An Efficient and Accurate Graph-Based Approach to Detect Population Substructure

Srinath Sridhar*, Satish Rao** and Eran Halperin***

Abstract. Currently, large-scale projects are underway to perform whole genome disease association studies. Such studies involve the genotyping of hundreds of thousands of SNP markers. One of the main obstacles in performing such studies is that the underlying population substructure could artificially inflate the p-values, thereby generating a lot of false positives. Although existing tools cope well with very distinct sub-populations, closely related population groups remain a major cause of concern.

In this work, we present a graph based approach to detect population substructure.Our method is based on a distance measure between individuals. We show analytically that when the allele frequency differences between the two populations are large enough (in the l_2-norm sense), our algorithm is guaranteed to find the correct classification of individuals to sub-populations.

We demonstrate the empirical performance of our algorithms on simulated and real data and compare it against existing methods, namely the widely used software method `STRUCTURE` and the recent method `EIGENSTRAT`. Our new technique is highly efficient (in particular it is hundreds of times faster than `STRUCTURE`), and overall it is more accurate than the two other methods in classifying individuals into sub-populations. We demonstrate empirically that unlike the other two methods, the accuracy of our algorithm consistently increases with the number of SNPs genotyped. Finally, we demonstrate that the efficiency of our method can be used to assess the significance of the resulting clusters. Surprisingly, we find that the different methods find population sub-structure in each of the homogeneous populations of the HapMap project. We use our significance score to demonstrate that these substructures are probably due to over-fitting.

1 Introduction

Studying the etiology of common complex disease such as cancer, or Parkinson's disease, is an important task in the search for better treatments and diagnosis tools for these diseases. A common practice towards

* Computer Science Dept, Carnegie Mellon University. Email: srinath@cs.cmu.edu.
** Computer Science Dept, University of California, Berkeley. Email: satishr@eecs.berkeley.edu.
*** International Computer Science Institute (ICSI), Berkeley. Email: heran@icsi.berkeley.edu.

this task is to perform an association study, in which the genetic variation of a set of cases (individuals carrying the disease) and a set of controls (background population) is compared, and large discrepancies between the two populations indicate an association of a specific locus with the studied phenotype.

There are different forms of genetic variations that can be studied in the context of association tests, the most common one is single nucleotide polymorphisms (SNPs), which are nucleotides in the genome that are found to be varying among different individuals. In general, these SNPs are bi-allelic, that is only two alleles are found in the population. SNPs are commonly used in association studies, as the SNP variation is believed to capture most of the human genetic variation [4–6], and furthermore, recent technology (e.g. Affymetrix or Illumina) allows the genotyping of hundreds of thousands of SNPs per individual for a couple of hundred dollars. Thus, whole genome association studies, in which hundreds of thousands of SNPs are genotyped for thousands of individuals is becoming a common practice.

The validity of the results of an association study heavily depends on the statistical analysis performed. One of the main growing concerns is that population substructure may raise spurious discoveries. In association studies, the discrepancies in the SNP-allele frequencies between the cases and the controls are believed to imply an association of the SNP with the disease, but if the cases and controls were collected from two very different populations, this discrepancy may be explained by the difference between the two populations, and hence the SNP is not necessarily associated with the disease. Even subtle differences in the population structures of the cases and the controls may result in spurious associations. In particular, this problem is becoming more acute when large scale association studies are performed(see for e.g., [8, 15]).

There are many computer programs that try to cope with this problem, most notably the widely used software `STRUCTURE` [13] and a recently developed method `EIGENSTRAT` [14]. `STRUCTURE` uses a Markov Chain Monte Carlo (MCMC) approach to find population substructure of a given population using DNA variation data. `EIGENSTRAT` is based on principal component analysis (PCA). Mathematically, this problem can be seen as a clustering problem, in which the different clusters correspond to different populations. Such clustering problems have been studied under a variety of different theoretical frameworks that share close similarities. For instance the max-cut of a graph shares properties with the eigenvectors of the corresponding (adjacency or Laplacian) matrix and therefore Spectral methods or PCA, `STRUCTURE` and finding max-cuts of graphs share close mathematical relationships [1–3, 12].

`STRUCTURE` has been used extensively in genetic studies (cited more than 700 times), and it has been shown to find population substructure quite accurately in many examples. Even though `STRUCTURE` performs very well in terms of accuracy, it is quite inefficient, and it may take weeks to run over one whole genome data-set. Furthermore, even though `STRUCTURE` outputs a likelihood score which assists in interpreting the results, it is not clear whether this likelihood score can be used to determine whether there is actually a significant presence of population substructure. Fi-

nally, as the MCMC is inherently a heuristic approach, it is hard to know which parameters to set for the algorithm; in particular, as we show in this paper, there is no uniform set of parameters that performs well for all the data-sets.

In order to cope with these problems, we introduce a new graph based method for clustering populations. We concentrate in this paper on the clustering of two populations, although the method can be easily extended to multiple populations. Our technique is based on a simple paradigm. We define a distance between every pair of individuals, and we then search for a maximum cut in the graph induced by these distances. From that cut, we perform a local search that maximizes the likelihood of the data, similar to the criterion used in `STRUCTURE`. The main advantage of our method is that it is extremely efficient, and at the same time very accurate. Furthermore, since eventually the algorithm optimizes the same score function as that of `STRUCTURE`, it can be viewed as a fast method that finds a local optimum for this criterion.

