Genomic Imputation in ultra low coverage sequencing data of Ashkenazi jews

Noam Bar

Advisor: Dr. Shai Carmi

August 2016

ABSTRACT

Genome wide association studies (GWAS) is currently the main approach to identifying the genetic basis of complex diseases and traits. GWAS gain their statistical strength by using a vast number of samples, ranging from hundreds to tens of thousands. Although sequencing technology improved unrecognisably over the past decade, deep coverage WGS (whole genome sequencing) of a single sample, is still relatively expensive. One approach of tackling this issue, is to use low coverage sequencing combined with genomic imputation of the missing sites. In this project, we used the Ashkenazi Jewish (AJ) population as a model of an isolated population, and developed a fast tool for imputing the haplotypes of a sample sequenced to extremely low coverage, using HMM based algorithms. Our algorithm used 128 deep sequenced genomes of AJ as it's reference panel, covering a large proportion of the population's genetic variation. Preliminary results shows an increase of up to 40 fold in the effective sample size for a fixed budget of 100,000$, when using the developed tool.

INTRODUCTION

Much of the genetic base of diseases known today, is due to the vast use of genome wide association studies (GWAS). While finding the genetic base of mendelian diseases (diseases caused by a mutation in a single genomic locus) is a relatively easy task, requiring only a small number of diseased individuals, since all of them must hold the mutant locus, finding the genetic base of complex (multifactorial) diseases and traits such as diabetes, heart disease and obesity, requires a large number of diseased individuals in order to pinpoint a statistical associated to genetic variation.

A major problem researchers face when designing such studies, is the high cost of deep coverage sequencing of the large number of samples. Followed by the development of high-throughput sequencing technology (Next Generation Sequencing), the cost of whole genome sequencing dropped dramatically. Still, high coverage whole genome sequencing is quite expensive (currently 2-3k $). A common alternative is microarray genotyping, together with statistical inference (also known as imputation) of untyped variants (Howie et al. PLoS Genet, 2009). However, since microarrays hold a fix set of genetic markers, the imputation is not optimal. Furthermore, the price per sample is still relatively high.

Since sequencing cost is determined by the cost of sample preparation (which is fixed) and the cost of the sequencing itself (which is ~linear in coverage depth), it implies that a decrease in the sequencing depth, will be followed by an increase in the number of samples one can sequence for a fixed budget. Indeed, low coverage sequencing followed by imputation is another alternative facing the issue presented above. For very large sample sizes, ultra-low coverage sequencing (ULCS; 0.1-1x) was recently shown to result in a ~3 fold larger effective sample size (see *Methods*) than microarray genotyping (Pasaniuc et al., Nat Genet, 2012). In addition, since exome sequencing (~0.25x) is widely used, and samples are already available by the thousands, imputation tools may be useful in order to recover the genomic sequences lying outside the exome using the off target reads, as it was recently shown (Pasaniuc

et al., Nat Genet, 2012), hence associating new genomic variation with diseases and traits.

The process of calling genotypes from low coverage data is usually divided into two steps:

(a) Genotype likelihoods are calculated where sequenced reads are available.

(b) The remaining sites are imputed based on linkage disequilibrium and population reference data (Li et al. Genome Res, 2011).

While in microarray and in medium-low coverage sequencing most sites are captured, in ULCS data, only a small fraction of sites are sequenced. Hence, accurately imputing the missing sites requires a large number of captured sites spanning long genomic stretches. At first, this might seem as a downfall, since long genomic blocks are usually shared only between relatives. But recent studies had shown that in some founder populations, even unrelated individuals share long genomic blocks (Kong et al. Nat Genet, 2008) (>3cM (~3M bps)). Those blocks, called *identical-by-descent* (IBD), are inherited from very recent, though previously unrecognized, common ancestors. Those IBD blocks may be useful for the purpose of imputing genotypes, since each sparsely sequenced sample might be fully genotyped along the genomic regions shared with the high coverage sequenced reference sample. So in theory, in isolated populations, a tool that could identify IBD blocks between ULCS data and high coverage reference panel, would be able to infer missing sites with high accuracy, and thus would be of great value in terms of achieving larger effective sample sizes for GWAS.

One such isolated population is the Ashkenazi Jews (AJ), a Jewish diaspora population who formed as a distinct community of Jews in the Holy Roman Empire around the end of the 1st millennium. The AJ avoidance of marriage and reproduction with surrounding populations contributed greatly for keeping their genomic variation at a minimal level. Furthermore, abundant IBD sharing within the AJ population was observed, an observation that is associated with a major founder effect (a population bottleneck) occurred ~25-35 generations ago. This founder effect event included a population with an effective size of ~300-400 individuals (Carmi et al., Nat Comm, 2014)

(see *Figure 1*). This event in the early history of the AJ led to a strong genetic drift, which caused chance increase in the frequency of Mendelian diseases-causing alleles (Charrow. Fam Cancer, 2004). Risk alleles which were associated with complex diseases were observed as well in AJ, a fact that further increases the relevance of AJ as a population suitable for diseases mapping.

The crucial step of creating a large reference panel of AJ genomes, was lately accomplished (Carmi et al., Nat Comm, 2014). 128 AJ genomes sequenced to high coverage (>50x) were used in order to call haplotypes accurately (see *Methods*). With an additional ~600 AJ genomes recently sequenced, and with the assumption that the AJ founder population effectively numbered only hundreds, this database is close to cover the majority of variation that existed in the time of the bottleneck.

A

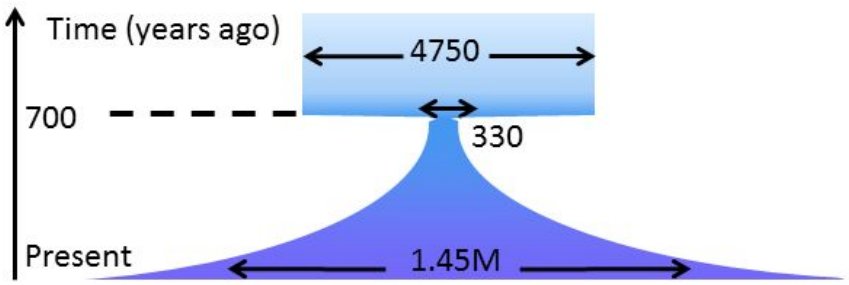

Figure 1. (A) The recent AJ demographic history. Horizontal arrows correspond to effective population sizes and are given in number of diploid individuals; times were computed assuming 25 years per generation.

