Bioinformatics, 32(11), 2016, 1686–1696
doi: 10.1093/bioinformatics/btv530
Advance Access Publication Date: 9 September 2015
Original Paper

Genetic and population analysis

Reveel: large-scale population genotyping using low-coverage sequencing data

Lin Huang, Bo Wang, Ruitang Chen, Sivan Bercovici and Serafim Batzoglou*

Department of Computer Science, Stanford University, CA 94305, USA

*To whom correspondence should be addressed.

Associate Editor: Gunnar Ratsch

Received on April 8, 2015; revised on August 24, 2015; accepted on September 1, 2015

Abstract

Motivation: Population low-coverage whole-genome sequencing is rapidly emerging as a prominent approach for discovering genomic variation and genotyping a cohort. This approach combines substantially lower cost than full-coverage sequencing with whole-genome discovery of low-allele frequency variants, to an extent that is not possible with array genotyping or exome sequencing. However, a challenging computational problem arises of jointly discovering variants and genotyping the entire cohort. Variant discovery and genotyping are relatively straightforward tasks on a single individual that has been sequenced at high coverage, because the inference decomposes into the independent genotyping of each genomic position for which a sufficient number of confidently mapped reads are available. However, in low-coverage population sequencing, the joint inference requires leveraging the complex linkage disequilibrium (LD) patterns in the cohort to compensate for sparse and missing data in each individual. The potentially massive computation time for such inference, as well as the missing data that confound low-frequency allele discovery, need to be overcome for this approach to become practical.

Results: Here, we present Reveel, a novel method for single nucleotide variant calling and genotyping of large cohorts that have been sequenced at low coverage. Reveel introduces a novel technique for leveraging LD that deviates from previous Markov-based models, and which is aimed at computational efficiency as well as accuracy in capturing LD patterns present in rare haplotypes. We evaluate Reveel's performance through extensive simulations as well as real data from the 1000 Genomes Project, and show that it achieves higher accuracy in low-frequency allele discovery and substantially lower computation cost than previous state-of-the-art methods.

Availability and implementation: http://reveel.stanford.edu/.

Contact: serafim@cs.stanford.edu

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

1 Introduction

Identification of genomic variation in human DNA sequences is a key first step in associating alleles with human traits and diseases (The 1000 Genomes Project Consortium, 2012). Genome-wide association studies (GWAS) have successfully linked genetic variation across thousands of genotyped individuals and hundreds of traits (Feero and Guttmacher, 2010; Franke *et al.*, 2010; Hindorff *et al.*, 2009; The Wellcome Trust Case Control Consortium, 2007). Beyond human, association of genomic variations with traits has many applications, such as in the quality breeding of plants and livestock (Feuillet *et al.*, 2011; Huang and Han, 2014; The Bovine HapMap Consortium, 2009). Despite their success in linking variation with traits, GWAS performed on genotypes have so far failed to explain a large portion of the heritability of common traits and

diseases such as diabetes, schizophrenia and heart disease (Billings and Florez, 2010; Cirulli and Goldstein, 2010; Manolio *et al.*, 2009; Visscher *et al.*, 2012). Genotype-based GWAS have only examined common single-nucleotide polymorphisms (SNPs), and one promising avenue for finding the 'missing heritability' is the association of rare variants with common traits (Gibson, 2012; Lee *et al.*, 2014, Zuk *et al.*, 2014), also known as the '*common disease rare variant*' hypothesis. Many recent efforts have focused on discovering such rare variants in large cohorts through sequencing rather than genotyping (Tennessen *et al.*, 2012).

Algorithms that call SNPs on a single target genome require the sample to be sequenced at a high coverage (>30×) to confidently differentiate alternate alleles from sequencing errors (Bentley *et al.*, 2008; DePristo *et al.*, 2011; Li *et al.*, 2009; McKenna *et al.*, 2010). High-coverage sequencing, however, is expensive when applied to large cohorts. Recently, low-coverage sequencing of large cohorts has been proposed as more cost-efficient and informative than sequencing fewer individuals at high coverage (Li *et al.*, 2011). Many on-going projects have adopted this low-coverage strategy, including the UK10K project (http://www.uk10k.org), the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2010), the Cohorts for Heart and Aging Research in Genomic Epidemiology (CHARGE) Project (CHARGE Consortium, 2009) and multiple participating cohorts in the Haplotype Reference Consortium (http://www.haplotype-reference-consortium.org). Each project sequences thousands of individuals at a relatively low coverage. For example, the 1000 Genomes Project sequenced 2535 whole genomes at depth 4–6×; the CHARGE Project sequenced ∼5000 whole genomes at depth 7×.

To leverage the wealth of genomic data that such large-scale population sequencing projects are providing, computational methods that perform accurate and efficient detection and genotyping of rare SNPs in a population are urgently needed. The corresponding computational problem is considerably more challenging than single-sample genotyping from deep sequencing data: to overcome the noise and missing data inherent in low-coverage sequencing, variant detection and genotyping require the joint estimation of all genotypes of all individuals simultaneously and need to infer and leverage the linkage disequilibrium (LD) present in the sequenced cohort. As a result, computation time can become prohibitive and accuracy is harder to achieve for rare alleles.

A number of existing computational methods can be applied to population genotyping. Although not designed for analyzing low-coverage sequencing data, SAMtools (Li *et al.*, 2009), GATK Unified Genotyper (DePristo *et al.*, 2011; McKenna *et al.*, 2010) and Beagle (Browning and Browning, 2009) can perform population genotyping (The 1000 Genomes Project Consortium, 2012). In particular, applying GATK Unified Genotyper to 62 CEU samples from the 1000 Genomes Project pilot phase collectively, followed by Beagle, leads to reasonably accurate genotyping for common polymorphisms (Nielsen *et al.*, 2012). QCALL (Le and Durbin, 2011) employs a dynamic programming algorithm to estimate, for every position of the genome, the posterior probability of presence of an alternate allele in the cohort. The QCALL algorithm then constructs a set of possible ancestral recombination graphs from samples to estimate the SNP posterior probability for each site in each sample from these graphs. The glfMultiples + Thunder pipeline employs a hidden Markov model (HMM) that leverages LD information across a population to genotype likely polymorphic sites, and is currently considered the state of the art for accurate genotyping of populations using sequencing data (Li *et al.*, 2011). In the underlying HMM, each hidden state is a pair of reference haplotypes, which are most closely related to the sample being considered, and observations are genotype likelihoods. To apply this HMM on a sequenced cohort, the sequenced individuals are used as references. SNPTools (Wang *et al.*, 2013) estimates genotyping likelihoods at putative polymorphic sites using a BAM-specific binomial mixture model, in which the parameters are empirically estimated using an expectation–maximization (EM) algorithm. With the resulting genotyping likelihoods, SNPTools utilizes a HMM approach based on the statistical LD pattern model proposed in (Li and Stephens, 2003) to infer genotypes and haplotypes. This method restricts the number of parental haplotypes to reduce the computation overhead.

Despite their considerable success, existing genotyping methods are not ideally suited for application to large cohorts (5000–1 000 000 individuals) because of their prohibitive computation time, as well as their reduced accuracy when calling low-frequency genomic variants, which are hard to differentiate from sequencing errors. In particular, the HMM model underlying Thunder links polymorphic sites to surrounding mosaics, modeling these links using a first-order Markovian model. However, the presence of low frequency (0.5–5%) and rare (<0.5% frequency) variants hierarchically breaks the common haplotypes into many uncommon or rare haplotypes, reducing the fit to a model with an underlying Markovian assumption. Additionally, given a cohort of size n, the HMM requires $O(n^2)$ hidden states, which results in prohibitively high computational overhead as n increases. The SNP detection dynamic programming algorithm of QCALL, on the other hand, is more computationally efficient because it does not account for the non-random associations between loci, but its accuracy is reduced for the same reason.

Here, we present Reveel, a novel method for large-scale SNP discovery and genotype imputation using low-coverage sequencing data sets. Reveel leverages the underlying complex LD structure by employing a simplified model that scales linearly with the number of individuals in a cohort for a given number of imputed SNPs, while producing highly accurate genotype calls for both high- and low-frequency SNPs. We evaluate the performance of Reveel on simulated data, as well as real data and demonstrate that Reveel achieves significant improvements in both efficiency and accuracy over previous state-of-the-art population-scale genotyping methods, making Reveel a practical approach for large-scale population genotyping.