It is important to note that the efficiency of our method allows us to measure the significance of the population substructure by running our algorithm on thousands of permutations of the data. For instance, we find that both our method and `STRUCTURE` find a population substructure in the YRI population, genotyped by the HapMap project [10]. On the other hand, after the permutation test, we observe that the p-value is 0.75, indicating that this partition is probably just an artifact. Since `STRUCTURE` is too slow to perform such a test, our method gives a rigorous alternative to the significance estimators of `STRUCTURE`.

We measured the performance of our method and compared it to `STRUCTURE` on the HapMap populations, as well as on simulated data. We find that our method is at least as accurate as `STRUCTURE`, and an order of magnitude more efficient. Furthermore, we find that the accuracy of `STRUCTURE` degrades when many SNPs are used (thousands), while the accuracy of our method consistently improves when the number of SNPs increases.

We have also compared our method to `EIGENSTRAT`, a recent program that corrects for population stratification using the eigenvalues of the genotype covariance matrix [14]. In [14] they suggest a method based on principle component analysis, that assigns each individual a vector representing its ancestral composition. Although their method is not specifically designed for clustering populations, we have adapted their method in a natural way and compared it to the method developed in this paper. We found that `EIGENSTRAT` is quite efficient, but it appears not to perform very well on many of our datasets. We believe that this is due to the fact that the principal component analysis fails when the sub-populations structures are not independent.

Technically, our method is based on a distance defined between pairs of individuals. There are many possible distance measures, and the resulting algorithm is very sensitive to the choice of the distance measure. Surprisingly, one of the most natural measures, i.e., the Hamming distance, performs quite poorly. We therefore use as a starting point the mother-father distance defined in [3]. This measure satisfies the property that the expected distance between two individuals drawn from the same sub-population is zero while the distance between individuals from

two different sub-populations is positive. Furthermore, in [3] it is shown that the max-cut induces the correct partitioning asymptotically, at least when the sub-population sizes are equal. Our final distance uses a more complicated procedure which takes into account the genotypes of the whole population in order to determine the distance between a pair of individuals. We show empirically that this procedure is advantageous and that the resulting distance better represents the population structure. This distance measure may be of independent interest, as it may be used in other population based applications.

2 Problem Formulation

We consider the setting in which a set of n individuals are genotyped over m SNPs. The problem of population stratification focuses on the assignment of each of the individuals to a population cluster. In practice, an individual could belong to more than one cluster, (for instance when the individual's ancestors come from two or more different populations). In this paper, however we concentrate on the simpler case, in which each individual is assumed to belong to exactly one population. Furthermore, we assume that the number of populations K is known. We will observe later that this assumption is not too restrictive, as one can test for the validity of the solution. Our goal is to cluster the set of individuals into K clusters, based on their genotype information.

In order to define the problem mathematically, we first introduce a random generative model for the individuals' genotypes. Each genotype is represented by a vector $g \in \{0,1,2\}^m$, where g_j represents the minor allele in SNP j, that is, $g_j = 1$ for heterozygous, and it is 0 and 2 for the homozygous major or minor alleles respectively. A population is characterized by the minor allele frequency in each of the SNPs. Thus, a population i is defined by an m-dimensional vector $\boldsymbol{p^i} = (p^i_1, \ldots, p^i_m)$, where p^i_j represents the minor allele frequency of population i in position j. The random generative model assumes that all individuals are sampled independently, and that for each individual g, the different SNP values are sampled independently, where g_j is sampled from the distribution $\{(p^i_j)^2, 2p^i_j(1-p^i_j), (1-p^i_j)^2\}$ (e.g., the probability that $g_j = 1$ is $2p^i_j(1-p^i_j)$). This model has been used by previous approaches, and in particular by `STRUCTURE`. The assumption that the different SNPs are independent can be justified if the SNPs are physically distant from each other (and thus, they are in *linkage equilibrium*). We define the distance between two sub-populations i, i' as:

$$d(i,i') = \sqrt{\sum_j (p^i_j - p^{i'}_j)^2}$$

Formally, we assume that we get as an input an $n \times m$ genotype matrix A, where the rows $R(A)$ denote diploid individuals and the columns $C(A)$ represent SNP sites. Each entry in A is in $\{0,1,2\}$. We search for a classification $\theta : R(A) \to \{1, \ldots, K\}$, that assigns every individual to a particular sub-population. Let $\hat{\theta}$ be the correct classification. Our

objective is to minimize the number of errors made by the algorithm, that is, we would like to minimize $|\{r \in R(A) \mid \theta(r) \neq \hat{\theta}(r)\}|$.

2.1 The Graph Based Approach

It is convenient to think of the above problem as a clustering problem in a graph. In this case, we construct a complete graph $G = (V, E)$, where vertex set V corresponds to the set of individuals, and edge set E is the set of all pairs of individuals. We assign a distance for each edge, which will intuitively represent the genomic distance between the two individuals. Then, the main idea of the algorithm is to find a max-K-cut in the resulting graph. This makes sense since G captures the fact that the genomic distance between two individuals from the same sub-population is small, while the distance between two individuals from different sub-populations may be large. Clearly, the resulting algorithm is sensitive to the choice of the distance measure.