RESULTS

Reference panel and testing data

A critical part of any imputation tool, is the reference panel. For an imputation tool to be able to infer missing sites accurately, it needs a reference panel that covers the majority of the genetic variation inside the relevant population. In this project I used 128 AJ high coverage sequenced genomes as the reference panel. This reference panel consists of 12,183,564 SNPs (no indels are included) across the 22 non sex

chromosomes, spreading the variants at average distance of ~235 bp. The reference panel includes diploid individuals and their genotypes were phased into haplotypes, still switch errors occur regularly throughout the genome. In order to test and build our algorithm, we used high coverage sequencing data of Ashkenazi jewish trio (parents and child) taken from *Genome in a bottle* consortium (see *Methods*). Trio's files were downloaded as Sequence Alignment/Map (SAM) format, and were down sampled randomly in order to achieve the desirable low coverage sequencing files. In addition, for each sample in the trio, a Variant Call Format (VCF) file was downloaded, and used as ground truth (real genotype) in order to compare the imputing algorithm's results.

Algorithm outline

The algorithm gets as input the low coverage sequencing files (currently SAM format files) of a sample, and outputs a sequence of genotypes corresponding to the set of variants appearing in the reference panel variants set (RPVS). The algorithm begins by parsing the sequencing files and performing variant calling, specifically over variants appearing in the RPVS. As mention before, long IBD segments are shared between individuals from the population. In order to find those segments, the algorithm then divides the genome into 3 sets of overlapping windows (see *Methods*). For each window, the algorithm computes a score for each one of the 128 individuals in the reference panel. The score is calculated over the sequenced variants appearing in the RPVS, compared to the genotypes in the reference panel (created by joining the two haplotypes of every individual), using the log likelihood ratio test (see *Methods*). Next, the algorithm picks, for each window, the two best scoring individuals out of the reference panel to serve as surrogate mother and father, where each final genotyped site in that window will be inferred as a result of a combination of haplotypes from its corresponding surrogate parents. Than, Using HMM based algorithms, and the selected pair, the algorithm determine the most likely genotype of every missing site in the window.

Bimodal distribution of the algorithm's scores

In order for the algorithm to find the individuals that most likely share IBD segment in each specific window, we must be able to distinguish between individuals that scored poorly compared to those who scored best. Investigating the distribution of scores created by the algorithm, it seems that in most windows, the scores creates a bimodal distribution (see *Figure 2*), separating the reference panel into two major subpopulations. The first sub-population consists of most of the individuals in the reference panel, and it has lower mode compared to the second subpopulation. The second subpopulation, usually consists of a small number of individuals, and having a higher mode, it is suspected of sharing an IBD segment across the window. This bimodal distribution is encouraging, since it implies that when there exists an IBD sharing at some window, our scoring method is able to separate the IBD sharing samples from the rest.

A

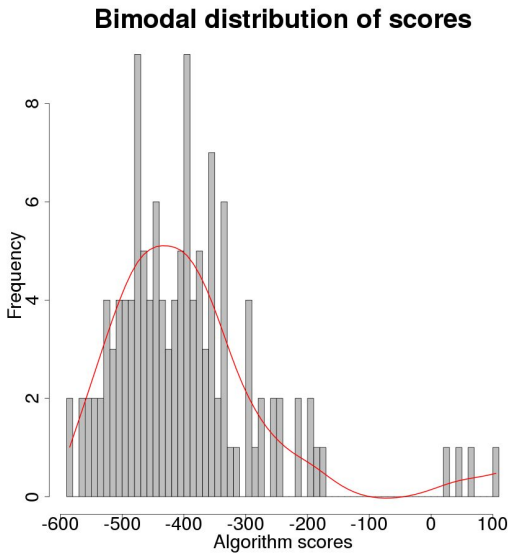

Figure 2. (A) An example for a bimodal distribution of the algorithm's scores. Results are shown for window number 20, in chromosome 1 of the trio's mother, when using an alignment file down sampled to coverage 1x. The left subpopulation has a lower mode and consists of 124 individuals, while the right subpopulation achieves a significantly higher mode and is consisted of only 4 individuals.

Sanity checks

An important step in the development of the algorithm, is performing sanity checks, in order to see that there are no bugs in the program, and to get a clear view of the algorithm's realistic capabilities. First basic test, includes computing correlations between the algorithm scores and the naive shared allele count per window at different

coverages (see *Figure 3*). Although the number of matching alleles is not the only factor that determines the algorithm's score, it is assumed to be significantly correlated with it. If this is not the case, it may imply that the relative weight of each single variant in a window is too high, affecting the score dramatically, and making the scoring depends heavily on capturing rare variants, damaging the algorithm robustness. Examining the correlations between the naive matching allele count and the algorithm corresponding scores, it appears that across different coverages, correlation values ranged between ~0.6-0.8. Even when running the algorithm on high coverage samples (60x) the correlations does not go above ~0.7. This shows that the algorithm scoring method is robust to different coverages input, and that the naive count of matching alleles in each window plays a major role in determine the algorithm resulted output. Correlations were calculated using the data of the AJ trio.

