BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 1 2013, pages 29–38
doi:10.1093/bioinformatics/bts645

Sequence analysis

Advance Access publication October 27, 2012

Rare variant discovery and calling by sequencing pooled samples with overlaps

Wenhui Wang, Xiaolin Yin, Yoon Soo Pyon, Matthew Hayes and Jing Li*

Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106, USA

Associate Editor: Michael Brundo

ABSTRACT

Motivation: For many complex traits/diseases, it is believed that rare variants account for some of the missing heritability that cannot be explained by common variants. Sequencing a large number of samples through DNA pooling is a cost-effective strategy to discover rare variants and to investigate their associations with phenotypes. Overlapping pool designs provide further benefit because such approaches can potentially identify variant carriers, which is important for downstream applications of association analysis of rare variants. However, existing algorithms for analysing sequence data from overlapping pools are limited.

Results: We propose a complete data analysis framework for overlapping pool designs, with novelties in all three major steps: variant pool and variant locus identification, variant allele frequency estimation and variant sample decoding. The framework can be used in combination with any design matrix. We have investigated its performance based on two different overlapping designs and have compared it with three state-of-the-art methods, by simulating targeted sequencing and by pooling real sequence data. Results on both datasets show that our algorithm has made significant improvements over existing ones. In conclusion, successful discovery of rare variants and identification of variant carriers using overlapping pool strategies critically depend on many steps, from generation of design matrixes to decoding algorithms. The proposed framework in combination with the design matrixes generated based on the Chinese remainder theorem achieves best overall results.

Availability: Source code of the program, termed VIP for Variant Identification by Pooling, is available at http://cbc.case.edu/VIP.

Contact: jingli@cwru.edu

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

Received on April 4, 2012; revised on August 27, 2012; accepted on October 24, 2012

1 INTRODUCTION

In the past few years, genome-wide association studies have successfully identified many genetic variants associated with complex human diseases and traits (McCarthy *et al.*, 2008; Wellcome Trust Case Control, 2007). However, many of those variants identified by genome-wide association studies can explain only a small proportion of inherited risk (Manolio *et al.*, 2009). One possible explanation is that genetic variants with low minor allele frequencies (MAFs) and rare variants contribute a substantial fraction of the missing heritability (Altshuler *et al.*, 2010). Although common genetic variations such as single nucleotide polymorphisms (SNPs) have been well studied and characterized through international collaborative effort [e.g. the HapMap project (Altshuler *et al.*, 2005; Frazer *et al.*, 2007)], investigation of low MAFs or rare SNPs is much harder. First of all, large sample sizes are required to discover and type rare SNPs. Furthermore, before a rare SNP has been discovered, array-based technologies cannot be used to type individual genotypes from samples. Only very recently, with the development of next generation sequencing technologies, the international community has been able to systematically survey variants with low MAFs using large number of population samples through collaborations (Thousand Genomes Project Consortium, 2010). Results from the 1000 genomes project (Thousand Genomes Project Consortium, 2010) about variant locations, types and allele frequencies lay a solid foundation in studying relationships of genotypes and phenotypes. However, the cost for a project at this scale (e.g. sequencing >2000 samples with deep coverage) is still prohibitively high for most individual investigators, even though sequencing costs have been continuously decreasing. On the other hand, for case–control-based studies on complex diseases and traits, thousands of samples are normally required for each study to achieve a reasonable power.

An effective strategy to reduce the overall cost is to pool DNA sequences from different individuals together and then to sequence the pooled DNA with high coverage. It not only capitalizes on the continually decreasing sequencing costs per se but also can effectively reduce the costs associated with DNA preparations and target capturing for targeted sequencing projects. The cost for target capturing is proportional to the number of samples (i.e. number of individuals without pooling versus number of pools with pooling) and remains stable over the years. Pooling strategies can save tremendously on sample preparations. Furthermore, for targeted sequencing projects, depending on the size of targeted regions and coverage, researchers may not be able to fully take advantage of instruments' capacity if sequencing one individual at a time, even with multiplexing using barcodes. Inspired by this strategy, both wet lab experiments using pooling (Calvo *et al.*, 2010; Momozawa *et al.*, 2011; Nejentsev *et al.*, 2009; Out *et al.*, 2009) and methodologies serving pooling (Bansal *et al.*, 2010; Druley *et al.*, 2009; Kim *et al.*, 2010; Lee *et al.*, 2011; Rivas *et al.*, 2011; Vallania *et al.*, 2010; Wang *et al.*, 2010; Wei *et al.*, 2011) have emerged recently. Among them, Syzygy (Calvo *et al.*, 2010; Rivas *et al.*, 2011;

*To whom correspondence should be addressed.

Rivas and Daly, 2011) has been used in several real sequencing projects that used the pooling strategy. However, the main limitation of the naive pooling strategy is its inability to detect variant carriers, which is of high importance for disease-association studies of rare variants. Multiplexing using barcodes can partially solve the problem (Nijman *et al*., 2010; Smith *et al*., 2010), but the cost is still high because of limited barcoding capacity each run and per sample cost for DNA preparation and target capturing.

A promising alternative, so-called overlapped pooling designs, has been proposed by several groups (Erlich *et al*., 2009; Prabhu and Pe'er, 2009). The basic idea is rooted in combinatorial designs. By allocating individuals into a small number, but different pools, it is possible to identify samples that carry variants based on their pool signatures. Overlapped pooling designs represent a very important and economically feasible approach to discover rare SNPs and to identify rare mutant carriers. By doing so, one can avoid the two-step approach where rare variant loci are first discovered by sequencing and then genotyped using customized arrays. However, the studies of such an important strategy have been limited so far. There are mainly two existing algorithms: DNA Sudoku (Erlich *et al*., 2009) and the logarithm design (Prabhu and Pe'er, 2009) (the latter is termed Overlap Log in this study). DNA Sudoku allocates samples to pools (i.e. to construct a design matrix) based on the Chinese remainder theorem (CRT) (Ding *et al*., 1996) and identifies variant carriers with the help of combinatorial group testing theory (Du and Hwang *et al*., 2000). However, DNA Sudoku has several limitations. First, it assumes that all variant loci are known as a prior, therefore, cannot be used in detecting novel rare variants. Second, its SNP calling algorithm, which is based on the ratio of counts of variant alleles and reference alleles, does not take sequencing errors into consideration, which can significantly affect calling results for rare variants. Finally, it does not provide variant allele frequency (VAF) estimation, which is commonly used in testing associations for case–control studies. The Overlap Log (Prabhu and Pe'er, 2009) constructs its design matrix based on the binary representation of integers (sample identification numbers). Although such a design is 'optimal' in terms of the number of pools required, it cannot effectively identify variant carriers, mainly because it allocates too many individuals in each pool.