2 Methods

The input to Reveel is a cohort of n sequenced individuals, for which read counts are available supporting each of the four possible nucleotides $\boldsymbol{\ell} = (\ell_X)_{1\times 4}$ at every site in each sample. To genotype the individuals, Reveel performs the following four steps: (i) Polymorphic site discovery. A set of m putative polymorphic sites are identified across the genome. (ii) Initialization of genotypes $\mathbf{G} = (g_{i,j})$ for every sample i and putative polymorphic site j identified in step 1. (iii) Calculation of a rank three tensor $\mathbf{P} = (p_{i,j,g})$, representing the probability of individual i having genotype g in position j given the current assignment $\mathbf{G}$. (iv) Calculation of new assignment $\mathbf{G}'$ that maximizes the current entries of $\mathbf{P}$; steps 3 and 4 are performed iteratively until convergence. (v) Final refinement of the genotypes $\mathbf{G}$.

2.1 Polymorphic site discovery

Knowing the set of observed reads supporting each of the four possible nucleotides $\boldsymbol{\ell} = (\ell_X)_{1\times 4}$ at a site in a sample, we can compute the probability that allele $X \in \{A, C, G, T\}$ is present by

marginalizing over possible genotypes given the read counts ℓ: $P_X = \sum_Y \Pr\{g = \{X, Y\}|\ell\}$, where the genotype $g = \{X, Y\}$ is an unordered pair of alleles. The probability of genotype g given read counts can be computed as

$$\Pr\{g|\ell\} = \frac{\Pr\{\ell|g\}\Pr\{g\}}{\sum_{g^*}\Pr\{\ell|g^*\}\Pr\{g^*\}} \quad (1)$$

To compute the genotype probability, we first calculate the genotype likelihood, i.e. the probability of observing ℓ when the genotype is $g = \{X, Y\}$. The genotype can take one of 10 possible assignments. The likelihood of a homozygous genotype can be written as a binomial probability mass function $f_{\text{binomial}}(\ell_X; \sum\ell, 1-\varepsilon)$, in which ε is the sequencing base error rate. Reveel takes ε as an input parameter, but is robust to differences between the input ε and the real sequencing error rate (see Results). The likelihood of a heterozygous genotype can be expressed as follows, in which the indicator function $1_{\text{condition}}$ equals to 1 if the condition is true; otherwise it equals to 0.

$$\Pr\{\ell|g\} = \binom{\sum\ell}{\ell_A \quad \ell_T \quad \ell_C \quad \ell_G} \cdot \prod_{\ell_Z \in \{\ell_A, \ell_T, \ell_C, \ell_G\}} \left[\frac{1}{2}\cdot\left(\frac{\varepsilon}{3}\right)^{1_{Z\neq X}}\cdot(1-\varepsilon)^{1_{Z=X}} + \frac{1}{2}\cdot\left(\frac{\varepsilon}{3}\right)^{1_{Z\neq Y}}\cdot(1-\varepsilon)^{1_{Z=Y}}\right]^{\ell_Z} \quad (2)$$

The polymorphic prior probability of a heterozygous genotype is assigned according to mutation type, as in previous work (Li *et al.*, 2011): if X and Y are transition mutations ($A \leftrightarrow G$ or $C \leftrightarrow T$), then the prior is set to 2/3; if X and Y are transversion mutations (A *or* $G \leftrightarrow C$ *or* T), then the prior is set to 1/6. The prior probability of a homozygous genotype is set to 1/3, the mean value of polymorphic prior probabilities.

We distinguish loci that contain true variations from those that arose from sequencing errors, as follows. Given a target locus and a candidate allele X, we define score_X representing the strength of the evidence for the existence of allele X at the target locus, using the summation of a monotonically increasing function over all the samples:

$$\text{score}_X = \sum_{\text{samples}} h(P_X) \quad (3)$$

We define the allele with the highest score as the reference allele. The allele with the second highest score is a putative alternate allele. We distinguish sites that exhibit alternate alleles from those that exhibit only sequencing errors using a threshold score_{th} (by default, 0.5).

The function h was trained using simulated annealing on simulated data sets in the design stage (different from the data sets used in the experiments), maximizing the overall recall under the perfect precision constraint. The function $h(z) = a \cdot z/(1 + a - z)$ with a sole parameter $a = 5 \times 10^{-6}$ fit the training data well, and was built in the tool and applied in all our experiments. During training of this function, score_{th} was set to 1 because our initial experiments showed that for large data sets (thousands of samples) threshold 1 gives near-perfect precision. However, to increase sensitivity during application of Reveel, we set the default threshold to the lower value of 0.5 and allow users to change it in command line.

The motivation behind h and Equation (3) is to capture strong evidence for the existence of allele X even if such strong evidence appear only once in the cohort. In the meantime, multiple weak evidences that are likely to be induced by sequencing errors only boost score_X to a limited extent.

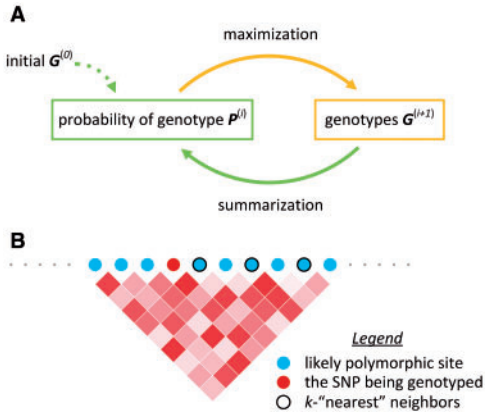

Fig. 1. An overview of Reveel. (**A**) Reveel infers the genotypes using a summarization–maximization iterative method. In each iteration we first apply the summarization step to every SNP and we then apply the maximization to every SNP. The summarization step calculates the genotype probabilities using the current estimation of genotypes and observed reads in the context of linkage disequilibrium. The maximization finds the genotypes that maximizing the genotype probabilities obtained in the summarization step. These genotypes are then used to refine the genotype probabilities in the next summarization step. We iterate these two steps until convergence. (**B**) The underlying network of Reveel is composed of a set of likely polymorphic sites and the linkage disequilibrium among them. For every polymorphic site, we pick its *k*-'nearest' neighbor sites in terms of linkage disequilibrium to facilitate genotype calling at the target site

2.2 Genotype-calling algorithm

Given m candidate polymorphic sites that were identified in the previous step, we determine the genotypes of n samples simultaneously across the m sites. Let $\mathbf{G}$ be a $n \times m$ matrix, in which $g_{i,j} = \{0, 1, 2\}$ represents the genotype of sample i at marker j being homozygous reference, heterozygous, homozygous alternate, respectively. Let P be a $n \times m \times 3$ rank three tensor, where $p_{i,j,h}$ represents the probability of $g_{i,j} = h$. We formulate the overall framework of our algorithm as a fixed-point model

$$P = f(P|\text{reads}) \quad (4)$$

The function $f(z)$ does not have a closed-form expression; instead, we estimate P by using an iterative algorithm that alternates between two steps: summarization and maximization. Given the genotype matrix G, in the summarization step, we estimate P in the context of LD and observed reads. In the maximization step, we update G with the genotypes associated with the largest probabilities within P. We iteratively apply these two steps until convergence (Fig. 1A).

$$\mathbf{P}^{(i)} = \Pr\{\mathbf{G}^{(i)}|LD,\ \text{reads}\} \quad (5)$$

$$\mathbf{G}^{(i+1)} \leftarrow \text{argmax}\mathbf{P}^{(i)} \quad (6)$$

In each iteration, we first apply summarization on all markers and then apply maximization on all markers. Using the subscript target to represent the marker being evaluated in a sample and $\overline{\text{target}}$ to represent all other makers in the same sample, we rewrite the above equations as:

$$p^{(i)}_{\text{target},h} = \Pr\{g^{(i)}_{\text{target}} = h|g^{(i)}_{\overline{\text{target}}},\ \text{reads}\} \quad (7)$$

$$g^{(i+1)}_{\text{target}} \leftarrow \arg\max_h p^{(i)}_{\text{target},h} \quad (8)$$

The main challenge lies in the summarization step, where the LD information needs to be leveraged in a computationally efficient

way that leads to high accuracy in estimating the conditional probabilities. Here, we introduce a technique that leverages the most informative markers in terms of LD. For each marker, we find its *k*-nearest neighbor markers in terms of LD, as defined in the next section (Fig. 1B). Equation (7) is replaced by:

$$p^{(i)}_{\text{target},h} = \Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}, \text{reads}\} \tag{9}$$

The observed reads provide two forms of evidence: the read counts supporting alleles at the target marker (denoted as r_{target}) and the allele frequencies at the evaluated marker across samples (denoted as θ). To utilize the read counts, we rewrite the conditional probability in Equation (9) using the chain rule to yield Equation (10).

$$p^{(i)}_{\text{target},h} \propto \Pr\{r_{\text{target}} | g^{(i)}_{\text{target}} = h\} \cdot \Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}\} \tag{10}$$

The calculation of the first term is straightforward. To calculate the second term, we use the probability of genotypes in the *i*-th iteration as follows. For each sample *j*, we calculate the probability that this sample has genotype *h* at the target locus and genotypes $g^{(i)}_{k\text{NN}}$ at the neighbor loci. We use subscript (target, *j*) to represent the marker on the same locus as the target but in sample *j*, distinguished from the target SNP being evaluated. Similarly, we use subscript (kNN,*j*) to represent the *k*-nearest neighbors in sample *j*. With these notations, the above probability can be expressed $\Pr\{g^{(i)}_{\text{target},j} = h, g^{(i)}_{k\text{NN},j} = g^{(i)}_{k\text{NN}}\}$. Summing this probability over all the samples yields the expected sample count C_h with genotype *h* at the target SNP and $g^{(i)}_{k\text{NN}}$ at the neighbors; summing over all the samples and all the possible *h*'s yields the expected count *C* having $g^{(i)}_{k\text{NN}}$ at the neighbors. We use the ratio C_h/C as the new conditional probability $\Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}\}$.