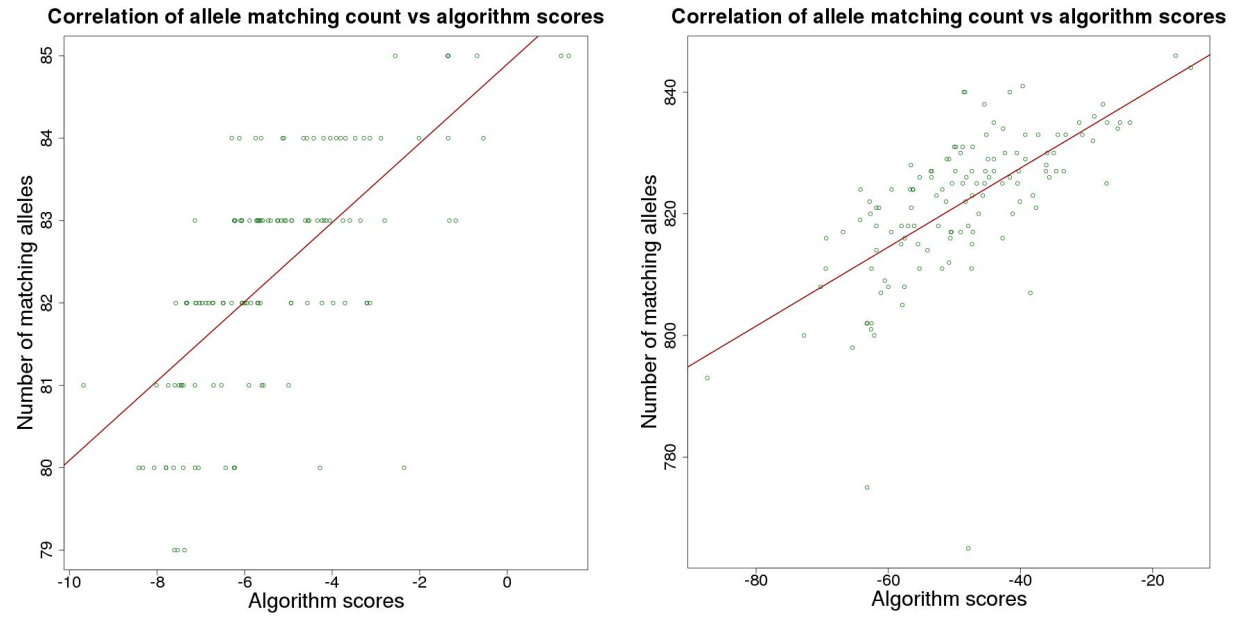

Figure 3. (A) Correlation of allele matching count vs algorithm scores in window 3 of chromosome 1 of trio's child, downsampled to 0.01x; correlation = 0.67. (B) Correlation of allele matching count vs algorithm scores in window 3 of chromosome 1 of trio's child, downsampled to 0.1x; correlation = 0.64.

Another important sanity check, involves examining the AJ trio. Since every child inherits one chromosome from each one of his parents, and since the trio includes parents and their child, it must hold that if for each window, if there is an individual that shares an IBD segment with the child, then at least one of his parents should share the IBD segment with that individual as well. In order to check this, we counted for each of the child's chromosomes, how many times (in how many windows) did the best scoring reference individual appeared in the five top scoring reference individuals of either the mother or the father (or both) (see *Table 1*). This was done with ranging coverages samples. We observed that as the coverage drops, the percentage of windows in each chromosome for which the algorithm found the best scoring reference individual within the five best scoring reference individuals of either the mother or the father (or both), drops as well. While in 1x this percentage is ~75%, in 0.2x it drops to ~65%, and in 0.01x it is as low as ~20%, indicating, that in coverage as low as 0.2x the algorithm still functions adequately.

Scores correlation across coverages

One of the principal questions regarding the algorithm's robustness to coverage, is: How low could the input's coverage be, while maintaining the ability to identify the most similar individuals from the reference panel, in each window. In order to answer that, we calculated the correlation of scores across all windows, between the algorithm scores for high coverage input against the algorithm scores for down sampled inputs (see *Figure 4*). We observed that at coverage as low as 0.1-0.2x, much of the information (correlation = ~0.8-0.9) is kept, implying that the algorithm could find the most similar individuals using sequencing inputs with coverage as low as 0.1x. It is important to mention though, that those correlations do not represent the accuracy of the imputation, rather reflecting the ability of the algorithm to work with low coverage inputs.

Choosing the two individuals for each window

The step of choosing the two individuals, from which the final genotypes will be inferred is crucial, since every possible output that the algorithm may produce,

A

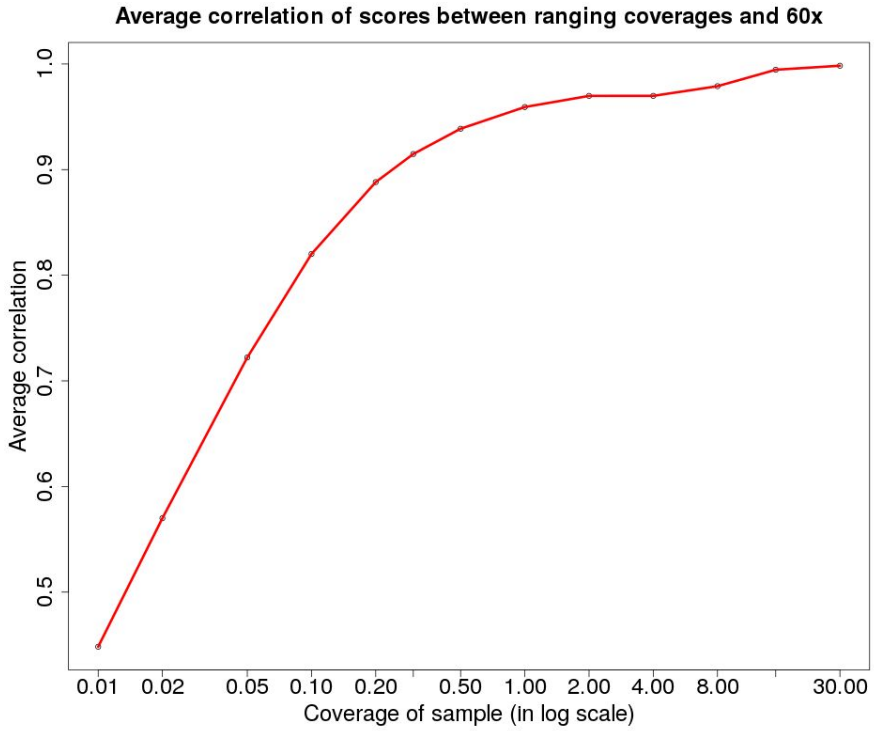

Figure 4. (A) The Average correlation between scores outputted by the algorithm when it ran with 60x coverage input, and a set of down sampled ranging coverages inputs. This was calculated using the trio's child as input for the algorithm, and is presented in log scale. Observing the graph, it seems that at coverage of 0.2x much of the data reachable by the algorithm is captured.

corresponds to a certain combination of those chosen individuals haplotypes. The first pick is straightforward, the algorithm chooses the individual that scored best in each window, since under the assumption that the algorithm's scoring method performs well, it is most likely that this individual shares an IBD segment across that window. In order to make the second pick, the algorithm uses a precalculated table, that holds for each individual in the reference panel a rank of the 127 other individuals in each window of each chromosome. In other words, given a window and an individual, the table holds a sorted list of the most similar individuals at that window. This table was created using the scoring method of our algorithm with the haplotypes in the reference panel as input. Using this precalculated table, and given the index of the first pick, the algorithm chooses the next best scoring individual, provided that it is not in the first pick's list of

five most similar individuals. If all five next best scoring individuals are in the former list, the algorithm chooses the second best scoring individual. This procedure is done because a sample may be heterozygous or homozygous in a specific window. If it is heterozygous, we want to choose the two best scoring individuals, that do not share an IBD segment between themself. If it is homozygous, than the two top scoring individuals would probably share an IBD segment between themself, hence taking the second best scoring individual would be the wiser choice.