To address these issues, we propose a complete data analysis framework for overlapping pool designs, which consists of several steps. The framework, termed VIP, for Variant Identification by Pooling, is very flexible and can be combined with any pool design approaches and sequence mapping/alignment tools. Our major contributions include algorithms for variant pool and variant locus identification, VAF estimation and decoding of variant samples (Fig. 1). To identify variants in pooled samples, we propose a log likelihood ratio statistic to estimate the variant allele ratio (VAR) in each pool. Theoretical analysis further reveals the relationship between the statistical power of the test to detect variants of certain frequencies and sequencing coverage (i.e. read depth) and error rate. Variant loci can then be declared based on variant pool identification. The VAF of each variant locus can consequentially be estimated based on a weighted average of the estimated VAR of all pools. Finally, a set of algorithms are proposed to identify variant carriers. To evaluate the effectiveness of VIP, we have performed extensive experiments by simulating a

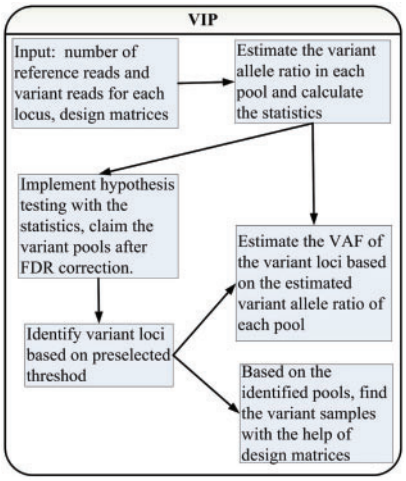

Fig. 1. The flow chart of VIP

targeted-sequencing project with different number of samples (500 and 1000), different design strategies and different per-sample sequence coverage. In comparison with the two existing approaches (i.e. DNA Sudoku and Overlap Log), VIP has made significant improvements in all steps. For example, for variant pool and variant locus identification, F-scores (the harmonic mean of precision and recall) have been increased from (0.03–0.33) to (0.94–0.99) for various datasets. In particular, VIP Sudoku (VIP in combination with Sudoku design matrixes) achieves best overall results. To further evaluate the performance of VIP under more realistic settings, we have created another dataset by pooling the real sequence data from the 1000 Genome project (Thousand Genomes Project Consortium, 2010) using the Sudoku design and compared its performance with another newer algorithm Syzygy (Calvo *et al*., 2010; Rivas *et al*., 2011; Rivas and Daly, 2011). Syzygy was recently developed for the non-overlapping pool design and has been used in real sequencing projects (Calvo *et al*., 2010; Rivas *et al*., 2011). It can detect variant pools/loci and estimate allele frequencies among other functionalities. However, Syzygy was not designed for overlapping pool strategies, and it cannot directly decode variant carriers. To apply Syzygy on this new dataset, we have combined Syzygy with some of the strategies in VIP for allele frequency estimation from overlapped pools and variant carrier decoding. The modified program, termed VIP_Syzygy, was then compared with VIP on this dataset. Results show that VIP has better performance than VIP_Syzygy in all tasks performed, including variant pool/variant locus detection, allele frequency estimation and variant carrier identification.

2 METHODS

2.1 Pool designs

The first step to use overlapping pool strategies is to construct a design matrix, specifying which pool contains which sample. In this subsection, we will first introduce some notations and then discuss two existing pool design strategies based on DNA Sudoku and Overlap Log, which will also be used in this study. A pool design can be represented by a design matrix $M = (m_{i,j}), i = 1, \ldots, t;\ j = 1, \ldots, n$, where t is the number of

pools, n is the number of samples, $m_{i,j}=1$ indicates that sample j is assigned to pool i. The j-th column of matrix M (denoted as M_j) represents the pools in which sample j participates. The i-th row $M_{(i)}$ represents the samples in pool i. Let w_j denote the weight of column M_j, i.e. the number of 1 s in M_j, which indicates the total number of pools in which sample j participates. Let $w_{(i)}$ denote the weight of row $M_{(i)}$, i.e. the number of individuals in pool i. A design is called column balanced if all w_j's are the same and is called row balanced if all $w_{(i)}$'s are the same. A design is balanced if it is both column balanced and row balanced. Let $\lambda(i,j)=<M_i,M_j>$, where $<M_i,M_j>$ is the dot-product of the two vectors. The minimal of all column weights, w_{min}, and the maximum of $\lambda(i,j)$, denoted as λ_{max}, are two important parameters for a design in determining its capacity to identify carriers for rare variants. Theoretically, it has been shown that variant carriers can be identified if the total number of such carriers in the samples is no more than the decoding robustness, which is defined as $d=\left\lfloor\frac{w_{min}-1}{\lambda_{max}}\right\rfloor$ (Erlich *et al.*, 2009; Kautz and Singleton, 1964).

The logarithm design (Prabhu and Pe'er, 2009) is based on binary representations of integers (sample IDs). More specifically, for each integer from 1 to n, one can use a bitword with a size of $\lceil log_2^n \rceil$ for its binary representation. For sample j, the bitwords column vector representation of $j-1$ and that of $n-j$ together form the j-th column of M. The total number of pools is thus equal to $2\lceil log_2^n \rceil$. The logarithm design is a balanced design with column weight $\lceil log_2^n \rceil$ and row weight $n/2$.

The Sudoku design (Erlich *et al.*, 2009) is based on the CRT (Ding *et al.*, 1996). For n individuals, it first finds integers $x_1, \ldots, x_w$, such that $\sqrt{n} \le x_1 < x_2 < \ldots < x_w$, $\sum_{i=1}^{w} x_i < n$ and $x_1, \ldots, x_w$ are pairwise co-prime. It then creates w groups of pools. The group corresponding to x_i has x_i pools. Therefore, it totally yields $\sum_{i=1}^{w} x_i < n$ pools. For the group of pools for x_i, individual j only participates in one pool, which is determined by j mod x_i. Or equivalently, for $0<r_i \le x_i$, the 1s in row/pool r_i are determined by $j \equiv r_i (mod x_i)$, which assign individual $j = r_i, r_i + x_i, r_i + 2x_i, \ldots$ to pool r_i in this group. The CRT guarantees $\lambda_{max}=1$ for the Sudoku design. The column weight is w. Therefore, the decoding robustness is equal to $w-1$. Figure 2 shows an example of a Sudoku design with 15 samples and 9 pools. The weight is two, and the decoding robustness is one. It is obvious that the logarithm design yields smaller number of pools, but the Sudoku design normally has larger decoding robustness.

		Sample															Sum
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Pool	a	X	0	0	0	X	0	0	0	X	0	0	0	X	0	0	2
	b	0	X	0	0	0	X	0	0	0	X	0	0	0	X	0	
	c	0	0	X	0	0	0	X	0	0	0	X	0	0	0	X	
	d	0	0	0	X	0	0	0	X	0	0	0	X	0	0	0	
	A	X	0	0	0	0	X	0	0	0	0	X	0	0	0	0	1
	B	0	X	0	0	0	0	X	0	0	0	0	X	0	0	0	
	C	0	0	X	0	0	0	0	X	0	0	0	0	X	0	0	
	D	0	0	0	X	0	0	0	0	X	0	0	0	0	X	0	1
	E	0	0	0	0	X	0	0	0	0	X	0	0	0	0	X	
Sum		2	0	0	1	1	1	0	0	2	0	1	0	1	1	0	

Fig. 2. An example to illustrate the Sudoku design and our decoding strategies. The design matrix is for 15 samples with nine pools. It has two groups of pools ($w=2$) with $x_1=4$ (for pools *a*–*d*) and $x_2=5$ (for pools *A*–*E*). Samples 1, 5, 9 and 13 are assigned to pool *a* (the first pool in the first group) because their ID values mod four is one. Other samples are assigned to other pools based on the same principle. Therefore, for any two individuals, they can potentially share at most one pool ($\lambda_{max}=1$). The decoding robustness d is thus one for this matrix. To determine samples with variants, assuming pools a, A and D being variant pools. Each number in the last row contains the number of *variant* pools in which an individual has participated. All columns with the last row equal to the column weight (two here) are potential variant carriers (columns 1 and 9 here). The numbers in the last column, corresponding to variant pools, are the numbers of potential variant carriers. The entries with 1 in the last column indicate that the potential variant carrier in each corresponding pool must be a real carrier. In the aforementioned examples, both potential variant samples (individual 1 and 9) are real variant carriers

Once a design matrix is given, DNAs from all samples will be pooled according to the design matrix and will be sequenced. Sequence reads can be aligned using existing mapping tools [e.g. MAQ (Li *et al.*, 2008)], and the observed results for each pool at a specific locus are the number of reads with the reference allele, the number of reads with variant alleles, as well as read mapping quality scores and base quality scores, which will serve as the input data for subsequent analysis.