In practice, because the sample size is usually limited to hundreds or thousands, the conditional probability assessment could be biased (Friedman *et al.*, 1997), which can significantly affect the performance. To reduce bias, we use Laplace smoothing (Hansen *et al.*, 2005). In summary, the second term is given by the following expression, in which we set $t = 1$ if $AF \geq 1\%$ and $t = 0.01$ otherwise.

$$\Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}\} \approx \frac{\sum_j \Pr\{g^{(i)}_{\text{target},j} = h, g^{(i)}_{k\text{NN},j} = g^{(i)}_{k\text{NN}}\} + t}{\sum_{h^*} \sum_j \Pr\{g^{(i)}_{\text{target},j} = h^*, g^{(i)}_{k\text{NN},j} = g^{(i)}_{k\text{NN}}\} + 3t} \tag{11}$$

Although Laplace smoothing is used, if the initial $G^{(0)}$ is biased towards the homozygous reference on certain markers, then $\Pr\{g^{(i)}_{\text{target}} = 1 \text{ or } 2 | g^{(i)}_{k\text{NN}}\}$ tends to be a very small number. Thus, the results after convergence are also likely to be biased. To address this issue, we leverage the other signal given by the reads, that is, the alternate allele frequency over samples, and rewrite Equation (10) as:

$$p^{(i)}_{\text{target},h} \propto \Pr\{r_{\text{target}} | g^{(i)}_{\text{target}} = h\} \cdot \Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}, \theta\} \tag{12}$$

Once again, we face the problem of assessing the conditional probability in the second term, but this time we obtain knowledge from an additional source. Let p^{kNN}_h be $\Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{kNN}\}$, and let p^{θ}_h be $\Pr\{g^{(i)}_{\text{target}} = h | \theta\}$. The probabilities evaluated from different sources are combined using a noisy-MAX gate (Zagorecki and Druzdzel, 2013). The expressions are as follows.

$$\Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k\text{NN}}, \theta\} = \sum_{\forall u,v: \max\{u,v\}=h} p^{kNN}_u \cdot p^{\theta}_v \tag{13}$$

In contrast to the estimate provided by Equation (11), which is biased towards homozygous reference, this estimate is biased towards homozygous alternate. We use each of the above two estimates alternatingly in the iterations of our summarization-maximization algorithm.

2.3 Nearest neighbor calculation

To define the *k* nearest neighbors of a locus, we introduce three metrics to approximate the LD between two loci. As this evaluation is performed on every pair of candidate polymorphic sites, we need metrics with low computational overhead. Commonly used metrics such as the correlation coefficient require the estimation of genotypes based on the observed reads; this estimation involves a considerable computational cost. The main benefit of the metrics we present here is that they can be directly applied on the read counts.

Let S_i be a set of samples that have at least one read at locus *i* supporting alternate alleles. The first metric is defined as the Jaccard index of two sets

$$sim_1(i,j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{14}$$

This metric utilizes the presence of reads that support alternate alleles. As a second, more informative metric, we apply the Jaccard index on multisets, accounting for repeated elements. Set S'_i is defined as the collection of $r_{i,t}$ copies of *t*'s, where $r_{i,t}$ is the number of reads at locus *i* of sample *t* supporting alternate alleles. The second metric is thus

$$sim_2(i,j) = \frac{|S'_i \cap S'_j|}{|S'_i \cup S'_j|} = \frac{\sum_t \min\{r_{i,t}, r_{j,t}\}}{\sum_t \max\{r_{i,t}, r_{j,t}\}} \tag{15}$$

Finally, we define the third metric that produces a more rapidly increasing score as both samples exhibit more reads that support alternate alleles:

$$sim_3(i,j) = \frac{\sum_t \min\{r_{i,t}, r_{j,t}\}^2}{\sum_t \max\{r_{i,t}, r_{j,t}\}^2} \tag{16}$$

We apply the summarization-maximization algorithm separately, leading to tensors $\mathbf{P}_i$ for $i = 1$, 2 and 3. Then, we combine the three tensors by using the average probability at each marker (also called the mean combination rule (Kittler *et al.*, 1998; Xu *et al.*, 1992)):

$$\mathbf{P} = \mathbf{E}[\mathbf{P}_i] \tag{17}$$

The combined genotype matrix is given by

$$\mathbf{G} \leftarrow \arg\max \mathbf{P} \tag{18}$$

We conduct three rounds of genotyping; in each round we pick or re-pick *k*-nearest neighbors and then apply 10 iterations of summarization-maximization inference, using the output genotypes of the previous round as initial genotypes. In the first round, the *k* nearest neighbors of each locus are selected using the above similarity metrics. In the second round, we estimate the LD between pairs of loci and re-pick *k* nearest neighbors accordingly. Because the linkage phase is unknown, we use a composite LD estimator Δ as proposed previously (Schaid, 2004):

$$\Delta = 2p_{aabb} + p_{aaBb} + p_{Aabb} + \frac{1}{2}p_{AaBb} - 2p_a p_b \tag{19}$$

where *A* and *B* represent the major alleles of two loci, *a* and *b* represent the minor alleles. In the third round, we re-pick *k* nearest

neighbors using a time-efficient approximation of Δ. Equation (19) is approximated as a function of $sim_1(i,j)$:

$$\Delta \approx \frac{1}{2} sim_1(i,j) + p_a p_b \cdot \left(p_a + p_b - \frac{1}{2} p_a p_b - 2 \right) \quad (20)$$

The estimation of Δ using Equation (19) costs $O(m^2 n)$ computation time, which is more computationally expensive than using Equation (20), which costs $O(m^2)$. Therefore we provide users with a Reveel-lite option, which utilizes Equation (20) in both round two and round three. This way the running time can be reduced by almost half, with minimal reduction in accuracy (see Results).

Instead of read counts that support each of the four possible nucleotides at every site in each sample, our tool can also take genotype likelihoods (GLs) as input. When genotype likelihoods are provided, we restore read counts from GLs to calculate sim_1, sim_2, sim_3; we directly use GLs in the genotype inference, that is, we substitute $\Pr\{r_{\text{target}} | g^{(i)}_{\text{target}} = h\}$ in Equation (10) with GLs.

The inter-marker LD at most extends to a few hundred kilobases (kb) (Reich *et al.*, 2001; Schaffner *et al.*, 2005). To compute nearest neighbors efficiently, we tile the genome with a set of non-overlapping blocks. The k-nearest neighbor markers are selected from the block to which the target marker belongs. We found that block sizes of 500 kb to 1 Mb result in high accuracy and practical running time. Our default block size is 1 Mb.

Parameter k has a great influence on the quality of the approximation in Equation (11). Let $Q_h(k) \cdot \sum_j \Pr\{g^{(i)}_{\text{target},j} = h, g^{(i)}_{k\text{NN},j} = g^{(i)}_{k\text{NN}}\}$ and $Q(k) \cdot \sum_h Q_h(k)$. An overly large k can result in very small $Q(k)$ and therefore low-quality conditional probability tables. Assuming LD, $Q(k)$ can be roughly estimated as $n \cdot [(1-maf)^2]^A \cdot [2 \cdot maf \cdot (1-maf)]^B \cdot [maf^2]^C$, where A, B, C are the counts of 0, 1, 2 in the genotype pattern $g^{(i)}_{k\text{NN}}$ and $A + B + C = k$. In other words, given a fixed sample size n, $Q(k)$ exponentially shrinks with the increment of k. Based on our experiments, we recommend the following settings: $n \leq 75$, $k = 2$; $75 < n \leq 250$, $k = 3$; $n > 250$, $k = 4$. As cohorts become considerably larger than 1KGP in the future, we expect that larger values for k will yield better performance.