Using HMM based algorithms in order to infer the missing sites

The heart of algorithm and its final stage includes using the previously chosen individuals in order to infer the missing sites at each window. This stage is done using Hidden Markov Model (HMM) based algorithms, which can take into account every possible combination of haplotypes derived from the chosen individuals. After picking the two individuals, we are left with four haplotypes, while the final output should be two haplotypes (humans are diploids), which could be any combination of one haplotype from each of the individuals. One may consider the two individuals as “surrogate parents”, transmitting to their child (the sample) one haplotype each. If the haplotype phasing in the reference panel was accurate, then all we had to do is calculate which combination of entire haplotypes is most likely to generate the observed sequenced variants. Since the phasing stage in the reference panel is not error prone, we must take into account that phasing switches may occur. This is where the HMM come in handy. The model’s hidden states are the four different combinations of haplotypes (each representing some genotype) (see *Figure 5*). The model’s visible observations are the sequenced variants in the RPVS. Changing states is equivalent to having a phasing switch, hence, between each two states there is a transition probability, that depends on the distance between the two variants (in bps), and the estimated number of phasing switch errors in the reference panel. The transition probability of phase switching in both parents, although very rare, is simply the product of the transition probabilities of each phasing switch. The probability of observing the sequenced variants, giving each state, is followed by binomial sampling and estimated sequencing

errors (see *Methods*). Considering the above description, it is possible to construct the relevant HMM for each window, using the two picked surrogate parents. Using Viterbi algorithm we can efficiently find the chain of states that achieves the highest likelihood, and from it we can deduce the corresponding genotypes in the missing sites. Furthermore, using the Forward-Backward algorithm, we can calculate the posteriors of every genotype in each site. This method is especially appreciated in association studies, where researchers take into account the probabilities of the appearing genotypes. Since the algorithm initially divides the genome into three sets of overlapping windows, every variant (except the ones in the edges) appears in three different windows. Hence, it is possible to use majority vote, in which the final output is determined by the majority rule, in our case, the final genotype is determined as the one that appears at least in two windows out of the three (using Viterbi), or alternatively, the one that has the highest combined posterior (using Forward-Backward).

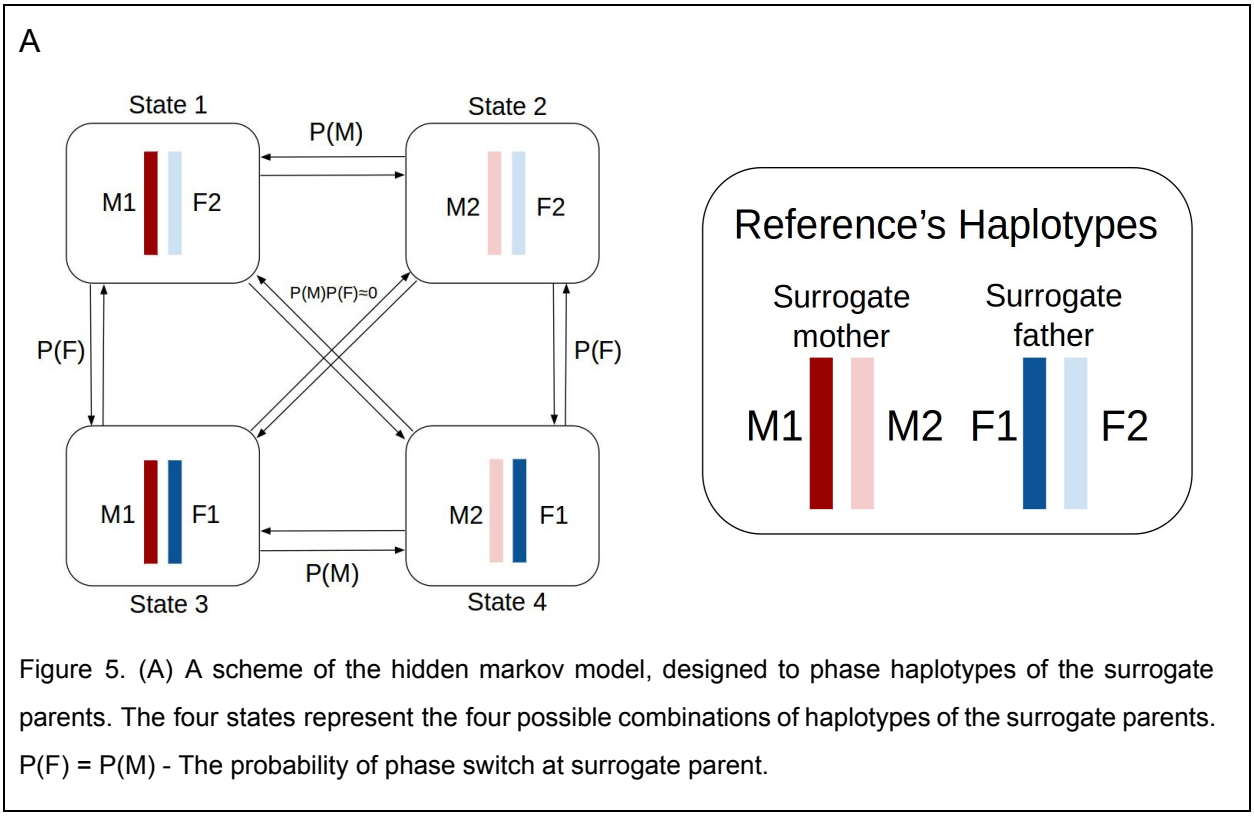

Figure 5. (A) A scheme of the hidden markov model, designed to phase haplotypes of the surrogate parents. The four states represent the four possible combinations of haplotypes of the surrogate parents. P(F) = P(M) - The probability of phase switch at surrogate parent.