2.2 Variant pool and variant locus detection

2.2.1 Statistical framework At each sequence position/locus, we consider whether a pool contains any samples with variant alleles. For this study, we assume there are at most two possible alleles/nucleotides at each locus. Let R_{1i} denote the number reads with the variant allele and R_{0i} denote the number of reads with the reference allele for pool i. Let μ_i denote the VAR in pool i (i.e. the fraction of variant alleles carried by all samples in the pool). Some variants are real, i.e. from samples with variants, whereas others might be sequencing errors. We first assume a constant sequencing error rate as a prior, denoted as ER. To determine whether a pool contains some variant samples is equivalent to test whether $\mu_i=0$. Therefore, we propose the following hypothesis test:

$$H_0 : \mu_i = 0 vs. H_1 : \mu_i > 0.$$

Assume that each allele from an individual in a pool is independently sampled with an equal probability, then all the reads are independently, identically distributed, with the probability of a read having a variant allele under H_0 being $\nu_i=ER$ and the probability of having a reference allele being one—ν_i (i.e. under H_0, each of the R_{1i} reads follows a Bernoulli distribution with the parameter ER). Under H_1, the probability of having a variant allele is $\nu_i = \mu_i(1-ER)+(1-\mu_i)ER$. Therefore, the maximum likelihood ratio test (LRT) can be written as:

$$y_i = \frac{\max_{\mu_i \in H_0} \nu_i^{R_{1i}}(1-\nu_i)^{R_{0i}}}{\max_{\mu_i \in H_0 \cup H_1} \nu_i^{R_{1i}}(1-\nu_i)^{R_{0i}}} \quad (1)$$

Under H_0, because ER is assumed known, there are no parameters, and the maximum value in the numerator of Equation (1) equals to $ER^{R_{1i}}(1-ER)^{R_{0i}}$. Under H_1, there is only one free parameter μ_i. Its maximum likelihood estimate is

$$\widehat{\mu_i} = \left(\frac{R_{1i}}{R_{1i}+R_{0i}} - ER\right)/(1-2ER) \quad (2)$$

When the number of reads is large enough ($N_i = R_{1i}+R_{0i} \to \infty$), the log likehihood ratio statistic $-2lny_i$ follows a χ^2 distribution with one degree of freedom (Mood *et al.*, 1973). Pool i is regarded as a variant pool if H_0 is rejected for a specific significance level. And the locus can therefore be regarded as a variant locus. In our analysis, because for each locus, we need to perform the hypothesis test for tens to hundreds of times depending on the number of pools, we use the false discovery rate (FDR) (Benjamini *et al.*, 2001) for multiple testing corrections.

Because the number of alleles in a pool is a discrete value, one can also estimate μ_i as a discrete variable. Suppose the number of samples in pool i is $w_{(i)}$. All the possible values for μ_i are $\left\{0, \frac{1}{2w_{(i)}}, \frac{2}{2w_{(i)}}, \ldots, 1\right\}$. The elements in this set producing the largest likelihood [the denominator of Equation (1)] is the maximum likelihood estimate of μ_i. In theory,

these two estimates should be very close to each other. Notice that the proposed likelihood test is independent from pool designs. Therefore, it can be used for any pooling strategies, including the non-overlapped design.

2.2.2 Theoretical analysis of the LRT statistic For each locus, the asymptotic power $(1-\beta)$ of the LRT statistic $(-2lny_i)$ can be derived theoretically, as a function of the significance level α, the error rate ER, the VAR μ_i and the locus coverage N_i for pool i. Consequently, for a given α, ER, the required coverage N_i to achieve a desired power $(1-\beta)$ to detect variant pools with VAR μ_i can also be theoretically obtained. Such a relationship can be used in practice to guide experimental designs to determine required coverage. For a given significance level α, the asymptotic power of $-2lny_i$ can be obtained based on its distribution under H_1, which is given by

$$Pr\left(-2lny_i \geq \chi^2_{1,(1-\alpha)} | H_1\right) = Pr\left(\chi^2_1(\delta) \geq \chi^2_{1,(1-\alpha)}\right), \quad (3)$$

where $\chi^2_{1,(1-\alpha)}$ is the critical value corresponding to α under H_0, $\chi^2_1(\delta)$ is the non-central χ^2 distribution with 1 degree of freedom. The non-centrality parameter δ can be represented as a function of locus coverage N_i, error rate ER and VAR μ_i, as a matter of fact, $\delta = N_i(\nu_i - ER)^2/\nu_i(1-\nu_i)$. The detailed derivation of this non-centrality parameter δ is provided in the Supplementary Section 1.1.

To obtain the required locus coverage N_i to achieve a desired power of $1-\beta$ under the alternative H_1 at a significance level α for a specific μ_i and ER, we rely on the following relationships:

$$\begin{aligned} Pr\left(\chi^2_1(\delta) \geq \chi^2_{1,(1-\alpha)}\right) &= 1-\beta, \\ \chi^2_{1,\beta}(\delta) &= \chi^2_{1,(1-\alpha)}, \end{aligned} \quad (4)$$

where $\chi^2_{1,\beta}(\delta)$ is the $100\times\beta$-th percentile of the non-central χ^2 distribution with one degree of freedom and non-centrality parameter δ. Given β and α, the non-centrality parameter δ can be numerically calculated using existing software package to satisfy Equation (4). The value of N_i can be readily obtained based on this equation: $N_i = \nu_i(1-\nu_i)\delta/(ER-\nu_i)^2$.

2.2.3 Incorporating base quality scores For real sequence data, it is difficult to estimate the sequencing error rate accurately. We propose to use read base quality scores provided by sequencing data to approximate locus specific error rates. More specifically, for each locus, and for pool i, let $Q_{j,i}$ and $\varepsilon_{j,i}$ denote the base quality score and the error probability of read j at this locus, respectively. According to the definition of base quality score (Li *et al.*, 2008), we have $\varepsilon_{j,i} = 10^{-Q_{j,i}/10}$. Therefore, Equation (1) can be rewritten as

$$y_i = \frac{\max_{\mu_i \in H_0} \prod_{j=1}^{R_{1,i}} \nu_{j,i} \prod_{j=R_{1,i}+1}^{R_{1,i}+R_{0,i}} (1-\nu_{j,i})}{\max_{\mu_i \in H_0 \cup H_1} \prod_{j=1}^{R_{1,i}} \nu_{j,i} \prod_{j=R_{1,i}+1}^{R_{1,i}+R_{0,i}} (1-\nu_{j,i})} \quad (5)$$

where $\nu_{j,i}$ is the probability of read j having a variant allele at pool i. Under H_0, $\nu_{j,i} = \varepsilon_{j,i}$. Under H_1, $\nu_{j,i} = \mu_i(1-\varepsilon_{j,i}) + (1-\mu_i)\varepsilon_{j,i}$. μ_i can be estimated similarly as described earlier. Because the simulated data do not have base quality score information, Equation (5) is only used on the data generated based on the 1000 genome project.