The most natural distance measure is the Hamming distance, which counts the number of differences between the two vectors. However, we observe that in practice the Hamming distance does not provide very good results[1]. We therefore follow [3], and start from the so called *Mother-Father distance (MF)*. The MF-distance satisfies the property that the expected distance between two individuals from the same population group is 0 and the expected distance between two individuals of different populations is positive. Actually, it is not hard to see that the MF-distance is the only pair-wise distance measure that satisfies this property (up to a constant factor).

Formally, for any two individuals r_1, r_2, we define $\delta_j(r_1, r_2)$, the MF-distance at SNP j as follows. We set $\delta_j(r_1, r_2) = -1$ if $r_{1j} = r_{2j} = 1$, $\delta_j(r_1, r_2) = 2$ if $r_{1j} = 0, r_{2j} = 2$ or $r_{1j} = 2, r_{2j} = 0$, and 0 otherwise. We then define the MF-distance $\delta(r_1, r_2)$ to be the sum of the MF-distances over all SNPs. That is, $\delta(r_1, r_2) = \sum_j \delta_j(r_1, r_2)$.

We can now compute the expected distance between two individuals r_1, r_2 from populations i and i' $E[\delta(r_1, r_2)]$:

$$
\begin{aligned}
&= \sum_j E[\delta_j(r_1, r_2)] \\
&= \sum_j 2(p_j^i)^2(1-p_j^{i'})^2 + 2(p_j^{i'})^2(1-p_j^i)^2 - 2p_j^i(1-p_j^i)(2p_j^{i'}(1-p_j^{i'})) \\
&= 2\sum_j (p_j^i(1-p_j^{i'}) - p_j^{i'}(1-p_j^i))^2 = 2\sum_j (p_j^i - p_j^{i'})^2 = 2d(i,i')^2
\end{aligned}
$$

Consequently, if $i \neq i'$, then the expected MF-distance between two individuals of different populations is positive. On the other hand, if $i = i'$, then $d(i, i') = 0$, and the expected MF-distance between two individuals of the same population is zero. It is further shown in [3], that if the distance between two different sub-populations $d(i, i') >>$

[1] For example, when $p_j^1 = 2/3, p_j^2 = 1$, the expected distance within population 1 is larger than the expected distance across

$\sqrt{1.5}(m \log n)^{0.25}$, then with high probability, all the pair-wise distances within a sub-population are at most $d(i,i')^2$, while pair-wise distances across the two sub-populations are at least $d(i,i')^2$. In that case, the max-K-cut algorithm may be reduced to a connected component algorithm. Furthermore, it can be shown that even with much smaller separation of the two populations, the max-cut on the graph with MF-distances produces the correct cut [3].

2.2 Triplets-based distance

Even though the MF-distance has some very nice properties, our empirical studies (see Appendix A) show that the max-cut solution obtained from this distance is sometimes biased towards an unbalanced partition. Intuitively, although the expected value of the MF-distance is monotone with the distance between the populations, unbalanced cuts may be chosen by the algorithm by pure chance. It is therefore essential to find a distance measure that has smaller variance than the MF-distance.

We build on top of MF-distances to obtain a more sensitive distance measure, which we call the triplet distance. The main idea of the triplet measure is to utilize information from all genotypes to determine the distance between a pair of individuals.

We will now formally define the triplet distance for a pair of individuals r_1 and r_2. The triplet distance depends on two parameters a, b that will be fixed later. For every third individual in the population, r, we consider the unordered set $\{r_1, r_2, r\}$, which we refer to as a *triplet*. For each such triplet, we define two indicator variables X_r and Y_r such that $X_r = 1$ if $\delta(r_1, r_2) \geq \max(\delta(r_1, r), \delta(r_2, r))$, and $X_r = 0$ otherwise. Similarly, $Y_r = 1$ if $\delta(r_1, r_2) \leq \min(\delta(r_1, r), \delta(r_2, r))$, and it is zero otherwise. We define the triplet-based distance as $d_{a,b}(r_1, r_2) = \sum_r (aX_r + bY_r)$. In other words, to compute the triplet distance of r_1 and r_2, we consider every third individual r and if $\delta(r_1, r_2)$ is the largest among the three MF-distances, then we add a and if it is the smallest, then we add b.