Increase in effective sample size

The original objective of this project is increasing the effective sample size of GWAS for a fixed budget. This is done by sequencing a large number of samples at low coverage (hence reducing the costs), and inferring the missing sites with the tool we developed. Since the imputation is not error prone, the number of samples we sequence does not equal the effective sample size, but rather there is a need to normalize this number by a factor taking into account the error rate of the imputation process. We down sampled the trio's alignment files, in order to create ranging coverages files (see *Methods*). We then ran the algorithm with those files as inputs. The phasing stage was done using Viterbi algorithm in majority vote mode. The use of the majority vote method turned out to be effective, although the difference between using it and using standard Viterbi algorithm for a selected set of non overlapping windows was rather small (few percent improvement). The output genotypes were compared against the VCF files downloaded from *Genome in a bottle* consortium - these files were treated as ground truth, since they are a product of several different sequencing, variant calling, and phasing methods. We calculated the algorithm's percentage of imputing success, for each type of genotype (homozygous reference allele, homozygous alternative allele, heterozygous) (see *Figure 6*). We used two constant controls, the first one was "dumb": for every input, it assigned all variants as homozygous reference allele. The second control was a bit "smarter": given any input, it assigns each variant with the most common genotype from the reference panel. It is clear that the smarter control was far better at predicting the genotype, mostly due to the fact that for heterozygous and homozygous alternative allele variants, the dumb control was always wrong. According to the results, in coverages >0.01x the algorithm performs better than the smart control, in addition it seems that at coverage of 0.1x the algorithm reaches maximum efficiency. Out of the homozygous reference variants, the algorithm accurately imputed over 96% even when working with low coverage sequencing inputs. Out of the heterozygous variants, the algorithm accurately imputed over 40% within low coverage. And out of the homozygous alternative, the algorithm success rate was around 50% through all coverages, and higher than the smart control's even at coverage as low as 0.01x. One

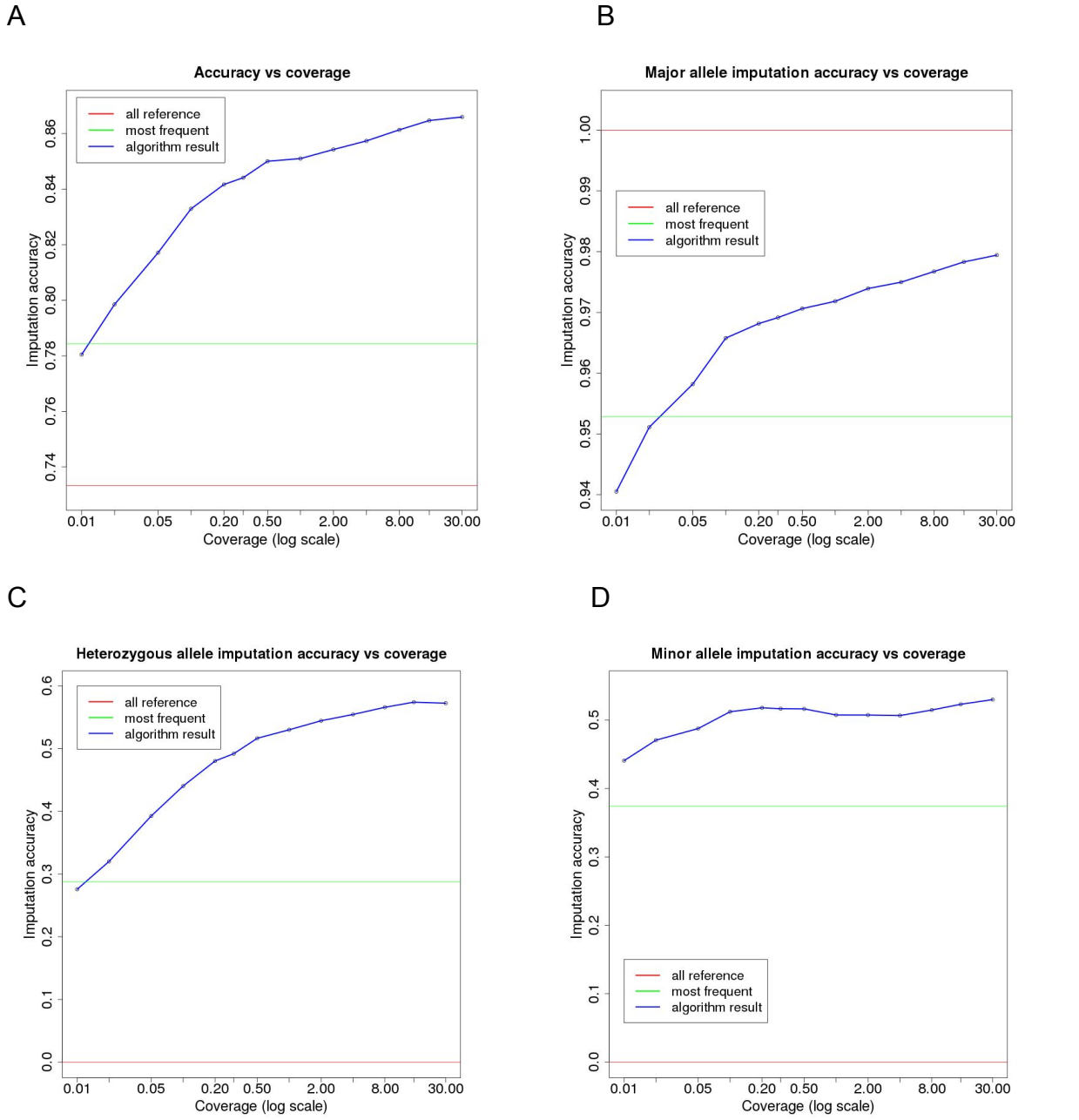

Figure 6. (A) Accuracy of the algorithm imputing the variants of the trio's child in chromosome 1. (B) Accuracy of the algorithm imputing the homozygous reference variants. (C) Accuracy of the algorithm imputing the heterozygous variants. (D) Accuracy of the algorithm imputing the homozygous alternative variants.

possible explanation for the accuracy rates of homozygous alternative variants, is that when working with ultra low coverage there is not enough evidence of homozygous alternative variants, hence, the algorithm will consider them as heterozygous and will

pick reference individuals that are heterozygous at that variant. Since the VCF files holds variants not appearing in the RPVS, the algorithm will always return a false answer for them. Meaning that, the algorithm success rate, when considering only variants appearing in the RPVS, are higher (by 6-7%). When looking at the combined results (all types of variants) of the algorithm, we can again notice that accuracy rates rise as the used coverage rises. Still, in very low coverage there seems to be enough information for the algorithm to identify individuals that share an IBD segment to some extent.