Finally, the conditional probability computed with different values for k conveys LD on different levels. To balance the impact of the selection of k, we rewrite Equation (10) as

$$p^{(i)}_{\text{target},h} \propto \Pr\{r_{\text{target}} | g^{(i)}_{\text{target}} = h\} \cdot \sum_{k^*=1}^{k} [w_{k^*} \cdot \Pr\{g^{(i)}_{\text{target}} = h | g^{(i)}_{k^*\text{NN}}\}] \quad (21)$$

where the weight w_{k^*} can be $1/k^*$ or $2k^*/(k+k^2)$. In our experiments, we use Equation (21) with $w_{k^*} = 1/k^*$.

2.4 Initial genotypes

Given a low-coverage sequencing data set, we observe only a few (if any) reads at a target site. Thus, using these reads to estimate $g^{(0)}_{\text{target}}$ is not a good initial guess. Instead, we use the reads at the k nearest loci to amplify the low-coverage data. More formally, let r_i and $\hat{r}_i$ be the number of reads at locus i supporting alternate and reference alleles. Instead of using r_{target} and $\hat{r}_{\text{target}}$, we use $R_{\text{target}} = r_{\text{target}} + \sum r_{kNN}$ and $\hat{R}_{\text{target}} = \hat{r}_{\text{target}} + \sum \hat{r}_{kNN}$ for the initial guess, which is equivalent to amplifying the depth of the target site. We assign

$$g^{(0)}_{\text{target}} \leftarrow \arg\max_g \Pr\{g | R_{\text{target}}, \hat{R}_{\text{target}}\} \quad (22)$$

2.5 Final refinement

The method described in the previous sections achieves sufficiently high performance with a very limited number of neighbor SNPs. To further improve the genotyping accuracy at common SNPs, the Reveel package provides an optional final refinement step, which applies a phasing method to the common and low frequency SNPs (allele frequencies $\geq$1%). Since previous publications have proposed high-quality phasing algorithms (Browning and Browning, 2009; DePristo *et al.*, 2011; McKenna *et al.*, 2010), we use BEAGLE (Browning and Browning, 2009) for this step. We feed the genotype likelihoods of high-frequency SNPs into BEAGLE, and then merge the phased dosage into our outputs.

3 Results

3.1 Performance on simulated data based on 1KGP

3.1.1 Experimental setup

We created a simulated data set, *1kgp-sim*, which mimicked the features of the 1000 Genomes Project (1KGP) dataset (The 1000 Genomes Project Consortium, 2010), including high variability of sequencing depth among loci and individuals. *1kgp-sim* included 2535 samples; each corresponded to a sample in the 1KGP data set. To create these samples, we simulated variants in 10 000 haplotypes for a 1-Mbp region using COSI with parameters from the best-fitting model (Schaffner, 2005). A 1-Mbp region on chromosome 20 (43 000 000–44 000 000) of the human genome build GRCh37 was used as a reference genome. Combining these variants with the reference genome resulted in 10 000 chromosomes. A simulated sample was a composition of two randomly selected simulated chromosomes. Then, for each sample, we downloaded the BAM files from the 1KGP database to obtain the mapping position and length of each real read, and we generated a simulated read with the same position and length, and with sequencing base errors injected into a haplotype of the simulated sample; the sequencing base error rate was set to 0.1% (Lou *et al.*, 2013; Robasky *et al.*, 2014). We further examined the performance of Reveel under a wide range of sequencing base error rates: 0.0001–1% (see 'Performance as a function of sequencing error rate' in Section 3.1.4). The base qualities of the simulated reads were copied from the downloaded bam files. Finally, the simulated reads were mapped to the reference genome by BWA (Li and Durbin, 2009). The resulting bam files (as opposite to the bam files downloaded from the 1KGP) were used in this set of experiments. The mapped depth was 7.4×. We generated three additional data sets, *1kgp-sim-n100*, *1kgp-sim-n500* and *1kgp-sim-n1000*, from *1kgp-sim* by using randomly selected 100, 500 and 1000 individuals, respectively.

3.1.2 Comparison against other methods

We compared the SNP-calling performance of Reveel with that of three state-of-the-art methods: SNPTools' pileup→varisite commands (v1.0), GATK Unified Genotyper (v3.3) and glfMultiples. While GATK HaplotypeCaller was a more recent tool, in practice Unified Genotyper was recommended for analyzing data sets with more than 100 samples for performance reasons (Van der Auwera *et al.*, 2013). Our experiments confirm that Unified Genotyper is dramatically faster with similar accuracy to HaplotypeCaller (Supplementary Table S1). We therefore compared Reveel to GATK Unified Genotyper.

We benchmarked the genotyping performance of Reveel against three state-of-the-art pipelines: (i) SNPTools + Beagle, in which SNPTools' bamodel→poprob commands (v1.0) estimated genotype

Table 1. Genotyping accuracy and running time

Method	*1kgp-sim-n100* (c-site # 4291)				*1kgp-sim-n500* (c-site # 7016)				*1kgp-sim-n1000* (c-site # 8590)			
	site #	acc (%)	c-acc (%)	Time (min)	site #	acc (%)	c-acc (%)	Time (min)	site #	acc (%)	c-acc (%)	Time (min)
Reveel	4393	99.7719	99.7749	2.0	7607	99.9246	99.9249	16	9927	99.9512	99.9697	52
Reveel + Beagle	4393	99.8288	99.8399	2.9	7607	99.9454	99.9567	24	9927	99.9670	99.9809	81
Reveel-lite	4393	99.7398	99.7488	1.6	7607	99.9107	99.9105	9	9927	99.9388	99.9568	26
SNPTools + Beagle	4571	99.7213	99.7252	8.2	7597	99.9150	99.9205	217	9399	99.9437	99.9595	1089
GATK + Beagle	4434	99.6786	99.6802	13.4	7524	99.8911	99.8912	388	9745	99.9258	99.9435	1806
glfMultiples + Thunder	4549	99.6747	99.6909	307.0	7700	99.9224	99.9216	2736	8886	99.9397	99.9375	6120

We evaluated the genotype-calling performance of Reveel and three state-of-the-art methods: glfMultiples followed by Thunder, GATK Unified Genotyper applied to all the samples collectively followed by Beagle, and SNPTools followed by Beagle. For each method, we measured the genotyping accuracy at the polymorphic sites discovered by the corresponding SNP discovery tool (acc). The site # columns show the number of the sites that the various SNP discovery tools called. We also measured the genotyping accuracy at the polymorphic sites discovered by all four methods (c-acc). These sites are referred to as consensus sites (c-site).

likelihoods at polymorphic sites and then Beagle 4 (r1399) inferred genotypes. SNPTools + Beagle was used as a baseline of our performance evaluation. (ii) GATK + Beagle, in which Beagle 4 (r1399) used the genotype likelihoods generated by GATK Unified Genotyper (v3.3) to infer genotypes. (iii) glfMultiples + Thunder, in which the computationally demanding yet accurate genotyping method Thunder is applied to the output of glfMultiples. Unless otherwise specified, we used the default parameters for previous methods.

3.1.3 SNP discovery

First, we measured the ability of Reveel, SNPTools, GATK and glfMultiples, to identify sites in the genome that are polymorphic in the samples. Reveel exhibited near-perfect performance in discovering common SNPs (Supplementary Fig. S1). Then, we compared the performance of Reveel in detecting rare and low frequency SNPs to the performance of various methods (Supplementary Fig. S2). SNPs were divided into three bins according to their allele frequencies (AF): <0.1%, 0.1–0.2% and 0.2–0.5%. For each bin, we report the recall of each method, as the fraction of SNPs that were identified among all SNPs in the bin. We report precision as the fraction of identified loci that showed more than one allele in the simulated 10 000 haplotypes out of all reported loci.

As shown in Supplementary Figure S2, Reveel outperformed the other methods in discovering SNPs with AF <0.1% for all data sets and SNPs with AF ranging between 0.1 and 0.2% for the $n = 100$, 500 and 1000 cases. On SNPs with AF 0.2–0.5%, Reveel showed similar recall with GATK for the $n = 500$, 1000 and 2535 cases, and higher recall for the $n = 100$ case; SNPTools showed slightly higher recall with Reveel for the $n = 2535$ case. In a large cohort, a SNP with AF 0.2–0.5% is very likely to be captured by multiple reads in a few samples; hence, state-of-the-art callers such as SNPTools and GATK are capable of discovering SNPs of moderate AF.

