeq2 (HapSeq + MH) and Thunder, the average improvement across all datasets is 105.1%. Notice that running Thunder for more iterations produces slightly longer SEF haplotype blocks, but the magnitudes of improvement are not comparable with Thunder + MH and HapSeq methods, especially for datasets with paired-end reads.

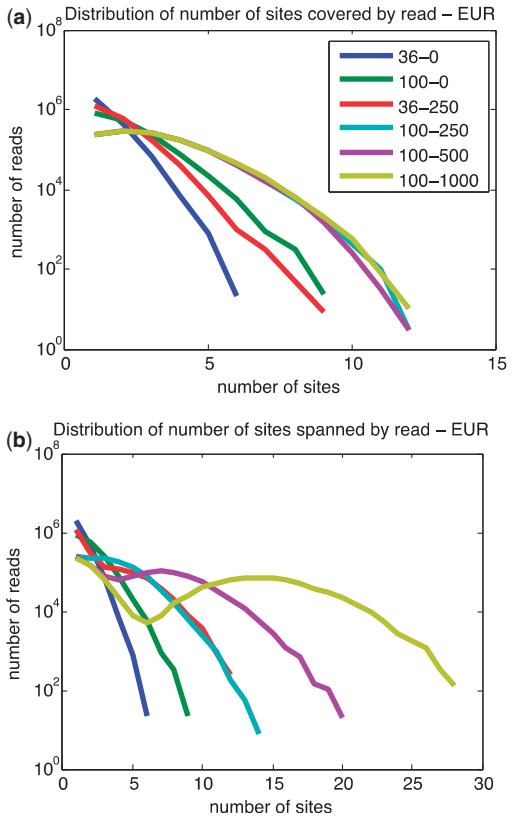

Fig. 1. Site coverage (a) and site span (b) statistics of simulated sequencing reads. Reads covering no potential polymorphic sites are not counted. The distributions for the AFR population are similar (data not shown)

In most simulated datasets, Thunder produced the shortest SEF blocks, serving as baseline performance. Both HapSeq (as Thunder + reads covering two adjacent sites) and Thunder + MH produce longer SEF blocks, although the performance of one is not always better than the other. Overall, HapSeq2 always produces the longest SEF blocks.

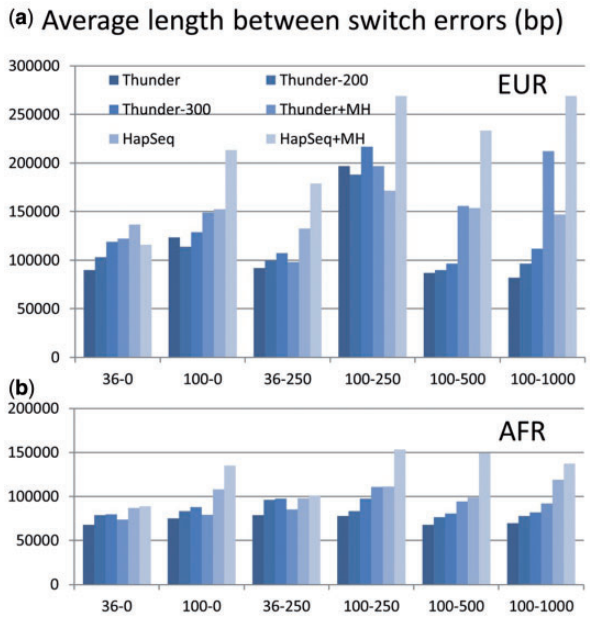

Fig. 2. Switch errors in simulated datasets: (a) EUR and (b) AFR. Six datasets for each population panel are labeled with 'x–y', where 'x' is the length of reads and 'y' is the length of inserts (0 if unpaired). Method with '+MH' refers to the original method with interlaced MH-flipping. All methods run for 100 iterations of HMM, except Thunder-200 and Thunder-300 (200 and 300 iterations, respectively)

Variations of HapSeq in general produce longer SEF blocks compared with variations of Thunder with otherwise the same settings. The average improvement is 41.4%. This is reassuring, as the improvement from adding interlaced MH-flipping is cumulative to the improvement from considering joint emission probabilities of reads covering two adjacent sites (HapSeq versus Thunder).