We now find the expected triplet distance $d_{a,b}(r_1, r_2)$ for a pair of individuals r_1, r_2. We will implicitly assume that all MF-distances are different (this is true if the number of SNPs is sufficiently large). For a triplet (r_1, r_2, r), we consider the following two cases. First, assume that all three individuals are from the same population. Then by symmetry, $\Pr(X_r = 1) = \Pr(Y_r = 1) = \frac{1}{3}$. Otherwise, if r_1, r_2 are from population i, and r is from another sub-population i', we will bound the probability $\Pr[\delta(r_1, r_2) \geq \delta(r_1, r)]$. Intuitively, this probability should be small if the distance $d(i, i')$ is large enough. Formally, we know that

$$E[\delta(r_1, r_2) - \delta(r_1, r)] = -2d(i, i')^2.$$

Furthermore, $\delta(r_1, r_2) - \delta(r_1, r)$ is the sum of m random variables that lie in the interval $[-2, 2]$. This is because, if $\delta(r_1, r_2) = -1$ then $\delta(r_1, r) \neq 2$ and vice-versa. Therefore, we could use the following tail bound, known as Hoeffding bound[9]:

Theorem 1. *Let $X_1, \ldots, X_n$ be n independent random variables, and let a, b be such that for every i, $a \leq X_i \leq b$. Denote $X = X_1 + \ldots + X_n$.*

Then,

$$\Pr(X - E[X] > \alpha) \leq \exp\left(\frac{-2\alpha^2}{n(b-a)^2}\right).$$

Thus, using the Hoeffding bound, we get $\Pr[\delta(r_1, r_2) - \delta(r_1, r) > 0]$

$$= \Pr[\delta(r_1, r_2) - \delta(r_1, r) + 2d(i, i')^2 > 2d(i, i')^2] \leq \exp\left(-\frac{d(i, i')^4}{2m}\right)$$

If $d(i, i') = (6tm \log n)^{0.25}$ for $t > 1$, we get by the union bound that with very high probability all triplets satisfy the property that edge distance within a sub-population is lesser than any edge distance across two populations. The probability that this event does not happen is smaller than $\frac{1}{n^{3t-2}}$. We now use these observations to compute the expected triplet distances. Assume that r_1, r_2 are from sub-population i, and that P_i is the frequency (prior) of this sub-population in the entire population. Then,

$$\begin{aligned} E[d_{a,b}(r_1, r_2)] &= E[a \sum_r X_r + b \sum_r Y_r] \\ &\leq n\left(a\frac{P_i}{3} + b(1 - \frac{2P_i}{3})\right) + \frac{|a| + |b|}{n^{3t-2}} \\ &\approx n\left(a\frac{P_i}{3} + b(1 - \frac{2P_i}{3})\right) \end{aligned}$$

Similarly, for r_1, r_2 from different sub-populations i, i', it is easy to see that

$$E[d_{a,b}(r_1, r_2)] \geq \frac{an}{2} - \frac{|a|+|b|}{n^{3t-2}} \approx \frac{an}{2}$$

If we know the frequency of the sub-populations in the entire population, then we can take $P = \max_i P_i$. For instance, if we set $a = (2/P) - 2, b = -1$ we get that the expected distance between individuals from two different populations is positive, while the expected distance between individuals of the same population is non-positive. For a balanced cut, selecting $a = 4, b = -1$, gives positive expected distance between individuals of different populations and zero otherwise. In practice, even though we do not know the correct value of P, we try different values of P to determine a, b, each giving different partitions. We then pick the partition with the largest likelihood score, where the likelihood score is similar to the one used for `STRUCTURE`, as we now describe.

Recall that A is the input genotype matrix with $R(A)$ being the genotypes of the n input individuals, $\theta : R(A) \mapsto \{1, \ldots, K\}$ is the classification of individuals to sub-populations and $\boldsymbol{p^i}$ is an m-dimensional vector of the MAF of sub-population i. Given θ, the maximum likelihood estimate of $\boldsymbol{p_i}$ is obtained by simply counting the allele frequencies in each of the sub-populations defined by the partition. The posterior probability is given by

$$\Pr[\theta, \boldsymbol{p^i}|A] \propto \Pr[\theta] \Pr[\boldsymbol{p^i}] \Pr[A|\theta, \boldsymbol{p^i}]$$

We set the priors for θ and $\boldsymbol{p^i}$ to be fixed and uniform, and thus maximizing the posterior is equivalent to maximizing the likelihood $\mathcal{L}(A|\theta, \boldsymbol{p^i}) = \Pr[A|\theta, \boldsymbol{p^i}]$.

Algorithm (`GRAPH-TRIPLETS`*).* We can now describe the whole algorithm. The algorithm begins by computing the MF-distance for each pair of individuals. Then, for every pair of individuals r_1, r_2, we compute $X(r_1, r_2) = \sum_r X_r$, and $Y(r_1, r_2) = \sum_r Y_r$. The algorithm then proceeds in iterations. In each iteration we pick a value for P, and we search for a partition that maximizes the likelihood score, based on the prior information that one of the sub-populations is of size P. We take values of P ranging from 0.5 through 0.9 in 0.1 increments. Each such value determines the values of a and b. The triplet distances are then computed for each pair of individuals, by setting $d_{a,b}(r_1, r_2) = ((2/P)-2)X(r_1, r_2)-Y(r_1, r_2)$. These distances induce a complete graph $G = (V, E)$, where the vertices represent individuals and the edges are weighted by the triplet distances. We are then interested in finding the maximum K-cut.