Finally, we present the effective sample sizes accepted when using the above results, for the range of coverages (see *Figure 7*), for a fixed budget of 100,000$, using reasonable assumptions regarding the sample preparation and sequencing costs (Pasaniuc et al., Nat Genet, 2012). It appears that for a coverage of 0.05x, maximum efficiency is achieved. Also there is an increase of ~40 times in the effective sample size, assuming the alternative includes sequencing to high depth (coverage ~30x) and achieving perfect variant calling (no correction to the sample sizes needed). This results

A

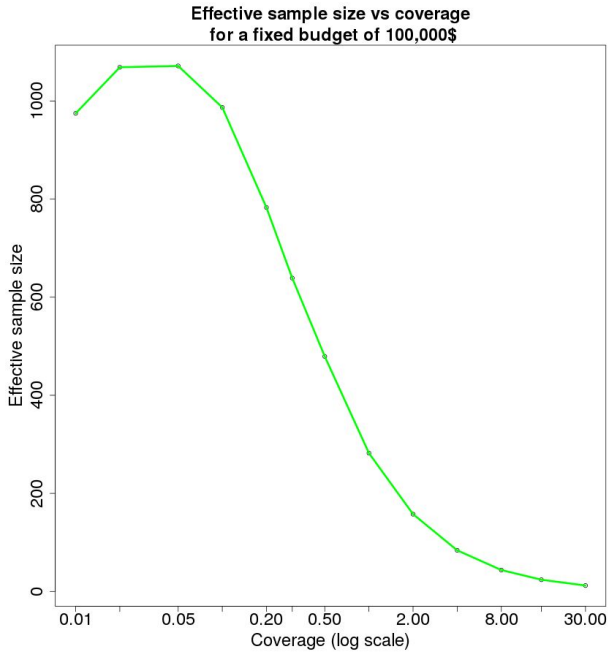

Figure 7. (A) Shown is the effective sample sizes (ESS) obtained by using the results of the algorithm on the trio's child in chromosome 1. The maximum of the ESS is achieved at coverage 0.05x.

are similar to other published methods, which is encouraging, since we used 128 genomes as our reference panel compares to 1000 in (Pasaniuc et al., Nat Genet, 2012).

DISCUSSION

Researchers who plan GWAS face major issues when it comes down to costs. Sequencing to high coverage the large number of samples needed for statistical inference is expensive and in a sense wasteful. In this project, we developed a tool aiming to increase the effective sample size, by inferring missing genomic sites in low coverage sequencing data. Our algorithm includes dividing the genome into windows, finding a pair of individuals most likely to share a genomic block with the sample, from which it will deduce the missing genotypes using HMM based algorithms. We used the AJ population as a model of isolated populations. Our algorithm used 128 AJ genomes recently sequenced to high coverage as a reference panel. The algorithm passed some sanity checks, and our tests implied that even when using very low coverage data (~0.1x), there is still enough information in order for the algorithm to identify IBD sharing individuals. Our algorithm imputed the missing sites with high accuracy, and for a coverage of 0.05-0.1x it achieved maximum efficiency in terms of increment of the effective sample size.

As it was shown before, using statistical inference it is possible to increase the effective sample size. Our algorithm shows performances similar to other methods published in the past (Pasaniuc et al., Nat Genet, 2012). The algorithm's implementation is basic and primal, and there are many improvements we can introduce to it that will effectively increase its accuracy. We discuss later some of these improvements.

The size of the reference panel on which the algorithm's implementation is based is 128. With increment in the reference panel size, more new variants will be added to the RPVS, and a larger portion of the AJ genomic variation will be covered. There are additional ~600 AJ genomes sequenced to high coverage, that are now in haplotype

phasing stage. Once those genomes will be added to the reference panel, the algorithm's ability to accurately impute missing sites will increase significantly (Carmi et al., Nat Comm, 2014).

About ~26% of variants in the RPVS are singletons, means, the alternative allele was found in only one haplotype out of the 256 haplotypes (128 individuals * 2 haplotypes). Those variants, following being rare, contributes greatly to the algorithm's scoring function. Finding a match for such a variant, heavily implies that the reference individual shares the genomic block surrounding the variant. Nevertheless, those variants are harder to phase, hence, changes to the HMM's transition probability function should be introduced (perhaps random chance of transition), in order to cope with situations where the relevant variant is a singleton. One possible change we can introduce to the algorithm, is assigning higher transition probabilities around singletons, in order to take into account potential phasing errors. Alternatively, we might eliminate singletons from the HMM and once the sequence of states is determined, we can place the singletons into the haplotype of the nearest SNP.

The algorithm is currently implemented such that it does not pay any special attention to singletons. Introducing the above changes is predicted to improve the algorithm accuracy at phasing level.

Every output the algorithm creates is a combination of the genotypes in the reference panel. So, the space of possible outputs is bounded by the variation existing in the reference panel. As we have shown, about 7% of the variants of the trio's child did not appear in the RPVS, making the algorithm incompetent of assigning them correctly. With an increase of the reference panel and the RPVS accordingly, this percentage is expected to get lower, increasing the algorithm success rates. The effective sample size achieved by the algorithm, could increase by as much as 50% (see *Figure 8*).

The results presented here were computed, using the rather simple Viterbi algorithm. The use of the majority vote method, over using Viterbi algorithm with one set of non overlapping windows, is rather efficient. Despite that, the difference of accuracy

is rather low, yet not neglectable. More advanced methods are implementable, such as the Forward-Backward algorithm which computes the posterior of each genotype for every variant. Major vote method for posteriors is also rather easy to implement, it would return the genotype having the highest combined posterior. Furthermore, working with probabilities, instead of fixed genotypes, is of greater value for studies trying to find an association between a variant and a trait.