3.1.4 Genotyping

Genotyping accuracy. We measured the genotyping accuracy of each method, defined as the percentage of inferred genotypes that are correct. Table 1 demonstrates Reveel's genotyping accuracy measured with default parameters, compared with SNPTools + Beagle, GATK + Beagle and glfMultiples + Thunder. Three genotyping modes of Reveel were examined: the default Reveel system, Reveel followed by Beagle, and Reveel-lite. Each method was applied on the variants discovered by its own pipeline. The performance was measured at the consensus sites, which were detected by all four SNP discovery tools. As the CPU overhead of Thunder was considerably high when Thunder was applied to *1kgp-sim* (expected 14.7 days), we did not report the comparison for this data set. We also compared the genotyping performance of Reveel and the lightweight pipeline SNPTools + Beagle on all four datasets (Supplementary Table S2). In this set of experiments, we tuned Reveel's threshold $score_{th}$ to achieve a 100% precision level in SNP discovery, and then infer genotypes at the variants discovered by Reveel. For comparison, we used SNPTools (bamodel→poprob) to estimate genotype likelihoods at the same call set, and then applied Beagle 4 for genotype inference.

Genotype calling using Reveel, and Reveel + Beagle, achieved superior accuracy over the other methods in all the experiments. Reveel-lite also outperformed the other methods for the $n = 100$ case; for the $n = 500$ case its genotyping accuracy was higher than GATK + Beagle but lower than SNPTools + Beagle and glfMultiples + Thunder; for the $n = 1000$ case its genotyping accuracy was comparable to SNPTools + Beagle and higher than the other two methods. The ratio between the measured genotyping error rate at consensus sites of Reveel + Beagle and those of three state-of-the-art methods ranged between 0.31 and 0.58. As the sample size increased from 100 to 1000, the genotyping error rate of Reveel + Beagle measured at the consensus sites (the Reveel-called sites) dramatically reduced by a factor of 8.4 (5.2). When the sample size increased to thousands, Reveel approached perfect performance. In our experiments, Reveel w/o Beagle (Reveel + Beagle) achieved 99.9703% (99.9803%) genotyping accuracy on the *1kgp-sim* data set with 2535 samples. These results indicate that Reveel will become increasingly powerful as cohort sizes increase in the future.

The summarization-maximization iterative algorithm of Reveel converges rapidly: with $n = 1000$, 99.67% of the loci converged after 10 iterations when alternating between Equations (11) and (13) in the iterative algorithm, and 99.75% of the loci converged when only Equation (11) was applied.

In all these experiments, we ran Beagle 3.3.2 for 20 iterations as the final refinement of Reveel + Beagle. In preliminary studies, we also tried running Beagle 4 for 5 burn-in iterations and 15 iterations of genotype phase estimation. We found the pipeline with Beagle 4 provided slightly higher genotyping accuracy with moderately longer running time (data not shown).

Computation time. We compared the running time of Reveel to the other three population genotyping methods on a 2.67GHz Intel

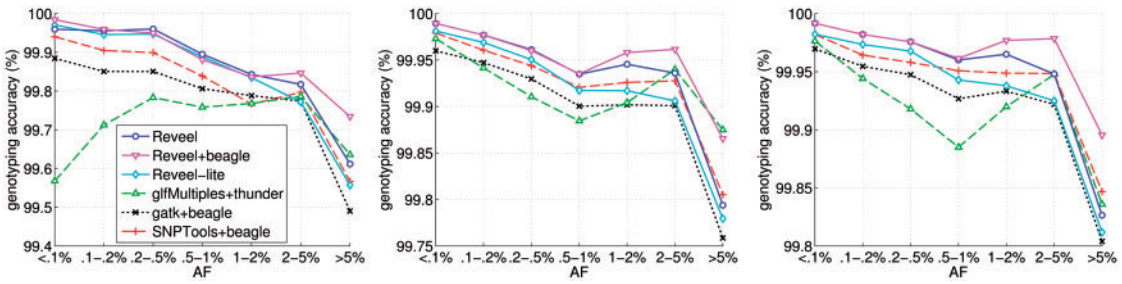

Fig. 2. Genotyping performance as a function of allele frequencies. The figures from left to right show the performance for the n = 100, 500, 1000 cases, respectively. The polymorphic sites were categorized according to their population minor allele frequencies, which were computed as the percentage of minor alleles in 10 000 simulated haplotypes. We compared the performance of Reveel and the other methods at the sites in each category

Xeon X5550 processor, as shown in Table 1 and Supplementary Table S2. The numbers shown in these tables are the computational overhead of SNP discovery and genotyping, unless otherwise specified. Reveel-lite, Reveel, Reveel + Beagle were more than 192, 118, 76 times faster than glfMultiples + Thunder respectively, and significantly faster than SNPTools + Beagle and GATK + Beagle especially when the sample size was greater than 100.

More importantly, Reveel scales well to larger datasets. Out of three genotyping modes, Reveel-lite has the best scalability. The process of finding k-nearest neighbors for m polymorphic sites across n individuals has a time complexity of $O(nm^2)$, which we further reduce by restricting the calculation to blocks of size<1Mbp; the time complexity of the iterative algorithm is $O(nm)$.

Although Reveel + Beagle and SNPTools + Beagle utilized Beagle, Reveel + Beagle was considerably faster than SNPTools + Beagle and that the ratio between Reveel + Beagle's and SNPTools + Beagle's running times increased from 3× to 13× as the sample size increased from 100 to 1000. The reason is as follows. In the Reveel + Beagle pipeline, Beagle was only applied to common and low-frequency SNPs (AF ≥ 1%), which were a fraction of discovered polymorphic sites. For the $n = 100$, 500, 1000, 2535 cases, the fractions were 67.9, 40.8, 31.3, 22.1, respectively. Moreover, increasing the sample size from 100 to 500 increased the number of common SNPs by a small factor (3.8%). Further increasing the sample size to 1000 did not increase the number of common SNPs. As a result, the computational overhead of Beagle in the Reveel + Beagle pipeline scaled well with cohort size. In contrast, SNPTools + Beagle applied Beagle on all the detected polymorphic sites, regardless of their allele frequencies. The site count significantly increased with the sample size.

We note that the other methods report haplotype phasing information because genotyping calls are computed by finding the best haplotype pairs for each individual. Reveel finds genotypes directly, and does not report phasing information, which is outside the scope of this work.

Performance on uncommon SNPs. We grouped SNPs according to their AFs and compared the performance of tools on each group (Fig. 2). Reveel + Beagle exhibits higher accuracy than the other methods in almost every group. The only exception is for the $n = 500$ case Thunder shows slightly higher genotyping accuracy on SNPs with AF > 5%. We further investigated the $n = 500$ case by categorizing SNPs into three types: homozygous reference, heterozygous and homozygous alternate (Supplementary Fig. S3). Reveel + Beagle achieved very low genotyping error rate at homozygous reference SNPs across the AF spectrum; at heterozygous SNPs Reveel + Beagle, SNPTools + Beagle and GATK + Beagle outperformed glfMultiples + Thunder; at homozygous alternate SNPs Reveel + Beagle demonstrated low genotyping error rate on most SNPs except for SNPs with AF > 5%. Because the reported genotyping accuracy was dominated by the large number of homozygous reference sites, we also measured the performance of tools on calling alternate alleles on each group (Supplementary Table S3). Again, Reveel exhibited better performance than the other three methods for most cases. We also grouped SNPs in *1kgp-sim-n1000* according to whether they are homozygous reference (hom-ref), heterozygous (het), or homozygous alternate (hom-alt) and reported accuracy in each class (Supplementary Table S4). It has been previously suggested that high genotyping accuracy at sites with low AF can be achieved by simply assigning homozygous reference (also known as a 'straw-man' approach) (Li *et al.*, 2011). Supplementary Table S4 shows that Reveel rarely calls alternate alleles as reference.

Performance as a function of sequencing error rate. Reveal is robust to sequencing reads' base error rate (Supplementary Table S5). In addition to *1kgp-sim*, we created five simulated data sets in which the injected sequencing base error rates were 1, 0.5, 0.2, 0.05, 0.0001%, as the reported sequencing base call accuracy of leading NGS technologies ranges from 99.9 to 99.9999% (Robasky *et al.*, 2014). As shown in the table, the performance of 0.5, 0.2, 0.05 and 0.0001% cases remains on the same level as the 0.1% case. In the 1% case, the genotyping error rate is 0.035%, which is still a very low number.