The results are consistent across European and African population data. Noticeably, haplotype phasing of European samples is generally better than that of African samples. This is expected, as European samples are simulated with severe bottlenecks and are known to have longer shared haplotype blocks.

We also compared different sequencing strategies from short and unpaired reads with long and paired-end reads. Throughout, Thunder produced consistent short SEF blocks, reflecting its inability to capitalize the haplotype information in sequencing reads. All other methods benefit from longer and paired-end sequencing designs. The greatest improvements between methods with MH-flipping versus those without are for 100 bp reads with 500 bp or 1000 bp inserts, with an average improvement over 100%. It is clear that 100-bp-long paired-end reads offer the best phasing results, although there are no major differences in terms of performance of HapSeq2 between these 100-bp paired-end designs.

Accurate haplotype phasing from interlaced MH-flipping is not an artifact of simply eliminating heterozygous sites. As shown in Figure 3, genotype calling error rates are lower in methods with MH-flipping than those without in both heterozygous sites (HET) and in overall sites. Looking across rare (minor allele frequency (MAF) $< 1\%$), low-frequency ($1\% \leq$ MAF $< 5\%$) and common (MAF $\geq 5\%$) single nucleotide polymorphisms, Thunder + MH offers higher genotype concordance and r^2 between the estimated genotypes and the true genotypes (Fig. 3, Supplementary Table S1). We think that the

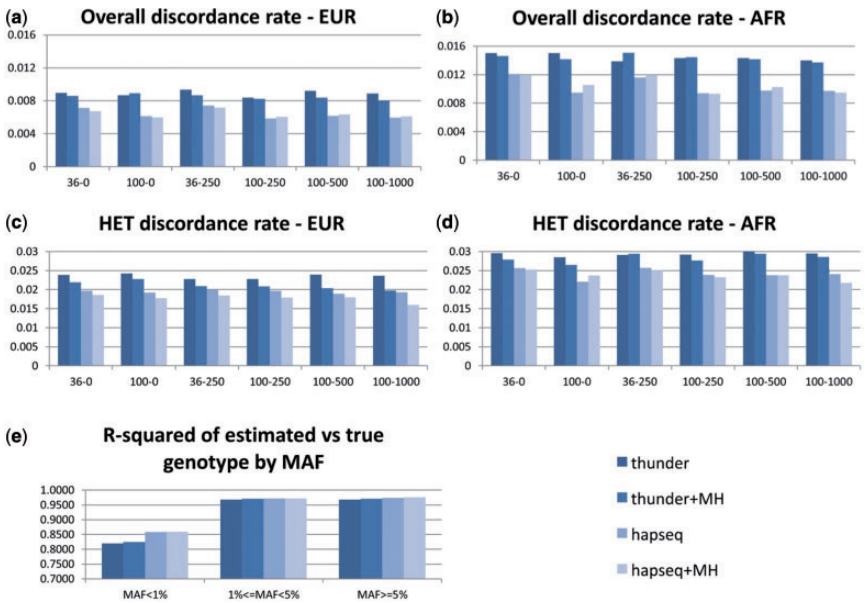

Fig. 3. Genotype calling accuracies of different methods over simulated datasets. (**a–d**) Genotype discordance rates. (**e**) r2 of estimated versus the true genotypes of European dataset 100–250. See Supplementary Table S1 for results of other simulated datasets

improvement of genotype calling is due to better haplotypes (served as internal reference haplotypes) resulted from more accurate haplotype phasing.

The interlaced MH-flipping procedure results in longer running time. The time increased is linear to the number of flips and to the number of potential MCMC moves in each round. In total, the running time for HapSeq2 is ~2–3 times that of Thunder, and is thus still practical.