A

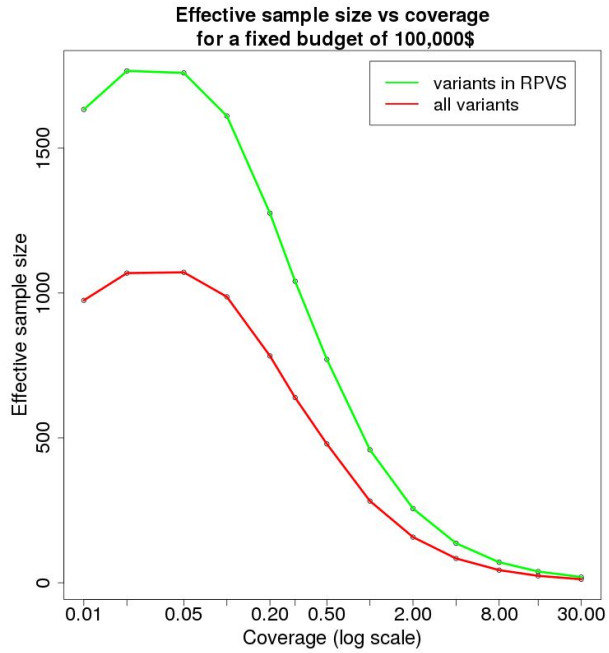

Figure 8. (A) Shown is the effective sample sizes (ESS) obtained by using the results of the algorithm on the trio's child in chromosome 1. The green curve correspond to variants in the RPVS. The red curve correspond to all variants in chromosome 1. There is an increase of ~47% in the maximum values achieved.

There are more clever ways of picking the pair of individuals in each window from the reference panel. According to the method described above, the individual with the highest score, always get picked, while the second one is picked according to the ranking of individuals between themselves. An optimization over the size of the first pick's most similar sub list (currently five), could be done in order to improve results. In addition, it is possible to examine the distribution of scores in each window, and divide the whole reference panel into two subpopulations, following by picking the entire subpopulation that achieves the higher average score. Next, using the entire subpopulation, in order to build a HMM with number of states, corresponding to the number of combinations of haplotypes within that subpopulation. Performing this will

increase the running time of the algorithm, but bounding the size of the subpopulation picked, the algorithm's running time will remain in the same magnitude.

In conclusion, implementing some of the issue proposed here could significantly improve the algorithm's accuracy rates. Using this developed tool, may help shed light over yet undiscovered associations between variants and traits/diseases. In addition, the tool may be applied to AJ exome data from one cohort of ~500 individuals with diabetes and another cohort of ~150 clinical pediatric genomes.

METHODS

Effective sample size

Suppose the true genotype is unknown, and we have an estimated genotype with correlation of r^2 between the true genotype and the estimated one. To achieve the same power (to identify associations) as a true genotype sample of size N, the sample size must be increased by a factor of $\frac{1}{r^2}$ (Pritchard et al. Am J Hum Genet, 2001).

Reference panel and simulated data

The 128 AJ genomes making up the reference panel, were sequenced using *Complete Genomics* to average coverage >50x, in three batches. The data was phased using *SHAPEIT*. The genomes were mapped to reference version hg19. (Carmi et al., Nat Comm, 2014)

The AJ trio's data was taken from *Genome In a Bottle* consortium. The samples were sequenced in a variety of different methods. We used the files sequenced by Illumina HiSeq to coverage 300x. Mapping to the genome was done using Novoalign. VCF files were created over three different technologies: Complete Genomics, Illumina HiSeq and Ion Proton. (For further information see *Genome In a Bottle*)

Window's division and length distribution

Breaking chromosomes into three sets of non overlapping windows is done as follows: Given the start of one window, the next break is determined by counting at least

3 centiMorgan and at least 3,000,000 bps. Information regarding the linkage distances were taken from *HapMap* project. When windows were close to centromeres sometimes found within the chromosome sequence, they were assigned with sizes smaller or bigger than mentioned. Centromeres coordinates were taken from *UCSC* Genome Browser. Finally, in order to create an effect of three sets of overlapping windows, after finding the first set of non overlapping windows, the second (and third) set, is determined by one (two) thirds the way between the end of the current window, to the end of the next window. The distribution of the window's lengths, implies that most of the windows are of length ~3MB (see *Figure 9*).

A

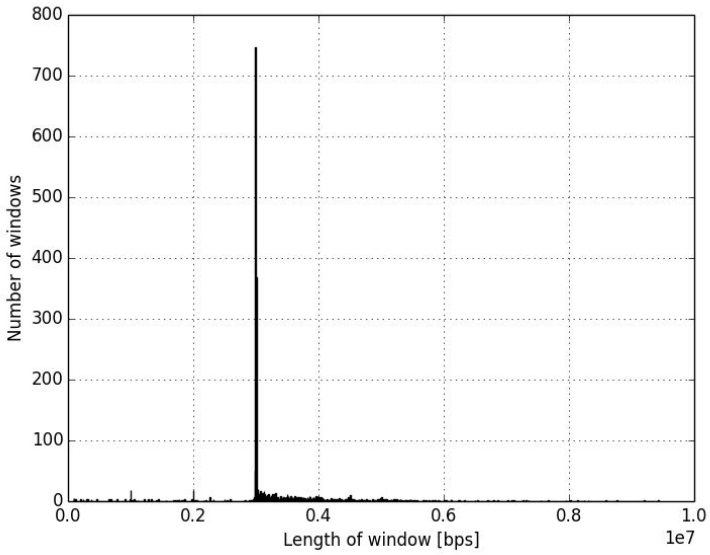

Figure 9. (A) Distribution of window sizes across all chromosomes. Shown are windows lengths from all the three overlapping sets.

Algorithm's scoring function

In every window, the algorithm scores each one of the reference panel's individuals. The Scoring function uses likelihood ratio test, which compares two hypotheses. One claims that the low coverage sequenced sample and the reference panel's individual do not share ancestry. Hence, the probability of observing these genotypes at random depends only on the frequency of the alleles in the population. The other claims that both samples share a recent common ancestor; hence, the

probability of observing similar genotypes is somewhat higher, and also depends on the frequency of the alleles in the population. Therefore, the scoring function is $log\frac{P(X|IBD)}{P(X|no-IBD)}$, where $P(X|IBD)$ is the probability of observing the two genotypes giving that they share ancestry, and $P(X|no-IBD)$ is the probability of observing the two genotypes giving that they do not share ancestry. In low coverage, most times it is not possible to determine in high certainty the diploid genotype of the sequenced sample; hence, we divided the probabilities table into two cases (see *Table 2*): 1. The full genotype is known. 2. One allele is masked.