Reveel does not require users to input an accurate sequencing error rate. In the previous examples we set the input sequencing base error rate ε of Reveel as 0.1% regardless of the true value. Unless the input $\hat{\varepsilon}$ for ε underestimated ε by a factor of 10, the parameter setting in our experiments did not reduce the performance by a non-negligible extent. When $\log_{10}\hat{\varepsilon}/\varepsilon \leq -1$, an accurate estimate of base error rate slightly boosted the genotyping performance. For instance, we also tried to set this value to be 1% for the 1% case, the genotyping accuracy increased to 99.9698%.

Performance as a function of sequencing coverage. We uniformly downsampled the simulated data sets to sequencing coverage 2–7.4× to examine the robustness of Reveel with respect to coverage (Supplementary Table S6). As the sequencing coverage decreased from 7.4× to 4× we observed a slight decline in genotyping accuracy. The rate of decline increased when we pushed the coverage to 2×, but even in this extreme case Reveel achieved 99.7109% genotyping accuracy for the $n = 1000$ case. This demonstrates the applicability of Reveel to large data sets with very shallow sequencing depth.

We also varied the parameter k to investigate the optimal k for different levels of sequencing coverage. No clear connection between k and sequencing coverage was observed (Supplementary Table S6).

3.2 Performance on 1KGP samples

We applied Reveel to the low-coverage sequencing data from the 1000 Genomes Project Phase 3. This data set includes 2535 samples from 26 populations (Supplementary Table S7). We restricted our analysis to a 5-Mbp region on chromosome 20 (43 000 000–48 000 000), which we call *1kgp-real*. Reveel was applied to call SNPs and genotypes from each population separately. The block size was set to 500 kb; the threshold $score_{th}$ was set to its default value 0.5. As a post-processing step, we merged SNPs detected in each population, and we reported genotypes of each sample measured at the merged set $\mathcal{S}_U$. The genotype of a sample from population p at a SNP that belongs to $\mathcal{S}_U$ but not $\mathcal{S}_p$ was treated as a homozygous major within population p.

For comparison, we applied SNPTools + Beagle, GATK + Beagle and glfMultiples + Thunder to the same data set *1kgp-real*. Similar to the application of Reveel, we merged the SNP sets called from all 26 populations and evaluated the genotyping accuracy at the union SNP set using the HapMap 3 benchmarks. Whenever no tool reported a locus as a SNP, for GATK + Beagle and glfMultiples + Thunder, we assumed all the samples were homozygous reference at this locus, where the reference allele came from the reference genome. As SNPTools did not report sites at which all the samples in the studied cohort have homozygous alternates as SNPs, we added these sites into the variant list before the list was fed into SNPTools (bamodel→poprob) for estimating genotype likelihoods.

3.2.1 SNP discovery

Reveel discovered 163 024 likely polymorphic sites in 26 populations from the 1KGP Phase 3. The African populations contributed 36 703 putative SNPs, while the other populations showed lower diversity (Table 2). The 1KGP Phase 1 called variants from 1092 samples and reported 68 208 SNPs in the analyzed region; our method identified 94.63% of those SNPs. The transition to transversion ratio (Ts/Tv) for the variants overlapping with the 1KGP Phase 1 was 2.58. The putative SNPs were primarily rare variants (Supplementary Fig. S4): more than 95% of the putative SNPs were with allele frequencies $\leq$1%; only 1.5% of putative SNPs had allele frequencies >5%. Their Ts/Tv ratio was 2.10.

We conducted a set of experiments to benchmark each method's false positive rates of variant discovery (Supplementary Table S8 and S9). In the first experiment, across all the samples sequenced by both 1KGP Phase 3 and complete genomes (CG) data set in the 1000 Genomes Project, we defined as gold standard positives as all the positions where CG data find non-reference alleles, and defined as test outcome positives as all the positions where the tested method finds non-reference alleles, where each locus occurrence in each individual is counted separately. We evaluated the false positive rate and the sensitivity of each method without applying any filters that aim to distinguish true polymorphisms from errors (Supplementary Table S8). Reveel achieved similar sensitivity level as other methods and much lower false positive rate. We also compared Reveel with the integrated call set reported by the 1KGP Phase 3 (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/supporting/input_callsets). The integrated call set was generated by integrating variant calls made by many tools, and multiple filters (such as SVM in the GotCloud package, Atlas2 in the SNPTools + Atlas2 pipeline) were applied during the generation; as expected, the integrated call set outperformed each individual method. The same experiment was repeated using the trios in 1KGP Pilot 2 as the benchmark.

In the second experiment, when we calculated the false positive rate and sensitivity, each locus was counted once, regardless of how many individuals it was detected to be in. More precisely, we defined gold standard positives as across the studied cohort the positions where CG data find at least one non-reference alleles; test outcome positives as across the studied cohort the positions where genotyping method finds at least one non-reference alleles. The comparison of various methods shown in Supplementary Table S9A is consistent with the first experiment. Again, this experiment was repeated on two 1KGP Pilot 2 trios that were sequenced at high coverage (Coverage $\sim$ 40×); for these trios, we treat the genotyping calls as gold standard and measure the methods' ability to genotype the 1KGP individual from the low-coverage sequencing data alone (Supplementary Table S9B).

Interestingly, Reveel discovered 1676 triallelic sites and 13 sites having all four nucleotides across all 1KGP individuals. We manually examined these 13 sites in Supplementary Table S10. Among them, three sites (chr20:45227442, 45341056, 47122443) were reported as quadallelic in the integrated call set of the 1KGP Phase 3; three sites (chr20:45227444, 46166386, 46316306) were reported as quadallelic by a SNP discovery pipeline adopted by the 1KGP Phase 3; one site (chr20:43132939) was reported as a triallelic site by two adopted SNP discovery pipelines, which aggregately discovered all four nucleotides. Out of the remaining six sites, one site (chr20:45440123) was reported as a triallelic site in the integrated call set; one site (chr20:47131459) was reported as a biallelic site in the integrated call set. The other four sites (chr20:44229327, 47131663, 47132131, 47552368) were not reported in the integrated call set. Loci chr20:44229327 and 47131663 appeared to have complex variants, as reported using Freebayes and recalibrated by GATK. 1KGP Phase 3 did not have consistent variant calls at these loci. Loci chr20: 47132131 and 47552368 were detected as SNPs using Freebayes but no other pipelines. Supplementary Table S10 describes all 13 sites.

We compared the SNPs discovered by Reveel and glfMultiples from 99 CEU samples and 109 YRI samples of the *1kgp-real* data set. The SNPs detected by only one method were compared to the SNPs reported in the CEU and YRI trios from the 1000 Genomes Project Pilot 2, because these trios are sequenced at high depth (42× on average) and their genotype calls are likely to be of high accuracy (The 1000 Genomes Project Consortium, 2010, Xu *et al.*, 2012) and consequently any putative SNPs detected by either method that are also called in these trios have strong evidence of being true. As shown in Supplementary Table S11, both methods discovered

Table 2. The SNPs discovered in populations

Population ancestry	Number of populations	Discovered SNPs	SNPs reported in Phase 1	Putative SNPs
East Asian	5	45 931	24 056	21 875
South Asian	5	52 431	20 593	31 838
African	7	80 236	43 533	36 703
European	5	46 006	27 164	18 842
Americas	4	48 875	33 734	15 141

We applied Reveel to each population from the 1000 Genomes Project separately, and then collected the SNPs discovered from the populations with the same ancestry. Our tool revealed a large number of putative SNPs from the South Asian populations and the African populations.

22 518 SNPs in the CEU trio and 36 141 SNPs in the YRI trio. In addition, each method identified a number of SNPs not found by the other method. Of those, glfMultiples identified more than three times as many as Reveel. The vast majority (~99%) of SNPs identified by only one method were not identified by the deep trio sequencing, and that proportion was slightly higher for glfMultiples, which was consistent with Reveel having a lower false positive rate than glfMultiples.

The Venn diagram Supplementary Figure S5 shows the overlaps of the SNP call sets obtained using Reveel and the other three methods.

3.2.2 Genotyping accuracy

HapMap 3 benchmark. We evaluated performance on the genotype calls using the genotypes reported in the HapMap Phase III panel (Altshuler *et al.*, 2010) as benchmarks. Out of 26 populations in the *1kgp-real* data set, HapMap 3 studied nine populations: ASW, CEU, CHB, GIH, JPT, LWK, MXL, TSI and YRI. The number of the common samples between HapMap 3 and 1KGP in these populations were 50, 90, 94, 93, 97, 90, 56, 96 and 103, respectively. Reveel + Beagle achieved high accuracy on most populations (Fig. 3). Performance on the ASW and MXL was lower, perhaps due to the fact that there are only 66 and 67 samples for these populations, respectively.