2.3 Estimation of VAFs

In many downstream applications such as association analysis, allele frequency is one important measure that is commonly used to compare different groups (e.g. cases versus control subjects). It will be desirable if one can reliably estimate allele frequencies using pooling designs. A simple approach to obtain frequencies of variant alleles is to take advantage of the estimated fraction of variant alleles in the identified variant pools at each variant locus (i.e. μ_i in Equation 2). Given $w_{(i)}$ individuals in pool i, the number of variant alleles in pool i can be estimated as $v_i = Round(2w_{(i)}\mu_i)$. For a column-balanced design, every sample participates in exactly the same number of pools (i.e. w). Therefore, the frequency of the variant allele can be calculated as $\mathrm{VAF} = \frac{\sum_{i=1}^{t} v_i}{2nw}$. Such an estimation is somewhat like bootstrap (Efron, 1979). We uniformly sample individuals with repeat into pools, estimate the number of variant alleles in each pool and then average the result from all the pools. In such a case, when increasing the weight, more precise estimation is expected. The accuracy of such an estimation also depends on the accuracy of VAR estimation from each pool. The method can be extended to unbalanced designs as long as the column weights of different samples do not differ much. We also use the same strategy in VIP_Syzygy in estimating MAFs.

2.4 Identification of variant carriers

Once a site is predicted as a variant locus, the next step is to predict which samples carry variant alleles at this site. Pooling designs in principle are (only) effective for variant sample predictions when variant allele frequencies are small, which is usually the case in many studies that seek rare causal disease variants. We discuss this problem in two settings: in an ideal case, all the variant pools are correctly called; in practice, mistakes from variant pool calling (false positives and false negatives) do occur and need special treatment. For the case with no errors, DNA Sudoku (Erlich *et al.*, 2009) has proposed a pattern consistent calling based on the d-disjunct theory (Kautz and Singleton, 1964), denoted as strategy S1 here. It basically says that if the number of individuals with variants is smaller than or equal to the decoding robustness (d) (Erlich *et al.*, 2009; Kautz and Singleton, 1964), then a sample carries a variant if and only if all the pools including this sample as a member are variant pools. When the number of individuals with variants is greater than the decoding robustness, the condition becomes necessary but not sufficient, i.e. when a sample carries a variant, all of its pools must be variant pools; while a sample may not carry any variants even if all its pools are variant pools. In practice, one does not know the number of samples with variants at a locus. One strategy to estimate this number is to use the largest number of samples with all their pools being variant pools. The set of all such samples (denoted as B) is a super set of all variant samples. Based on B, we propose another strategy, denoted as S2. If there is a variant pool that only contains one sample from B, then the variant allele(s) must be from that sample. Therefore, that sample must carry variant allele(s). Figure 2 provides an example to illustrate decoding strategies S1 and S2. The effectiveness of the aforementioned two strategies for variant sample identification critically depends on the accuracy of variant pool calling. When there are false negatives in pool identification, it is possible that at some loci, the number of variant pools is less than the minimum of column weights, and some variant pools have no identified variant samples (using S1 and S2) associated with them. We propose two boosting algorithms to address these issues (details of the two algorithms can be found in Supplementary Section 1.2).

2.5 Data generation

To generate dataset I, we simulate a targeted sequencing project of a region of size 100 kbps with different coverage (10–30×/individual in each pool). For convenience, we randomly choose one region (Chr7: 27 124 046–27 224 045) from the ENCODE project (Birney *et al.*, 2007) because ENCODE regions have been extensively studied, and information about SNP variants (including rare SNPs) is more reliable in those regions. The reference sequence and SNP information in this region are obtained from NCBI's databases. SNP VAFs are either directly available or can be derived based on SNP heterozygosity information. Genotypes of each individual are generated based on VAFs assuming Hardy–Weinberg equilibrium. Once the two homologous sequences of an

individual are generated, they are randomly sheared into short reads with a fixed length. This is done independently for different pools. The number of reads is determined by the expected coverage per sample. To be more realistic, we also randomly introduce sequencing errors in the reads at a fixed rate of 0.005. The short reads are then aligned to the reference sequence using MAQ (Li *et al.*, 2008).

To generate dataset II, we use a subset of real sequence data from the Pilot 3 project of the 1000 Genomes Project (Thousand Genomes Project Consortium, 2010), which has performed high-coverage (56× in average) sequencing of 697 samples from seven populations on 8140 exons from 906 randomly selected genes by multiple institutions using multiple platforms. We only use data generated using Illumina platforms and genotypes called by one institution (i.e. Boston College), which limits the number of samples to 364. We select the largest 41 exons ($\geq$1600 bp) on chromosome 11, with the total length roughly equal to 100 kbps. Only reads within these regions are retained. Reads from potential PCR duplications are removed by applying Picard MarkDuplicates (http://picard.sourceforge.net). Finally, only top 300 samples with larger numbers of reads are used in generating dataset II. The genotypes called from individual samples by the 1000 Genomes Project team (i.e. Boston College for these samples) are regarded as ground truth.

We only focus on the Sudoku design in generating dataset II because it provides better decoding capacity than the logarithm design. When choosing weight six for the 300 samples, the Sudoku design yields a matrix of 145 pools. To generate reads for each pool, we randomly sample reads from each individual assigned to the pool. By doing so, local coverage variation is expected to mimic real sequence data. The total number of reads sampled from each individual in a pool is determined by the total throughput of the pool assuming that all the individuals in the pool contribute equal amounts of reads. The total throughput of the design is determined based on a specified expected individual coverage. It is then averaged out across pools (Supplementary Section 1.3). Reads for different pools are generated independently. Therefore, when an individual participates in multiple pools, different reads from the individual may be assigned to different pools. Data generated by this process are more similar to real data than dataset I and have many features that mimic real data.

3 RESULTS

3.1 Data generated by simulations

For dataset I, we have created two groups of samples: one with 500 individuals (dataset I.1) and the other with 1000 individuals (dataset I.2). For dataset I.1, there are total 1685 variant loci, about half of which (804) can be regarded as rare SNPs (sample VAF $\leq$ 0.05). The smallest VAF is 0.001 and the largest one is 0.537. For dataset I.2, there are total 1694 variant loci, and 803 of them with sample VAF $\leq$ 0.05. The smallest VAF is 0.0005 and the largest one is 0.526.

To evaluate the performance of VIP on dataset I, we adopt two different pool design strategies (i.e. Sudoku design and the logarithm design, denoted as VIP Sudoku and VIP Log) and compare the results with the methods in the original article [denoted as DNA Sudoku (Erlich *et al.*, 2009) and Overlap Log (Prabhu and Pe'er, 2009)]. Because their codes are not available, we implement DNA Sudoku and Overlap Log according to their descriptions in their corresponding articles. We adopt the same set of parameters as those in the articles. For dataset I.1, we use a fixed per individual coverage (10×), and for dataset I.2, we use two different coverages (10 and 30×). For dataset I.1, the logarithm design creates 18 pools, and its column weight equals to nine. By choosing the column weight of seven, Sudoku design yields 210 pools. For dataset I.2, the logarithm design creates 20 pools, and its column weight equals to 10. For Sudoku design, we use the same the column weight of seven, and it yields 270 pools.