3.2 Results from 1000 Genomes Project phase 1 data

Although the patterns of read-cover in the 1000 Genomes Project phase 1 data are consistent to what we observed in the simulated data, the patterns of read-span is more complex (Fig. 4). There are pairs of reads that span over 1000 sites. This is likely because of structural variations or technical artifacts. Still, bimodal patterns of read-span are observed in both CEU and YRI data. Note that CEU has a fraction of 454 reads that are longer, and thus CEU data have longer read-cover and read-span.

Over the entire chromosome 20, haplotype phasing by interlaced MH-flipping produced longer SEF-blocks, whereas genotype calling accuracy is also improved, especially for heterozygote genotypes (Fig. 5). For all pairs of methods, one with and one without interlaced MH-flipping, interlaced MH-flipping increases the average length of SEF block by 70–86 KB. This represents 23.6–44.6% improvement. Between Thunder and HapSeq2, the improvement of the length of the SEF block is 148 KB (77.6%) for CEU and 148 KB (66.7%) for YRI. All these improvements coincide with improvement of genotype calling accuracy. Interestingly, YRI has longer SEF blocks than CEU, which is different from what we observed in the simulated data. This may be due to the significantly higher heterozygote rate in the YRI samples compared with the CEU sample, which would result in a greater number of sequencing reads being useful for haplotype phasing. Another possible reason may be due to the fact that we use the same set of sites across YRI and CEU datasets in the real data analysis, whereas we used the promoted sites individually defined in each dataset in simulations.

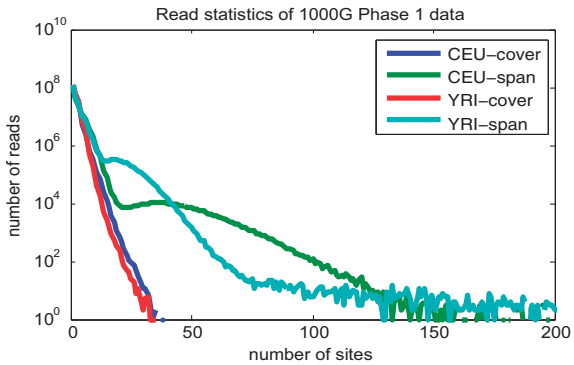

Fig. 4. Read cover and span distributions in the 1000 Genomes Project datasets

4 DISCUSSION

We developed a new approach for haplotype phasing and genotype calling from sequencing data of a set of population samples. We designed an MH-flipping algorithm that can be embedded into traditional Gibbs sampling algorithms based on the Li and Stephens HMM model. Using simulated and real datasets, we showed that our new method can greatly improve the accuracy of haplotype phasing over current state-of-the-art methods. In the 1000 Genomes Project phase 1 data, our HapSeq2 method produces 60–80% longer SEF haplotype blocks than Thunder. Although the primary goal of introducing the MH-flipping procedure is to improve haplotype phasing, we found that this technique also improves genotype calling accuracy.

Accurate haplotype phasing will have broad impacts on genomic and genetic research areas. First, reconstructing long haplotype blocks in reference panel will improve the accuracy of genotype imputation. Second, long haplotype blocks will help haplotype-based genetic association studies. Third, accurate haplotype phasing will produce more insights into population genetics inferences.

This work is one of the first to prove the feasibility of incorporating haplotype information over multiple sites in ultra-long reads and long insert paired-end reads for phasing sequencing data with improved accuracy. This provides additional methodological support for the ultra-low coverage sequencing design (Pasaniuc *et al.*, 2012). In our simulation studies, we only used the read length of 36 and 100 bp and the fixed insert size of 250, 500 and 1000 bp. For the 1000 Genomes Project chromosome 20 data, the insert size varies and the portion of the proportion of R_2 and R_3 reads also varies across different regions. For the chromosome 20, the proportions of reads covering two sites and at least three sites are 28.0% and 24.8% for the CEU samples and 27.3% and 24.1% for the YRI samples, respectively. For the major histocompatibility complex region, the proportions of reads covering two sites and at least three sites are higher: 25.4% and 34.6%% for the CEU samples and 25.5% and 31.5% for the YRI samples, respectively. The performance