Unfortunately, finding the max-K-cut of the graph is an NP-hard problem even when $K = 2$ [7]. We therefore use the Kernighan-Lin heuristic [9], which is a hill-climbing method to find the optimal cut. The algorithm for the case when $K = 2$ is presented in Figure 1. The algorithm randomly partitions the vertices $V(G)$ into two disjoint sets V_1 and V_2. The algorithm then proceeds in *rounds* each of which involves performing $|V(G)|$ *iterations*. At each iteration we move a vertex u from one side of the cut to the other. The vertex u is chosen so that the resulting cut is maximized. Unlike standard local search techniques, the algorithm swaps u even if this results in the reduction of the cut-size. Once a vertex u is swapped, it cannot be swapped again until the next round. At the end of a round all vertices have been swapped, and the best partition in that round is chosen for the next round. We repeat until the cuts in the beginning and the end of a round are identical, and thus no improvement can be achieved. In Figure 1, set $V_x(V_{x'})$ denotes the vertex set of V_1, V_2 that currently contains x (does not contain x) and $\text{cut}(V_1, V_2)$ denotes the cost of the cut.

Using Kernighan-Lin, for each setting of a, b we find a max-K-cut and we select the one that maximizes $\mathcal{L}(A|\theta, \boldsymbol{p^i})$. Finally, we perform a greedy local search by moving a vertex from one side to another, if it improves the likelihood. This final step, typically improves the accuracy by a little in practice.

3 Results

To evaluate the performance of our method, we compared
`GRAPH-TRIPLETS` to two state of the art methods that deal with population stratification, namely `STRUCTURE` and `EIGENSTRAT`, which are described below.

`STRUCTURE` [13] is a well established package that uses Markov Chain Monte Carlo method (MCMC) to heuristically maximize the posterior probability. Structure can be seeded with a number of different parameters. We used $K = 2$, to denote that the program should look for two sub-populations. By default, the program assumes that the allele frequencies of the two populations are independent. For closely related

kernighanLin(graph G)

1. randomly partition $V(G)$ into V_1, V_2
2. $\alpha \leftarrow 1$
3. `while` $\alpha = 1$ `do` `(* rounds *)`
   (a) $\alpha \leftarrow 0, \chi \leftarrow V_1 \cup V_2$
   (b) `while` $|\chi| > 0$ `do` `(* iterations *)`
      i. $u \leftarrow \text{argmax}_{x \in \chi} \text{cut}(V_x \setminus \{x\}, V_{x'} \cup \{x\})$
      ii. $\chi \leftarrow \chi \setminus \{u\}$
      iii. `if` $u \in V_1$ `then` $V_1 \leftarrow V_1 \setminus \{u\}, V_2 \leftarrow V_2 \cup \{u\}$
      iv. `else` $V_1 \leftarrow V_1 \cup \{u\}, V_2 \leftarrow V_2 \setminus \{u\}$
      v. `if` $\text{cut}(V_1, V_2) > \text{cut}(V_1^*, V_2^*)$ `then` $V_1^* \leftarrow V_1, V_2^* \leftarrow V_2$, $\alpha \leftarrow 1$
   (c) $V_1 \leftarrow V_1^*, V_2 \leftarrow V_2^*$

Fig. 1. Kernighan-Lin heuristic to find max-cut of graph G.

sub-populations, however, the software allows for a mode in which the frequencies are assumed to be correlated. We ran the program on both modes, with the parameter turned off and on. The default number of *MCMC* and *Burnin* iterations is 2000 each. We varied this number to analyze the trade-off between run-time and accuracy. We used the default values for the rest of the parameters.
We note that `STRUCTURE` is a software that does much more than just clustering individuals. Among other things, it can cope with admixed populations, and it can incorporate linkage disequilibrium into its model. We have not compared our method to these modes of `STRUCTURE`, as it is beyond the scope of this paper, and our algorithm is not optimized for such tasks at this point.
`EIGENSTRAT` [14] is a relatively new software tool, which corrects population sub-structure by the spectral properties of the covariance genotype matrix. In a nutshell, `EIGENSTRAT` takes A an $m \times n$ input genotype matrix, where rows are SNPs and columns are individuals and normalizes each entry of A by subtracting the row mean (minor allele frequency of the SNP) and dividing by the row's standard deviation. It then takes the largest eigenvectors of the covariance $n \times n$ matrix Ψ, and uses those to correct for population sub-structure. Even though `EIGENSTRAT` is not explicitly described as a genetic clustering method, we adapt their algorithm in a natural way, resulting in a clustering algorithm in which the clusters are determined by using the sign of the entries of the highest eigenvector of Ψ. We implemented this clustering algorithm in MatLab and compared it to our method. We refer to our implementation as `Spectral` in the results presented.