For dataset II, the total number of SNPs called by Boston College is 783, and 679 of them with VAF $\leq$ 0.05. The smallest sample VAF is 0.0017. The largest VAF is 1.0000. Only Sudoku design is used. By choosing the column weight of six, Sudoku design yields 145 pools. We use a fixed per individual coverage that is roughly the same as the original data (55×). Based on the results on dataset I, we only compare VIP and VIP_Syzygy on this dataset. To consider the base quality scores based on Equation (5), we choose to use the recalibrated scores by the program GATK (DePristo *et al.*, 2011; McKenna *et al.*, 2010), which are expected to be more reliable than the original ones. We further extract allele counts information using SAMtools' mpileup utility tool (Li and Handsaker, 2009) using base quality score 10 and mapping quality score 20 as filters. We use the same filtering parameters for Syzygy for fair comparison and keep all its other parameters as default.

3.2 *In silico* power of the LRT statistics

Because variant pool identification is the foundation of the proposed framework, in that the identification of variant loci is a direct application of pool identification and variant sample decoding relies heavily on the prediction accuracy of variant pools, we first evaluate the performance of the proposed log-LRT in variant pool identification. Although VAF affects the detection of variant pools, a more direct factor is actually the VAR in each pool, which is affected by both VAF and the number of individuals in each pool. Furthermore, the capacity of the LRT also depends on the sequencing coverage and error rate as illustrated in Section 2.2.2. Because the theory is based on asymptotic analysis, in this subsection, we use *in silico* experiments to illustrate how practical the theoretical relationship between power and other factors is in real cases. Because the accuracy of the LRT for pool identification does not affected by pool designs, for this experiment, we use three non-overlap pool datasets based on dataset I.1, with 25, 50 and 125 pools, respectively. The number of individuals in each pool is 20, 10 and 4, respectively. The minimum VAR in each pool is 0.025, 0.05 and 0.125, respectively. In general, results show that the experimental and theoretical power of LRT is consistent for VAR $\geq$ 0.05. We also observe that for VAR $\leq$ 0.05 and for a fixed coverage, the practical power is much higher than the theoretical one, indicating that the theoretical prediction may be somewhat conservative. Owing to page limitation, more details can be found in Supplementary Section 2.1.

3.3 Identification of variant pools and variant loci

In this and next two subsections, we discuss the performance of VIP Sudoku, VIP Log, DNA Sudoku and Overlap Log on dataset I, and the performance of VIP Sudoku and VIP_Syzygy Sudoku on dataset II in terms of variant pool/locus identification, VAF estimation and variant sample identification.

Table 1 summarizes the results of variant pool and variant locus identification on dataset I and II. For dataset I, because the two designs (Sudoku and logarithm) differ very much in

Table 1. Performance of the algorithms in identifying variant pools and variant loci

Samples and coverage	Method	Variant pools	Variant pools			Variant loci		
			Precision	Recall	*F*-Score	Precision	Recall	*F*-Score
500 and 10× dataset I.1	VIP sudoku	250 753	0.9998	0.9310	0.9642	0.9953	0.9994	0.9973
	DNA sudoku	250 753	1	0.2000	0.3334	1	0.1614	0.2780
	VIP log	29 870	1	0.8925	0.9432	1	0.9484	0.9735
	Overlap log	29 870	0.0164	0.9825	0.0322	0.0169	1	0.0331
1000 and 10× dataset I.2	VIP sudoku	354 158	1	0.9139	0.955	0.9988	0.9988	0.9988
	DNA sudoku	354 158	1	0.1906	0.3202	1	0.1659	0.2846
	VIP log	33 725	1	0.8989	0.9467	0.9994	0.9368	0.9671
	Overlap log	33 725	0.017	0.9908	0.0329	0.0169	1	0.0333
1000 and 30× dataset I.2	VIP sudoku	354 158	1	0.9907	0.9953	1	1	1
	DNA sudoku	354 158	1	0.1893	0.3183	1	0.1647	0.2828
	VIP log	33 725	1	0.9155	0.9559	1	0.9587	0.9789
	Overlap log	33 725	0.0167	0.991	0.0329	0.0169	1	0.0333
300 and 55× dataset II	VIP sudoku	28 164	0.9668	0.7374	0.8366	0.8609	0.8455	0.8531
	Syzygy sudoku	28 164	0.9254	0.6823	0.7855	0.5027	0.8391	0.6287

terms of the number of pools, the total numbers of pools with variants across all loci differ very much (from 30 K to 33 K for data using the logarithm design to ~250–354 K for data using Sudoku design). Regardless of design strategy, sample size or coverage, VIP achieves very high precisions (0.9998–1) and very high recalls (0.8925–0.9907). In contrast, DNA Sudoku is very conservative in calling variant pools. Although its precision is always one, it has very low recalls (0.1893–0.2). Such low sensitivity in detecting variant pools directly results in its low sensitivity to detect variant loci and to identify variant carriers (to be discussed later). On the contrary, Overlap Log has very high recalls (0.9825–0.991) but with very low precisions (0.0164–0.017). Essentially for almost every locus and every pool, Overlap Log calls it as a variant pool, which will create many false positives when identifying variant loci and variant carriers. The comparable performance of VIP on Sudoku and logarithm designs shows that with proper algorithms, variant pool identification is not affected by different designs, as long as the per-sample coverage is kept the same. With higher coverage, the recall of identifying variant pools by VIP Sudoku has increased significantly. The increase of VIP Log is moderate, whereas little improvements have been observed for DNA Sudoku or Overlap Log. For locus identification, similar trends exist for all the approaches. In particular, VIP Sudoku achieves the best overall results with F-score $\geq$0.99 for all cases (Table 1).

For dataset II, Table 1 shows that VIP performs better than Syzygy both in terms of precision and recall on variant pool identification, as well as on variant locus detection. The difference in F-scores for variant pool identification is ~5% (0.8366 versus 0.7855), and the difference is >20% for variant locus identificavtion (0.8531 versus 0.6287). On the other hand, the performance of VIP on dataset II is not as good as its performance on dataset I, mainly because of variability of locus coverage in dataset II (Supplementary Fig. S4A). We therefore examine variation of VIP's performance for different coverage and different VARs, by grouping the pools according to these two measures and calculate the F-Score of variant pool identification for each group. Results (Supplementary Fig. S4B) show VIP has lower F-scores for pools that either have very low coverage or have very low VARs.

3.4 Estimation of VAFs

Because DNA Sudoku does not provide a component for estimating variant allele frequencies, we choose to show the comparison of VIP Log, Overlap Log and VIP Sudoku on estimating VAFs of variant loci on dataset I (Fig. 3A and B), and VIP Sudoku and VIP_Syzygy Sudoku on dataset II (Fig. 3C). Overall, all the approaches are highly accurate in estimating VAFs for both datasets, with the r^2 between the estimated and real VAFs >0.99 on dataset I, and r^2 of 0.958 and 0.888 for VIP Sudoku and VIP_Syzygy Sudoku, respectively, on dataset II. On dataset I, all the approaches have extremely high accuracy when VAF $\geq$0.01. For loci with smaller frequencies, the variance gets somewhat bigger relative to the frequencies themselves, which is common for statistical estimation of small values. Also, for loci with small VAFs, the estimated VAFs by VIP Log and VIP Sudoku are less than the real ones, whereas the estimated VAFs by Overlap Log are greater than the real ones. This can be explained by the conservative variant pool identification by VIP while Overlap Log has reported many more false positives on loci with low VAFs. On dataset II, the performance of VIP Sudoku is a little worse than its own performance on dataset I. This is mainly because dataset II contains higher level of noise than dataset I. However, VIP still outperforms Syzygy and has a better r^2.