Figure 3 shows that Thunder and Reveel + Beagle perform similarly, whereas in simulations Reveel + Beagle outperforms Thunder. One possible explanation for the discrepancy is that the SNPs reported by HapMap 3 are primarily common SNPs (Supplementary Table S12), in which the two methods have similar performance in simulations. SNPTools + Beagle and GATK + Beagle had comparable performance with Thunder in simulations, but they did not perform equally well as Thunder on real data.

Complete Genomics benchmark. We also measured performance using the genotypes called from the Complete Genomics data set in the 1000 Genomes Project. Of the 427 samples in the CG data set, 287 samples were also in the *1kgp-real* data set, including 63 CEU, 62 CHS, 3 KHV, 10 LWK, 62 PEL, 32 PJL, 3 PUR and 52 YRI samples. Since the CG variants were called from high-coverage sequencing data, this benchmark contains variants with any allele frequencies.

We evaluated accuracy at heterozygous sites and homozygous non-reference sites for each method. Specifically, for every sample we focused on the sites where the CG data reported heterozygous or homozygous non-reference and calculated the percentage of correctly called sites. Figure 4 shows the boxplots of performances for each population. Reveel + Beagle demonstrated higher accuracy than the other methods in the vast majority of the cases, except that glfMultiples + Thunder had higher accuracy than Reveel + Beagle in the YRI case.

Supplementary Figure S6 demonstrates that using read counts and using genotype likelihoods as inputs result in almost identical genotyping accuracy on the 1KGP data set.

Reveel exhibited substantially lower running times than other methods on the 1KGP data (Supplementary Table S13).

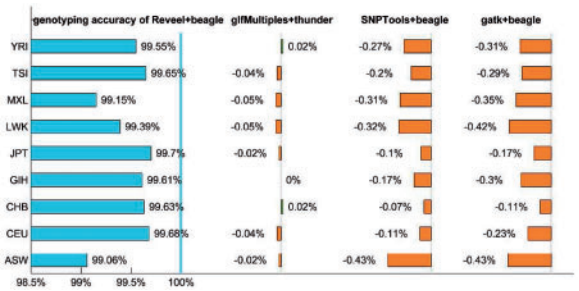

Fig. 3. Genotyping accuracy evaluated using HapMap3 benchmark. Genotyping accuracy was evaluated using the genotypes of 50 ASW, 90 CEU, 94 CHB, 93 GIH, 97 JPT, 90 LWK, 56 MXL, 96 TSI, and 103 YRI samples reported by HapMap 3 as the benchmark. The blue bars represent the genotyping accuracy of Reveel + Beagle. For the other three methods, the bars show the difference from Reveel + Beagle: orange indicates lower accuracy than Reveel + Beagle; green indicates higher accuracy than Reveel + Beagle

3.2.3 Discordance of alleles between HapMap3 and 1KGP

We also observed a few sites where alleles between HapMap3 and 1KGP are discordant (Supplementary Table S14). Four sites where alleles called by HapMap3 and 1KGP (type 1) were discovered by Reveel and verified by GATK + Beagle, and matched the report of a previous publication (Qin *et al.*, 2013). Three alleles where the allele frequencies are considerably different between the genotypes reported by HapMap3 and the genotypes reported by 1KGP (type 2), were also found by both Reveel and GATK + Beagle, and these were not reported (Qin *et al.*, 2013). For example, at locus chr20: 44697887 HapMap3 reported the vast majority of haplotypes having G (99.53%) and only a small portion having T (0.47%), while Reveel inferred 2.19% G and 97.81% T from 1KGP. At locus chr20:47590564 HapMap3 reported 99.82% C and 0.18% T, while 1KGP exhibited 9.25% C and 90.75% T. At chr20:48661748 although both data sets supported the major allele being A and the minor allele being G, the minor allele frequency reported by HapMap3 was 44.99% and that obtained from 1KGP was only 9.01%. Finally, three loci that were reported as SNPs in HapMap3 and not in 1KGP (type 3), were also reported as constant by both GATK + Beagle and Reveel. When we evaluated the genotyping accuracy of tools, we excluded all the above loci.

4 Discussion

A rare genetic variant that originated from a recent mutation event tags many of the other genetic variants surrounding it, as these were present at that time, including variants at long genetic distance from it; rare variants present an extremely high LD, yielding long rare haplotypes. The nearest-neighbor concept in Reveel uniquely leverages this observation: common SNPs tend to have nearest neighbors that are proximal in genetic distance, while rare SNPs tend to have nearest neighbors that are much more distant (Supplementary Fig. S7, Supplementary Fig. S8A); moreover, the allele frequencies of target SNPs and their nearest neighbors are in almost perfect linear correlation (Supplementary Fig. S8B).

HMM-based methods face a tradeoff between either explicitly modeling each rare haplotype, which leads to computational overhead due to the large number of parameters, or compressing the state space which leads to the loss of long-distance rare-haplotype LD information. In particular, previous state-of-the-art methods, such as MaCH and Thunder, apply a first-order Markovian model between two subsequent haplotypic positions. While such models have been demonstrated to work well for genotyping common variants, they face a challenge in modeling rare variants. On the lower side of the rare SNPs spectrum ($\leq 0.1\%$), the incorporation of LD information in previous methods did not improve the genotyping accuracy in the 1000 Genomes Project Phase 1; rather, the resulting

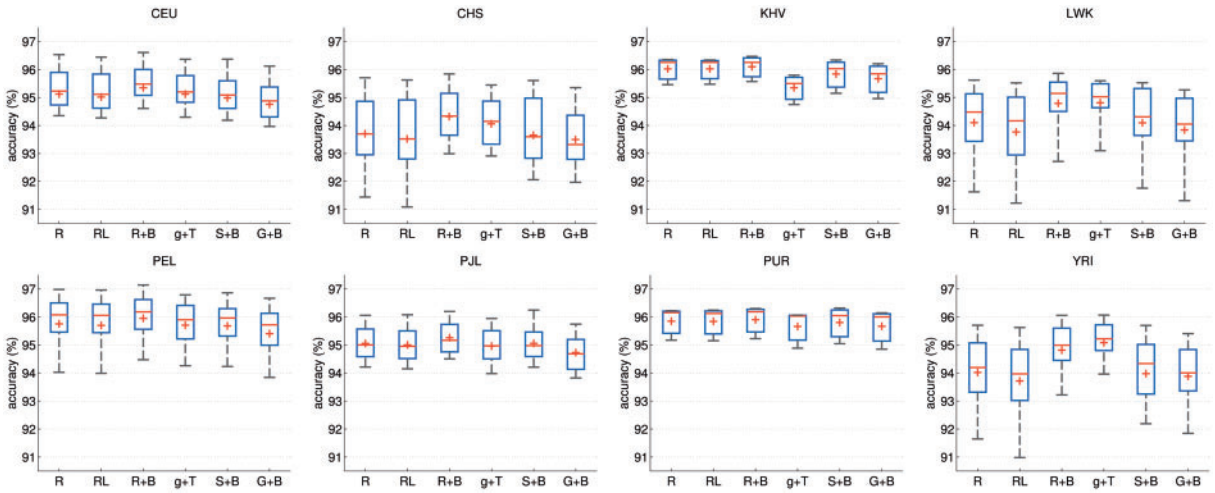

Fig. 4. Genotyping accuracy evaluated at heterozygous sites and homozygous non-reference sites of Complete Genomics benchmark. For every sample, we evaluated three Reveel genotyping modes (R: Reveel, RL: Reveel-lite, R + B: Reveel + Beagle) and three other methods (g + T: glfMultiples + Thunder, S + B: SNPTools + Beagle, G + B: GATK + Beagle) at the sites where the sample is heterozygous or homozygous non-reference and reported the percentages of correctly inferred sites. The evaluations were performed on the same set of sites. The samples from a population were aggregated into a subfigure. In the boxplots, the central marks are the median, the red lines are the mean, the edges of the box are the 25th and 75th percentiles, and the whiskers span 9th to 91st percentiles

genotyping accuracy was modestly lower than when not using LD information (The 1000 Genomes Project Consortium, 2012). The explanation underlying this phenomenon may be as follows. Although rare haplotypes share common variants, they usually contain distinct rare variants that can serve as a signature. Leveraging those correlations within a simple Markovian model is impractical: every rare haplotype needs to be encoded in the model, captured as a distinct sequence of states in the HMM. The HMMs underlying currently available methods tend to eliminate rare alleles as noise, which contributes to biases towards homozygous reference. Conversely, to infer genotypes, our approach aims to identify the most informative sites in a way that is less sensitive to their genetic distance. The strategy is different from previous models that implicitly weaken the association between remote sites. By focusing on the most informative markers based on their LD, our method provides considerable improvement in the genotype calling of rare variants.