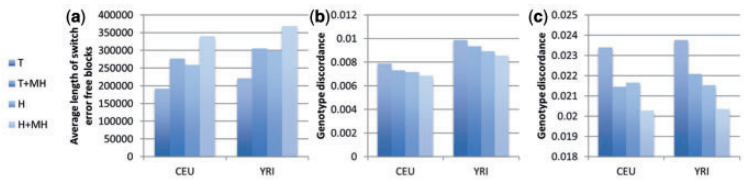

Fig. 5. (**a**) Switch errors for haplotype phasing and (**b**) genotype discordance rates for all genotypes and (**c**) heterozygote genotypes in real data. Results are over the 1000 Genomes Project phase 1 data for CEU and YRI individuals. Methods labels: T = Thunder, H = HapSeq and MH = interlaced MH-flipping

of the proposed method for such regions is expected to be further improved. Therefore, more studies are needed to show how the accuracy of haplotype phasing and genotype calling is affected by the length of reads, the length of inserts and the proportion of R_2 and R_3 reads using our newly developed method. It will be future work to conduct extensive simulations including the simulation of reads with varied insert sizes to investigate the optimal design strategies.

In the MH sampling, we proposed the new haplotype pair as a single crossover of the current haplotype pair and chose the recombinant point with the probability that is proportional to a weight. We defined the weight as the function of the difference of the number of sequencing reads that are in conflict with the current haplotype pair and the proposed haplotype pair. Although we also used the uniform weight and found the results from the uniform weight to be just slightly worse than the proposed weight, it is not clear whether the proposed weight is optimal. We will investigate this with more simulations in the future. To investigate whether the single crossover is sufficient for convergence of the MH sampling, we ran HapSeq2 $5c$, $10c$ and $20c$ iterations for the MH sampling, where c is the number of heterozygote sites of that individual. We found the results from $5c$ are similar to those obtained from $10c$ and $20c$. In addition, we found the acceptance ratios from $5c$, $10c$ and $20c$ are similar, indicating it is sufficient for convergence.

ACKNOWLEDGEMENTS

We thank Yingrui Li for helpful discussions. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Funding: NIH R00RR024163 (to D.Z.), NIH R01GM074913 (to K.Z.) and R01GM081488 (to K.Z.).

Conflict of Interest: none declared.

REFERENCES

Abecasis,G.R. *et al.* (2012) An integrated map of genetic variation from 1,092 human genomes. *Nature*, **491**, 56–65.

Bansal,V. *et al.* (2008) An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Res.*, **18**, 1336–1346.

Danecek,P. *et al.* (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.

Delaneau,O. *et al.* (2012) A linear complexity phasing method for thousands of genomes. *Nat. Methods*, **9**, 179–181.

Durbin,R.M. *et al.* (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

He,D. *et al.* (2012) Optimal algorithm for haplotype phasing with imputation using sequencing data. In: *The Fifteenth Annual Conference on Research in Computational Biology (RECOMB-2012)*. Barcelona, Spain.

Levy,S. *et al.* (2007) The diploid genome sequence of an individual human. *PLoS Biol.*, **5**, e254.

Li,H. (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.

Li,N. and Stephens,M. (2003) Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, **165**, 2213–2233.

Li,Y. *et al.* (2010) MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet. Epidemiol.*, **34**, 816–834.

Li,Y. *et al.* (2011) Low-coverage sequencing: implications for design of complex trait association studies. *Genome Res.*, **21**, 940–951.

Marchini,J. *et al.* (2007) A new multipoint method for genome-wide association studies by imputation of genotypes. *Nat. Genet.*, **39**, 906–913.

Pasaniuc,B. *et al.* (2012) Extremely low-coverage sequencing and imputation increases power for genome-wide association studies. *Nat. Genet.*, **44**, 631–635.

Schaffner,S.F. *et al.* (2005) Calibrating a coalescent simulation of human genome sequence variation. *Genome Res.*, **15**, 1576–1583.

Venter,J.C. *et al.* (2001) The sequence of the human genome. *Science*, **291**, 1304–1351.

Zhi,D. *et al.* (2012) Genotype calling from next-generation sequencing data using haplotype information of reads. *Bioinformatics*, **28**, 938–946.