Datasets. For the evaluation, we used datasets from two different sources. First, we used simulated data generated using the following model. Each sub-population i is represented by an m-dimensional vector of allele frequencies $\boldsymbol{p^i}$, and an individual of the population is sampled by randomly and independently picking allele counts according to the

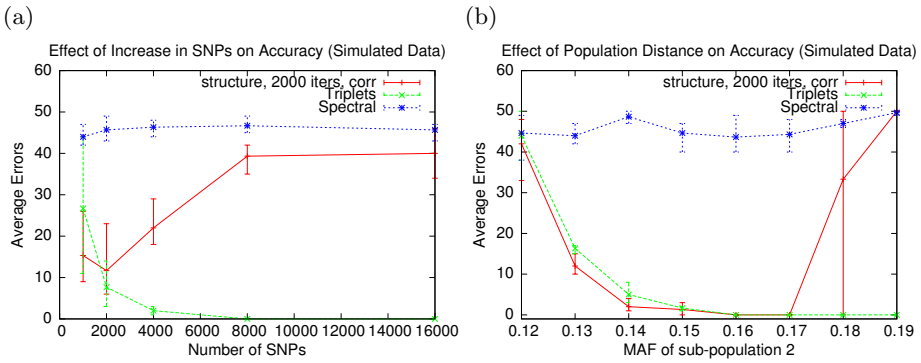

Fig. 2. Comparison of accuracy on simulated data. `GRAPH-TRIPLETS` is consistent in its accuracy and converges to the correct partition with increase in SNPs or increase in distance. On (a) MAF of sub-populations 1 and 2 were 0.1 and 0.13 respectively. On (b) the number of SNPs was fixed to 1000 and MAF of sub-population 1 was fixed at 0.1. We used an average of three randomly generated data sets to obtain every point. Error bars indicate highest and lowest values obtained.

allele frequency distributions of the sub-population. For simulations, we assumed that all SNPs within a sub-population had the same allele frequency, i.e., for any $i, p_j^i = p_{j'}^i$.
We have also used the publicly available data from the International HapMap consortium [10]. This data-set consists of four population groups: Utah residents with ancestry from northern and western Europe(CEU), Yoruba in Ibadan, Nigeria (YRI), Han Chinese in Beijing, China (CHB) and Japanese in Tokyo, Japan (JPT) with 90, 90, 45 and 44 individuals respectively. The Central Europeans and Yoruba Africans consisted of thirty trios each, and therefore in order to avoid these dependencies, we used the 60 parents from each of the two populations, ignoring the 30 children. To obtain a test set where the SNPs are independent, we sampled m SNPs uniformly at random from chromosome 10. We evaluated the programs on each of the six pairs of populations, with different numbers of SNPs, ranging from 1000 to 8000.

Evaluation Measures. There are many possible ways to evaluate the performance of the algorithms. We chose to let each of the program separate the genotypes of two populations (say Africans and Chinese in the HapMap data) into two clusters, and the error rate of such an experiment would be the number of individuals misclassified (for instance, the number of Africans classified as Chinese). We have also compared the running-time of the methods. In summary, our experiments show that the graph-based method is significantly faster while being at least as accurate as existing methods.

Simulated Data. On simulated data, we studied closely related populations. We fixed the minor allele frequency of one population to be 0.1 for each of its SNPs, while for the other population the minor allele frequency

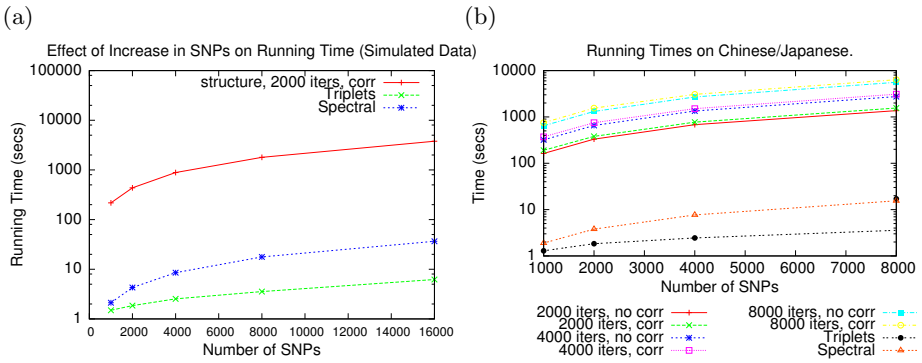

Fig. 3. Comparison of run-times on simulated and real data. `GRAPH-TRIPLETS` is hundreds of times faster than `STRUCTURE`.

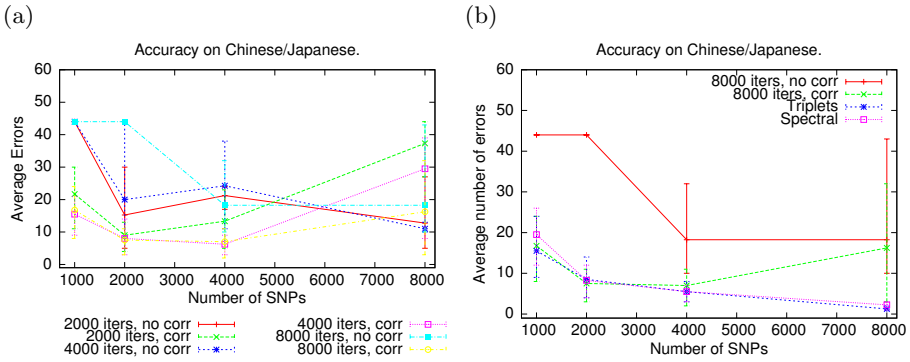

Fig. 4. Comparison of accuracy on HapMap Chinese/Japanese. `GRAPH-TRIPLETS` is accurate and converges to the correct partition with increase in SNPs. It is unclear what parameters of `STRUCTURE` to use. We used the average of four randomly drawn data sets to obtain every point. Error bars indicate the highest and lowest values obtained.