3.5 Identification of variant carriers

To evaluate the performance of different approaches on variant sample identification, we only consider the true variant loci. Because VIP Sudoku and VIP Log recall almost all variant loci with high precision on dataset I, this consideration will not affect VIP algorithms much. It does not affect DNA Sudoku,

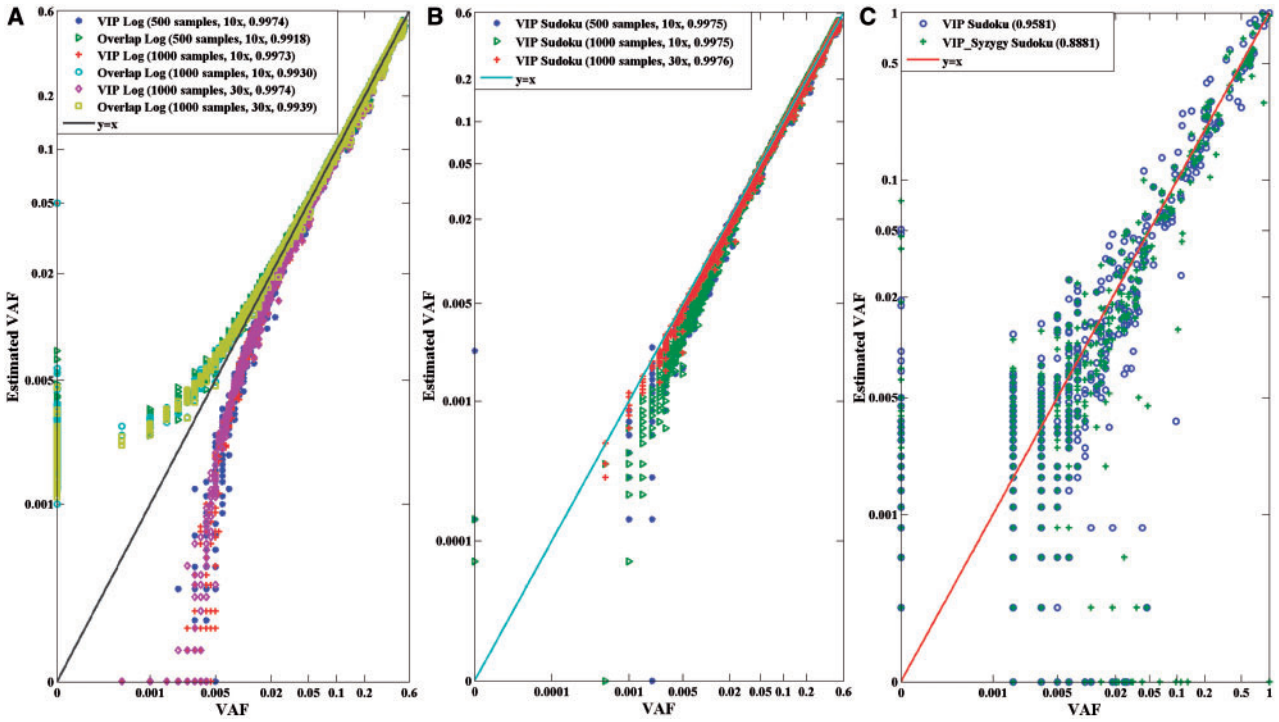

Fig. 3. VAF estimation by VIP Log and Overlap Log on dataset I (A), VIP Sudoku on dataset I (B) and VIP Sudoku and VIP_Syzygy Sudoku on dataset II (C). The parameters and r^2s are also provided in the parentheses

nor does it affect VIP or Syzygy much on dataset II. On the contrary, this treatment is in favour of Overlap Log because the majority of false positives from miscalled variant loci are ignored. Also, only results using the decoding strategy S1 in combination with the two boosting algorithms are presented in the main text. The two boosting algorithms actually greatly enhance the decoding effectiveness of strategy S1. Owing to space limitation, the results using strategy S2 on top of the results from strategy S1 are provided in Supplementary Material. Table 2 shows the precisions, recalls and F-scores of variant sample detection on variant loci of different VAFs. For dataset I, the three algorithms (VIP Sudoku, VIP Log and Overlap Log) have almost the same performance when VAF ≥ 0.2. In such a case (high VAFs), these algorithms basically called a significant majority of samples as variant carriers, therefore their recalls reached 1.0, whereas precisions were only ~0.63. The only algorithm that behaves differently is DNA Sudoku, in which case, its recalls are ~0.50–0.59, whereas precisions are from 0.77 to 0.83. This is a consequence of DNA Sudoku's conservative calling of variant pools. For the same reason, DNA Sudoku cannot identify any variant samples when VAF ≤ 0.2. For VAFs in the range of (0.05, 0.20), VIP Log and Overlap Log obtained the same results, although VIP Log has much better performance in variant pool identification. This basically shows that the logarithm design is not an effective design in terms of variant sample identification. It cannot correctly predict variant samples even when almost all variant pools are correctly predicted, mainly because the number of individuals in each pool is too huge. In the same range, VIP Sudoku achieves comparable recall with the highest precision. For VAF in (0, 0.05), VIP Log and Overlap Log have good recalls but extremely low precisions. In contrast, VIP Sudoku has a much more balanced recall and precision. When the sample size increases from 500to 1000, the trends for different algorithms are different in different VAF intervals. For example, for VAFs in (0, 0.05), both the precision and the recall of VIP Sudoku get lower when increasing the sample size. This is mainly because with the same VAF, the absolute number of variant samples gets large for a large sample size, which makes decoding harder. This point will be further illustrated in Figure 4, when we examine the performance of the algorithms for rare variants at a finer scale. Keeping the sample size the same and increasing the coverage from 10× to 30×, the recalls get improved for most cases, but precisions are a little worse or unchanged.

To further examine the performance of different algorithms on variant samples identification for rare variants, Figure 4 shows the precisions and recalls on loci with VAF < 0.05 with a higher resolution. From the figure, a fact is that VIP Sudoku achieves much higher precision than any other methods on dataset I. VIP Log and Overlap Log have similar performance in most cases. In general, they have higher recalls when VAF ≥ 0.02, but their precisions are very low. For VAF $= 0.01$, where it is more likely that the number of samples with variants is smaller then the decoding robustness, VIP Sudoku also has higher recalls than VIP Log and Overlap Log (except the case 1000 samples 10× coverage). For VIP Sudoku, both precisions and recalls decrease when the sample size increases to 1000 from 500. This is also owing to the fact that for the same VAF, the number of samples with variants gets larger for a larger sample size. When increasing coverage from 10× to 30×, the recalls of VIP Sudoku