High AF SNPs are caused by one or more mutation events that occurred in the distant past; after many generations of recombination, the LD between high AF sites could become very complex. Therefore, perfect LD between a high AF site and a set of surrounding sites may not exist. In this particular case, genotype phasing on common variants is a useful complementary method to our genotype-calling algorithm. We incorporate a post-processing step into Reveel in the Reveel + Beagle pipeline: after imputing the genotypes and genotype probabilities at likely polymorphic sites, we pick SNPs with AF > 1% and feed their genotype probabilities into Beagle (Browning and Browning, 2009) for phasing. Finally the output dosages of Beagle are merged with the genotypes at rare SNPs.

An important feature of our algorithm at high AF sites is providing high-quality genotype probabilities. To demonstrate this point, we conducted an experiment for comparison, labeled as *Reveel-gatk-beagle*. In this experiment, we forced GATK to make calls across the sites identified by our algorithm with AF > 1%. Then, Beagle was trained on the outputs of GATK, producing dosages at these sites. Finally, we merged the outputs of Beagle and the genotypes called by our algorithm at rare SNPs for evaluation. The only difference between this approach and our Reveel + Beagle pipeline is which tool was used to created genotype probabilities. The comparison shown in Supplementary Figure S9 clearly illustrates that Reveel + Beagle outperforming *Reveel-gatk-beagle*.

The running time of Reveel scales linearly with the number of individuals n and linearly with the number of polymorphic sites m in our algorithms, expect that the process of estimating LD between every pair of polymorphic sites requires m^2 computations. As we restrict the LD estimation within windows of size 500 kb–1 Mb (see Section 2), m is usually within the range of a few thousands depending on the size of the studied cohort, which results in practical running times in our experiments.

Reveel has been demonstrated to be robust to high variability of sequencing depth across loci and individuals. The 1000 Genomes Project Phase 3 data set used in our experiments has non-uniform sequencing coverage: the mapped coverage of 2535 samples in the studied region ranges from 2.13 to 35.3, with mean value 6.99 and standard deviation 2.56. The average coverage of these loci across the cohort ranges from 0.0197 to 20.7, with standard deviation 1.19.

In summary, Reveel is a highly accurate and efficient tool for single nucleotide variation calling and genotyping of large cohorts of low-coverage sequenced individuals. In future work, similar techniques may be applied to leverage a population's LD to provide rapid and accurate genotyping of other types of variation, such as insertions, deletions and structural variants.

Acknowledgements

We thank our labmates, especially Yuling Liu, for discussions on this project, and Robyn Brinks Lockwood for improving the writing of the manuscript.

Funding

This work was supported in part by a grant from the Stanford-KAUST alliance for academic excellence. L.H. was supported in part by a Stanford Graduate Fellowship.

Conflict of Interest: S. Batzoglou is a co-founder of DNAnexus, and a member of the scientific board of 23andMe and Eve Biomedical. S. Bercovici is Chief Technology Officer of Lifecode, Inc.

References

Altshuler,D.M. *et al*. (2010) Integrating common and rare genetic variation in diverse human populations. *Nature*, **467**, 52–58.

Bentley,D.R. *et al*. (2008) Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, **456**, 53–59.

Billings,L.K. and Florez,J.C. (2010) The genetics of type 2 diabetes: what have we learned from GWAS? *Ann. N. Y. Acad. Sci.*, **1212**, 59–77.

Browning,B.L. and Browning,S.R. (2009) A unified approach to genotype imputation and haplotype phase inference for large data sets of trios and unrelated individuals. *Am. J. Hum. Genet.*, **84**, 210–223.

CHARGE Consortium. (2009) Cohorts for Heart and Aging Research in Genomic Epidemiology (CHARGE) Consortium: Design of prospective meta-analyses of genome-wide association studies from five cohorts. *Circ. Cardiovasc. Genet.*, **2**, 73–80.

Cirulli,E.T. and Goldstein,D.B. (2010) Uncovering the roles of rare variants in common disease through whole-genome sequencing. *Nat. Rev. Genet.*, **11**, 415–425.

DePristo,M.A. *et al*. (2011) A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nat. Genet.*, **43**, 491–498.

Feero,W.G. and Guttmacher,A.E. (2010) Genome wide association studies and assessment of the risk of disease. *N. Engl. J. Med.*, **363**, 166–176.

Feuillet,C. *et al*. (2011) Crop genome sequencing: lessons and rationales. *Trends Plant Sci.*, **16**, 77–88.

Franke,A. *et al*. (2010) Genome-wide meta-analysis increases to 71 the number of confirmed crohn's disease susceptibility loci. *Nat. Genet.*, **42**, 1118–1125.

Friedman,N. *et al*. (1997) Bayesian network classifiers. *Mach. Learn*, **29**, 131–163.

Gibson,G. (2012) Rare and common variants: twenty arguments. *Nat. Rev. Genet.*, **13**, 135–145.

Hansen,G.A. *et al*. (2005) Mesh Enhancement: Selected Elliptic Methods, Foundations and Applications. Imperial College Press, London, UK.

Hindorff,L.A. *et al*. (2009) Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *PNAS*, **106**, 9362–9367.

Huang,X. and Han,B. (2014) Natural variations and genome-wide association studies in crop plants. *Annu. Rev. Plant Biol.*, **65**, 531–551.

Kittler,J. *et al*. (1998) On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**, 226–239.

Le,S.Q. and Durbin,R. (2011) SNP detection and genotyping from low-coverage sequencing data on multiple diploid samples. *Genome Res.*, **21**, 952–960.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, **25**, 1754–1760.

Lee,S. *et al*. (2014) Rare-variant association analysis: study designs and statistical tests. *Am. J. Hum. Genet.*, **95**, 5–23.

Li,H. *et al*. (2009) The sequence alignment/map format and samtools. *Bioinformatics*, **25**, 2078–2079.

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

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

Lou,D.I. *et al*. (2013) High-throughput DNA sequencing errors are reduced by orders of magnitude using circle sequencing. *PNAS*, **110**, 19872–19877.

Manolio,T.A. *et al*. (2009) Finding the missing heritability of complex diseases. *Nature*, **461**, 747–753.

McKenna,A. *et al*. (2010) The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome Res.*, **20**, 1297–1303.

Nielsen,R. *et al*. (2012) Genotype and snp calling from next-generation sequencing data. *Nat. Rev. Genet.*, **12**, 443–451.

Qin,P. *et al*. (2013) A panel of ancestry informative markers to estimate and correct potential effects of population stratification in Han Chinese. *Eur. J. Hum. Genet.*, **22**, 248C–253.

Reich,D.E. *et al*. (2001) Linkage disequilibrium in the human genome. *Nature*, **411**, 199–204.

Robasky,K. *et al*. (2014) The role of replicates for error mitigation in next-generation sequencing. *Nat. Rev. Genet.*, **15**, 56–62.

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

Schaid,D.J. (2004) Linkage disequilibrium testing when linkage phase is unknown. *Genetics*, **166**, 505–512.

Tennessen,J.A. *et al*. (2012) Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *Science*, **337**, 64–69.

The 1000 Genomes Project Consortium. (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

The 1000 Genomes Project Consortium. (2012) An integrated map of genetic variation from 1 092 human genomes. *Nature*, **491**, 56–65.

The Bovine HapMap Consortium. (2009) Genome-wide survey of snp variation uncovers the genetic structure of cattle breeds. *Science*, **324**, 528–532.

The Wellcome Trust Case Control Consortium. (2007) Genome-wide association study of 14 000 cases of seven common diseases and 3 000 shared controls. *Nature*, **447**, 661–678.

Van der Auwera,G.A. *et al*. (2013) From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Curr. Protoc. Bioinformat*, **43**, 11.10.1–11.10.33.

Visscher,P.M. *et al*. (2012) Five years of GWAS discovery. *Am. J. Hum. Genet.*, **90**, 7–24.

Xu,F. *et al*. (2012) A fast and accurate SNP detection algorithm for next-generation sequencing data. *Nat. Commun.*

Xu,L. *et al*. (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst Man Cybern. Syst.*, **22**, 418–435.

Wang,Y. *et al*. (2013) An integrative variant analysis pipeline for accurate genotype/haplotype inference in population NGS data. *Genome Res.*, **23**, 833–842.

Zagorecki,A. and Druzdzel,M.J. (2013) Knowledge engineering for bayesian networks: how common are noisy-max distributions in practice? *IEEE Trans. Syst. Man Cybernet. Syst.*, **43**, 186–195.

Zuk,O. *et al*. (2014) Searching for missing heritability: designing rare variant association studies. *PNAS*, **111**, E455–E464.