varied from 0.12 to 0.19. On all the data presented, `STRUCTURE` running on default parameters fails to find two sub-populations, i.e., it returns solution with all individuals in one cluster. We therefore only report `STRUCTURE` running for 2000 iterations with the correlation mode turned on. Figure 3(a) shows that `GRAPH-TRIPLETS` is an order of magnitude faster than `STRUCTURE`. Figure 2 shows that it makes substantially less errors overall than `STRUCTURE` and the spectral clustering which is based on `EIGENSTRAT`. It is conceivable that `STRUCTURE` would find better solutions if it uses even more time, although this seems rather prohibitive. We note that the performance of `STRUCTURE` seems not to be monotonic with the number of SNPs used.

HapMap. For the HapMap datasets, we considered all six pairs of populations. For most pairs, all methods worked perfectly (no errors were made) with as few as 200 SNPs. The only hard instance was the

Chinese-Japanese pair. For this pair, none of the methods could give a perfect clustering prediction even when 8000 SNPs were used. As can be seen from Figure 4, the spectral method and `GRAPH-TRIPLETS` seem to produce lesser errors than the classification returned by `STRUCTURE`. Furthermore, we used the two different modes of `STRUCTURE` (with or without correlations), and the results were inconclusive regarding which one works better on this dataset, as when small number of SNPs were used (1000, 2000 or 4000), the correlation mode seemed to perform better. When more SNPs were used (8000 SNPs, 2000 or 4000 iters), the 'no correlation' mode of `STRUCTURE` was better.
As before, Figure 3(b) demonstrates that `GRAPH-TRIPLETS` is much more efficient than `STRUCTURE`. Specifically, the run time for `GRAPH-TRIPLETS` $\approx$ 300 times faster for 1000 SNPs and $\approx$ 1000 times faster for 8000 SNPs.

4 Significance of Clusters

An obvious and important question to consider is whether the clusters obtained from the methods are significant. In practice, we could run `STRUCTURE`or `GRAPH-TRIPLETS` with K, the number of sub-populations set to 2. When the software returns a solution, with individuals divided into two populations, there is no guarantee on whether the input set of taxa actually even contains two sub-populations. To test the significance of the clusters, one could perform the statistical tests described by Pritchard et al. [13], which as the authors themselves point out either uses dubious assumptions or does not work for large number of SNPs in practice. Alternatively, we could examine the change in the likelihood function or use information based evaluation such as minimum description length to decide on whether there is truly two sub-populations or not.
A simple and direct approach is to permute the alleles in each site independently such that the input no longer has sub-structure. We can then re-run the algorithm on the new input. The p-value is simply the fraction of times, the permuted input had solution larger (or more likely) than that of the original input.
However, such tests can only be performed if the algorithm to find clusters is very efficient. Here, we simply considered a randomly drawn data set with 8000 SNPs from each of the HapMap populations. We ran structure (default parameters) and triplets with $K = 2$. We report the number of errors made (size of the smaller set) along with the p-value which can be efficiently computed using the triplets method. We believe that the computation of this p-value is a great benefit in practice that our new technique can offer. The results for 1000 permutations is presented in Table 1.

5 Conclusions

The problem of population stratification is an increasing concern in the context of disease association studies. In particular, its influence on whole genome association studies (for e.g. [8, 15]) are severe. Even though existing methods for clustering individuals based on their SNPs provide

	Errors		
	`structure`	Triplets	Triplets p-value
Central Europeans, CEU	21	2	0.01
Yoruba Africans, YRI	26	20	0.75
Chinese, CHB	17	16	0.267
Japanese, JPT	18	12	0.929

Table 1. Triplets can be used to compute p-values directly.

relatively accurate predictions, there is no rigorous theory that ensures the convergence of these methods to the correct solution. Furthermore, there is no study that compares these methods on a variety of datasets, both real and simulated. Our paper has been motivated by these two concerns.

In this paper, we suggest a graph based method for detecting population stratification. The distance measure used in our method builds on the Mother-Father distance that was suggested in [3], where it has been rigorously and analytically shown that if the sample size is large enough, the measure will represent the correct distance between individuals, and therefore our algorithms will converge to the true population clusters. We believe that this theoretical foundation for our algorithm is an important advantage that proves itself in practice. In particular, we show that our algorithm is consistently at least as accurate as `STRUCTURE` and `EIGENSTRAT`.

One of the questions we raised in this paper is the validity of the population substructure found by the clustering algorithm. We have demonstrated that the different methods will tend to cluster the individuals in two clusters, even if in reality there is only one population. In order to assess the significance of these partitions, we suggest a permutation test, which seems to give the correct significance scores on the HapMap populations. This permutation test can only be carried out if the clustering methods are highly efficient. As our method runs in seconds over thousands of SNPs and hundreds of individuals, this was feasible.

We note that our paper focuses on the clustering of two populations while `STRUCTURE` and `EIGENSTRAT` can do much more than that. In particular, they can deal with admixed populations, correlated populations, and linkage disequilibrium. We hope that a combination of the existing methods such as `STRUCTURE` and `EIGENSTRAT`, together with our graph based approach may lead to improved tools for these cases as well.