Table 2. Performance of different algorithms in identifying variant samples

Method	Variant carriers	0∼0.05	0.05∼0.1	0.1∼0.15	0.15∼0.2	>0.2
500 samples and 10×	195 373					
VIP sudoku		0.6581/0.5080/0.5734	0.3100/0.8421/0.4532	0.2792/0.9739/0.4340	0.3293/0.9957/0.4949	0.6321/1.0000/0.7746
DNA sudoku		–/0/–	–/0/–	–/0/–	–/0/–	0.8368/0.5047/0.6296
VIP log		0.0568/0.8761/0.1067	0.1403/1.0000/0.2461	0.2346/1.0000/0.3800	0.3182/1.0000/0.4828	0.6312/1.0000/0.7739
Overlap log		0.0449/0.8742/0.0854	0.1403/1.0000/0.2461	0.2346/1.0000/0.3820	0.3182/1.0000/0.4828	0.6312/1.0000/0.7739
1000 samples and 10×	390 914					
VIP sudoku		0.3635/0.4954/0.4193	0.1977/0.9225/0.3256	0.2422/0.9945/0.3895	0.3160/0.9993/0.4802	0.6315/1.0000/0.7741
DNA sudoku		–/0/–	–/0/–	–/0/–	–/0/–	0.7734/0.5935/0.6716
VIP log		0.0493/0.9218/0.0936	0.1394/1.0000/0.2447	0.2351/1.0000/0.3807	0.3150/1.0000/0.4791	0.6315/1.0000/0.7741
Overlap log		0.0393/0.9263/0.0754	0.1394/1.0000/0.2423	0.2351/1.0000/0.3750	0.3150/1.0000/0.4701	0.6315/1.0000/0.7510
1000 samples and 30×	3 901 914					
VIP sudoku		0.2902/0.9245/0.4417	0.1700/0.9947/0.2904	0.2374/0.9999/0.3837	0.3150/1.0000/0.4791	0.6315/1.0000/0.7741
DNA sudoku		–/0/–	–/0/–	–/0/–	–/0/–	0.7731/0.5964/0.6734
VIP log		0.0473/0.9444/0.0901	0.1394/1.0000/0.2447	0.2351/1.0000/0.3807	0.3150/1.0000/0.4791	0.6315/1.0000/0.7741
Overlap log		0.0394/0.9290/0.0756	0.1394/1.0000/0.2447	0.2351/1.0000/0.3807	0.3150/1.0000/0.4791	0.6315/1.0000/0.7741
Dataset II	17 236					
VIP sudoku		0.7709/0.4254/0.5483	0.4573/0.5839/0.5129	0.3723/0.6143/0.4636	0.3759/0.9054/0.5312	0.7047/0.9862/0.8220
VIP_syzygy sudoku		0.9514/0.3112/0.4690	0.4585/0.5240/0.4891	0.3657/0.4533/0.4048	0.3671/0.8500/0.5128	0.6999/0.8515/0.7683

Variant loci are grouped into five bins according to their VAFs. Each bin is labelled by its VAF interval indicated in the first row of the table. The last interval for dataset I is 0.2∼0.6, and for dataset II is 0.2∼1. The number of variant carriers is the total number of times that a sample carries at least one variant allele at any locus. Values in other cells indicate the precision/recall/*F*-score of identifying variant carriers on the corresponding VAF interval. The numbers of variant loci in the intervals for dataset I.1 are as follows: 804, 226, 133, 102 and 420. For dataset I.2, they are as follows: 803, 238, 125, 108 and 420. For dataset II, they are as follows: 679, 22, 15, 10 and 57.

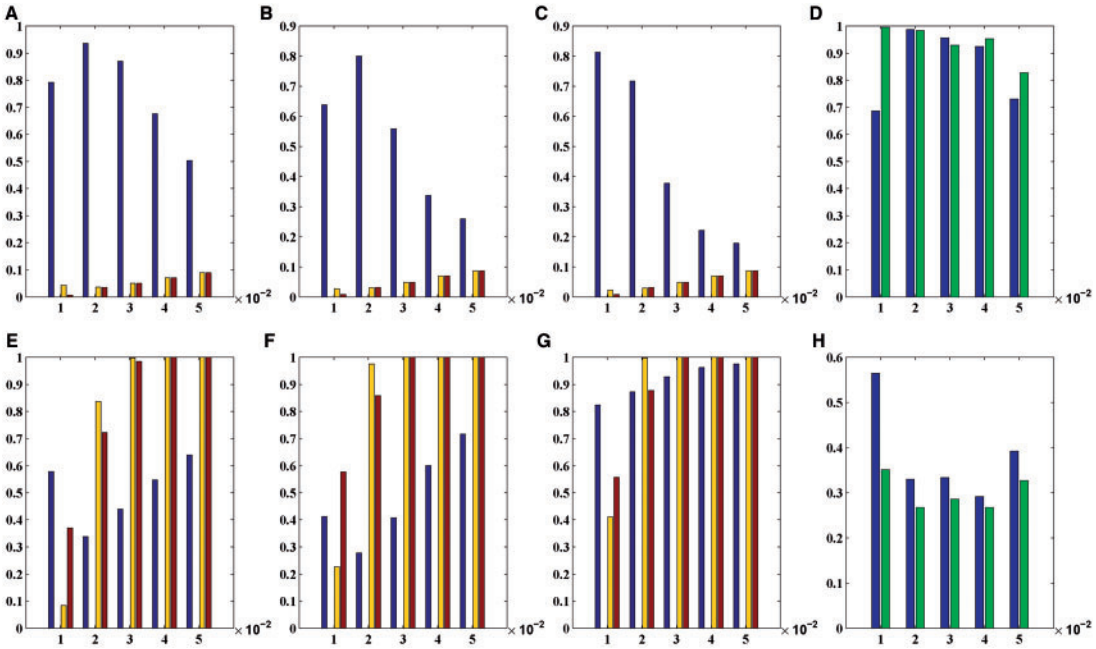

Fig. 4. Variant sample identification by different algorithms on rare variant loci. The subgraphs represent the precisions (top panel) and recalls (bottom panel) for 500 samples with coverage of 10× (A and E), 1000 samples with coverage of 10x (B & F), 1000 samples with coverage 30× (C and G) and 1000 Genome simulated data of 300 samples with coverage 50× (D and H). The *x*-axis is the VAF of variant loci. At each VAF in panels A–C and E–G, the vertical bars show performance of VIP Sudoku, DNA Sudoku (0 for all VAFs), VIP Log and Overlap Log, respectively. For panels D and H, the two bars show the performance of VIP Sudoku and VIP_Syzygy Sudoku, respectively. For 500 samples, the numbers of variant loci in each interval are: 272, 231, 118, 86, 97; For 1000 samples, the numbers of variant loci in each interval are: 270, 239, 119, 92, 83. For 1000 Genome simulated data, the numbers of variant loci in each interval are: 567, 49, 30, 19, 14

increase with comparable precisions. Also, with the increase of VAF, the recalls of VIP Sudoku gets lower when VAF ≥ 0.02. Overall, VIP Sudoku performs the best in variant sample detection for dataset I and has highest F-scores on all cases. In particular, it is well suited to decode rare variant carriers. Precision and recall of VIP, DNA Sudku and Overlap Log on dataset I in higher resolutions can be found in Supplementary Figures S5 and S6.

For dataset II, results in Table 2 show that VIP Sudoku performs better than VIP_Syzygy Sudoku in identifying variant samples for all VAF intervals in terms of F-scores. For loci with VAF >0.05, VIP and VIP_Syzygy have almost the same precisions, but VIP always has better recalls. For loci with VAF ≤ 0.05, VIP has a better recall (0.425 versus 0.311), but VIP_Syzygy has a better precision (0.951 versus 0.771). We further examine the precisions and recalls of the two approaches on variant loci with VAF ≤ 0.05 with higher resolution in Figure 4 D and H and Supplementary Figure S5D and H). Results show that VIP has better recalls on all the intervals and comparable precisions for $0.005 < \text{VAF} \leq 0.05$. The only interval that VIP has a lower precision is for VAF ≤ 0.005, in which case there are at most three variant alleles in the samples.