HMM emissions and transitions

The transition function as described above reflects the chance of having a phasing switch in the reference panel. When computing this probability, we consider the phasing errors existing in the reference panel, and the distance between the two observed variants. Denote by d the distance between two observed variants, each representing a hidden state. In the reference panel, phasing errors occur every 30,000 bps in average. Now, since the average distance between two adjacent variants in the RPVS is ~270 bps, the probability of having a phasing switch somewhere between the two variants (neglecting singletons) should be $270/30,000 \approx 0.01$. So the probability of not having a phasing switch between two adjacent variants is $1-0.01=0.99$, on average. When considering the distance (d) between the two adjacent observed variants, we get that the probability of having a phasing switch somewhere between the two, is $p=1-0.99^{d/270}$. When the distance between two variants is too large, making $p>0.5$, the transitions between all states is set to 0.25 (equal probability). Denoting the surrogate mother's haplotypes by M1,M2, and the surrogate father's haplotypes by F1,F2, the transition table is as follows:

Case (example)	Probability of transition
F1M1 $\rightarrow$ F1M1 (no switch)	$(1-p)^2$

F1M1 → F1M2 (one switch)	$p(1-p)$
F1M1 → F2M2 (double switch)	p^2

The emission function reflects the probability of observing the sequenced reads at a specific site, giving the surrogate parent's genotypes. The function follows binomial sampling and estimated sequencing errors, which we estimated as 0.01 (1%). Denote the number of reads covering a specific site by n, n_a reads of allele A, and n_b reads of allele B. In addition, consider ε the sequencing error probability. The emissions table is as follows:

Induced genotypes at shared haplotypes	Probability of observation
AA	$C(n,n_a)(1-\varepsilon)^{n_a}\varepsilon^{n_b}$
AB	$C(n,n_a)2^{-(n_b+n_a)}$
BB	$C(n,n_a)(1-\varepsilon)^{n_b}\varepsilon^{n_a}$

C(n,k) stands for the binomial coefficient indexed by n and k (n choose k).

Down sampling alignment files

The original sequencing files of the trio, had ~300x coverage. We used ~60x coverage files created by taking the first 20% of the reads. In order to create the down sampled files, we used python's *random* function in order to randomly sample reads from the 60x alignment file. The coverage of the down sampled files are calculated as $\frac{total-number-of-bps}{3{,}200{,}000{,}000}$.

REFERENCES

S. Carmi, K.Y. Hui, E. Kochav, X. Liu, J. Xue, et al. Sequencing an Ashkenazi reference panel supports population-targeted personal genomics and illuminates Jewish and European origins. Nat Commun 5, 4835 (2014).

B.N. Howie, P. Donnelly, and J. Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet 5, e1000529 (2009).

B. Pasaniuc, N. Rohland, P.J. McLaren, K. Garimella, N. Zaitlen, et al. Extremely low-coverage sequencing and imputation increases power for genome-wide association studies. Nat Genet 44, 631 (2012).

Y. Li, C. Sidore, H.M. Kang, M. Boehnke, and G.R. Abecasis. Low-coverage sequencing: implications for design of complex trait association studies. Genome Res 21, 940 (2011).

A. Kong, G. Masson, M.L. Frigge, A. Gylfason, P. Zusmanovich, et al. Detection of sharing by descent, long-range phasing and haplotype imputation. Nat Genet 40, 1068 (2008).

J. Charrow. Ashkenazi Jewish genetic disorders. Fam Cancer 3, 201 (2004).

Pritchard, J. K. & Przeworski, M. Linkage disequilibrium in humans: models and data. Am J Hum Genet 69, 1 (2001).

SUPPLEMENTARY TABLES

chromosome	Number of windows	son-father	son-mother	son-both	% covered
1	69	25	24	3	0.66666666667
2	69	27	25	2	0.72463768115
3	56	21	22	3	0.71428571428
4	54	19	18	0	0.68518518518
5	51	22	17	1	0.74509803921
6	48	15	19	4	0.625
7	46	15	15	2	0.60869565217
8	40	15	16	1	0.75
9	38	15	14	3	0.68421052631
10	41	21	12	3	0.73170731707
11	39	10	21	3	0.71794871794
12	40	16	17	4	0.725
13	28	13	7	1	0.67857142857
14	27	11	7	0	0.66666666667
15	26	12	4	0	0.61538461538
16	25	5	9	1	0.52
17	24	6	7	1	0.5
18	24	7	12	0	0.79166666666
19	19	10	3	0	0.68421052631
20	19	5	8	1	0.63157894736
21	13	5	4	1	0.61538461538
22	12	5	4	1	0.66666666667

Table 1. Shown for each chromosome, the number of times the algorithm identified the best scoring individual of the child within the five best scoring individuals of the father, the mother and both. This table represents the results when using coverage of 0.2x. The percentage is calculated as follows: $\frac{(son-father)+(son-mother)-(son-both)}{the\ number\ of\ windows}$.

A

Case	Reference	Proband	P|No-IBD	P|IBD
C1	AA	AA	P^4	P^3
C2	AA	AB	$2P^3Q$	P^2Q
C3	AA	BB	P^2Q^2	0
C4	AB	AA	$2P^3Q$	P^2Q
C5	AB	AB	$4P^2Q^2$	PQ
C6	AB	BB	$2PQ^3$	PQ^2
C7	BB	AA	P^2Q^2	0
C8	BB	AB	$2PQ^3$	PQ^2
C9	BB	BB	Q^4	Q^3

B

Cases from above	Reference	Proband	P|No-IBD	P|IBD
C1+C2/2	AA	A	P^3	$P^3+P^2Q/2$
C3+C2/2	AA	B	P^2Q	$P^2Q/2$
C4+C5/2	AB	A	$2P^2Q$	$P^2Q+PQ/2$
C6+C5/2	AB	B	$2PQ^2$	$PQ^2+PQ/2$
C7+C8/2	BB	A	PQ^2	$PQ^2/2$
C9+C8/2	BB	B	Q^3	$Q^3+PQ^2/2$

Table 2. P - Frequency of allele A in the population. Q - Frequency of allele B in the population. (A) Case when all alleles are known. (B) Case when one allele is masked.