6 Acknowledgments

EH and SS were supported by NSF grant IIS-0513599. SS was also partially supported by NSF grant IIS-0612099.

References

1. R. Boppana. Eigenvalues and Graph Bisection: An Average Case Analysis. In proc *IEEE Symposium on Foundations of Computer Science*, 1987.
2. W. Buntine and A. Jakulin. Applying Discrete PCA in Data Analysis. In proc *Uncertainty in AI*, 2004.
3. K. Chaudhuri, E. Halperin, S. Rao and S. Zhou. Separating Populations with Small Data. To appear in proc *Symposium on Discrete Algorithms (SODA)* 2007.
4. D. F. Conrad, T. D. Andrews, N. P. Carter, M. E. Hurles and J. K. Pritchard. A high-resolution survey of deletion polymorphism in the human genome. *Nature Genetics*, 38, 2006.
5. M. J. Daly, J. D. Rioux, S. F. Schaffner, and T.J. Hudson. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29, 2001.
6. S. Gabriel, S. Schaffner, H. Nguyen, J. Moore, J. Roy, B. Blumenstiel, J. Higgens, M. DeFelice, A. Lochner, M. Faggart, S.N. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, E.S. Lander, M.J. Daly, and D. Altschuler. The Structure of Haplotype Blocks in the Human Genome. *Science*, 296, 2002.
7. M. R. Gary and D. S. Johnson. Computers and Intractability. *Freeman*, 1979.
8. J. N. Hirschhorn and M. J. Daly. Genome-wide Association Studies for Common Diseases and Complex Traits. *Nature Review Genetics*, 6, 2005.
9. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association(JSTOR)*, 58, 1963.
10. The International HapMap Consortium. A Haplotype Map of the Human Genome. *Nature*, 437, 2005.
11. B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 1970.
12. F. McSherry. Spectral Partitioning of Random Graphs. *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2001.
13. J. K. Pritchard, Matthew Stephens and Peter Donnelly. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155, 2000.
14. A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38, 2006.
15. D. C. Thomas et al. Recent Developments in Genome wide Association Scans: a Workshop Summary and Review. *American Journal of Human Genetics*, 77, 2005.

(a) (b)

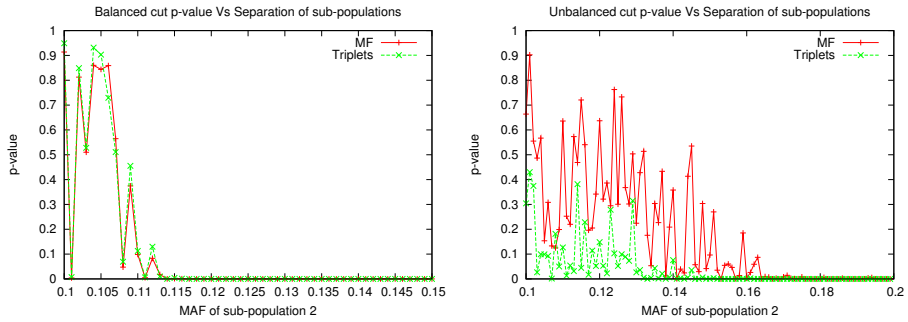

Fig. 5. Comparison of MF and Triplet distance measures. Figure (a) shows that when only balanced cuts are considered, the distance measures provide comparable p-values. Figure (b) shows that more often unbalanced cuts in the MF distance contains cost larger than that of the correct cut. In both figures MAF of sub-population 1 was fixed to 0.1. For Figure (a) we used 100 individuals in each population and for Figure (b) we used 10 individuals in each population. We used parameters $a = 2, b = -1$ for the triplets.

A Appendix: Empirical Comparison of MF with Triplets.

To motivate our choice of triplet based distance instead of MF-distance, we show empirically that the triplet distance gives a better separation of the two populations into two clusters. In order to do so, we randomly generate two sub-populations according to the following model. We generate the two populations, such that one population has minor allele p_1 for all the SNPs, and the other population has minor allele p_2 for all the SNPs. Furthermore, each of these sub-populations has the same number of individuals. We measure the cut distance of the correct cut using the MF-distance and the triplet distance. Let d_{mf} and d_t be the correct cut weight for mother-father and triplets. We then find $10,000$ random balanced cuts of the graph and measure the cut weights. We then measure the fraction of times the random cut cost was larger than d_{mf} or d_t respectively. We call this measure the *balanced p-value*. We also measured the max-cut for $10,000$ randomly generated unbalanced cuts, where 0.25-fraction of the vertices were on one side. We call this measure the *unbalanced p-value*. .

The results of the various simulation tests are shown in Figure 5. Figure 5(a), shows that the balanced p-values are similar for triplets based distance and for the MF-distance, and that the balanced p-value of both methods quickly go down to zero. More importantly, Figure 5(b) shows that triplets clearly has a lower p-value in the case of unbalanced cuts. While MF-distance contains several unbalanced cuts of high weight, on the triplets, the correct cut has cost typically higher than all other cuts. This provides an evidence that the triplet distance is advantageous.