4 DISCUSSION AND CONCLUSION

In this article, we have presented a complete data analysis package, VIP, mainly for rare SNP discovery and calling from data generated based on overlapping pool designs. VIP itself consists of several algorithms, including variant pool and variant locus identification, VAF estimation and variant sample detection. It can be combined with any design matrices. For the proposed LRT for variant pool identification, we also investigated the quantitative relationship between power, coverage and VAR, for a given significance level and sequencing error rate. We have performed extensive experiments on simulated data and on a dataset generated based on real sequence data from the 1000 Genome project to demonstrate the effectiveness of VIP. In summary, VIP significantly outperforms all three approaches considered in this study and is very effective in identifying variant pools and variant loci, and in estimating VAFs. Furthermore, in combination with the Sudoku design, VIP Sudoku is also more effective in identifying variant samples than other approaches.

In terms of costs, in most cases, pooling is cheaper than no pooling, mostly because of savings in target capturing (See Supplementary Section 2.5 for more discussions). Among the two designs tested, the logarithm design has the least overall costs, but it is not effective in identifying variant samples. Another possible strategy is to combine the logarithm design with customized SNP arrays to type the samples after rare SNPs being discovered. The relative merit of such a strategy will be explored further and will be compared with the VIP Sudoku algorithm in future studies. In addition to overlapping pool designs, the frequency estimation algorithm in VIP can be used in other biological applications. For example, viruses in patients naturally consist of many different strains/species. It is important to examine population diversity of viruses in patients and their correlations with treatment effectiveness. Many SNP mutations in virus populations will have small frequencies. One can use the proposed algorithm and its extensions to discover SNPs in viruses and estimate their frequencies.

In our derivation, we did not consider possible sampling bias generated from uneven amount of DNAs from different samples within a pool and in different pools. Bias can also be generated from non-uniform amplifications of DNAs from different samples. We will take these possible biases into account in our future work. In addition to SNPs, indels are another type of important genetic variations. Indel identification from pooled samples will also be one of our future working directions.

One limitation of pooling designs is that they cannot identify carriers of more common SNPs. Recently, He *et al*. (2011) shed light on this issue. By taking advantage of linkage disequilibrium information and prior information on genotypes of some common SNPs from the same samples, one can potentially impute the genotypes on common SNPs. Therefore, genotypes of both rare SNPs and common SNPs can be recovered. We will work on improving the performance on common variant sample detection by incorporating external information into our framework.

ACKNOWLEDGEMENT

The authors thank the reviewers for their constructive comments.

Funding: This research is supported by National Institutes of Health/National Library of Medicine grant LM008991.

Conflict of Interest: none declared.

REFERENCES

Altshuler,D.M. *et al.* (2005) A haplotype map of the human genome. *Nature*, **437**, 1299–1320.

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

Bansal,V. *et al.* (2010) A statistical method for the detection of variants from next-generation resequencing of DNA pools. *Bioinformatics*, **26**, i318–i324.

Benjamini,Y. *et al.* (2001) The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.*, **29**, 1165–1188.

Birney,E. *et al.* (2007) Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, **447**, 799–816.

Calvo,S.E. *et al.* (2010) High-throughput, pooled sequencing identifies mutations in NUBPL and FOXRED1 in human complex I deficiency. *Nat. Genet.*, **42**, 851–858.

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

Ding,C. *et al.* (1996) *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*. World Scientific, Singapore/River Edge, NJ.

Druley,T.E. *et al.* (2009) Quantification of rare allelic variants from pooled genomic DNA. *Nat. Methods*, **6**, 263–265.

Du,D. and Hwang,F. (2000) *Combinatorial Group Testing and its Applications*. World Scientific, Singapore/River Edge, NJ.

Efron,B. (1979) 1977 Rietz lecture—bootstrap methods—another look at the jackknife. *Ann. Stat.*, **7**, 1–26.

Erlich,Y. *et al.* (2009) DNA Sudoku-harnessing high-throughput sequencing for multiplexed specimen analysis. *Genome Res.*, **19**, 1243–1253.

Frazer,K.A. *et al.* (2007) A second generation human haplotype map of over 3.1 million SNPs. *Nature*, **449**, 851–861.

He,D. *et al.* (2011) Genotyping common and rare variation using overlapping pool sequencing. *BMC Bioinformatics*, **12** (Suppl. 6), S2.

Kautz,W. and Singleton,R. (1964) Nonrandom binary superimposed codes. *IEEE Trans. Inf. Theory*, **10**, 363C377.

Kim,S.Y. *et al.* (2010) Design of association studies with pooled or un-pooled next-generation sequencing data. *Genet. Epidemiol.*, **34**, 479–491.
Lee,J.S. *et al.* (2011) On optimal pooling designs to identify rare variants through massive resequencing. *Genet. Epidemiol.*, **35**, 139–147.
Li,H. *et al.* (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, **18**, 1851–1858.
Manolio,T.A. *et al.* (2009) Finding the missing heritability of complex diseases. *Nature*, **461**, 747–753.
McCarthy,M.I. *et al.* (2008) Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nat. Rev. Genet.*, **9**, 356–369.
McKenna,A. *et al.* (2010) The genome analysis toolkit: a mapreduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.
Momozawa,Y. *et al.* (2011) Resequencing of positional candidates identifies low frequency IL23R coding variants protecting against inflammatory bowel disease. *Nat. Genet.*, **43**, 43–47.
Mood,A.M. *et al.* (1973) *Introduction to the Theory of Statistics*. McGraw-Hill, NY, USA.
Nejentsev,S. *et al.* (2009) Rare variants of IFIH1, a gene implicated in antiviral responses, protect against type 1 diabetes. *Science*, **324**, 387–389.
Nijman,I.J. *et al.* (2010) Mutation discovery by targeted genomic enrichment of multiplexed barcoded samples. *Nat. Methods*, **7**, 913–967.
Out,A.A. *et al.* (2009) Deep sequencing to reveal new variants in pooled DNA samples. *Hum. Mutat.*, **30**, 1703–1712.
Prabhu,S. and Pe'er,I. (2009) Overlapping pools for high-throughput targeted resequencing. *Genome Res.*, **19**, 1254–1261.
Rivas,M.A. and Daly,M.J. (2011) Syzygy Documentation. Release 1.1.0. http://www.broadinstitute.org/software/syzygy/sites/default/files/Syzygy_2011.pdf (July 2012, date last accessed).
Rivas,M.A. *et al.* (2011) Deep resequencing of GWAS loci identifies independent rare variants associated with inflammatory bowel disease. *Nat. Genet.*, **43**, 1066–1073.
Li,H. *et al.* (2009) The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*, **25**, 2078–2079.
Smith,A.M. *et al.* (2010) Highly-multiplexed barcode sequencing: an efficient method for parallel analysis of pooled samples. *Nucleic Acids Res.*, **38**, e142.
Thousand Genomes Project Consortium. (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.
Vallania,F.L.M. *et al.* (2010) High-throughput discovery of rare insertions and deletions in large cohorts. *Genome Res.*, **20**, 1711–1718.
Wang,T. *et al.* (2010) Resequencing of pooled DNA for detecting disease associations with rare variants. *Genet. Epidemiol.*, **34**, 492–501.
Wei,Z. *et al.* (2011) SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data. *Nucleic Acids Res.*, **39**, e132.
Wellcome Trust Case Control. (2007) Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, **447**, 661–678.