BIOINFORMATICS

Vol. 29 ISMB/ECCB 2013, pages i162–i170
doi:10.1093/bioinformatics/btt237

IBD-Groupon: an efficient method for detecting group-wise identity-by-descent regions simultaneously in multiple individuals based on pairwise IBD relationships

Dan He

Computational Genomics, IBM T.J. Watson Research, Yorktown Heights, NY 10598, USA

ABSTRACT

Motivation: Detecting IBD tracts is an important problem in genetics. Most of the existing methods focus on detecting pairwise IBD tracts, which have relatively low power to detect short IBD tracts. Methods to detect IBD tracts among multiple individuals simultaneously, or group-wise IBD tracts, have better performance for short IBD tracts detection. Group-wise IBD tracts can be applied to a wide range of applications, such as disease mapping, pedigree reconstruction and so forth. The existing group-wise IBD tract detection method is computationally inefficient and is only able to handle small datasets, such as 20, 30 individuals with hundreds of SNPs. It also requires a previous specification of the number of IBD groups, or partitions of the individuals where all the individuals in the same partition are IBD with each other, which may not be realistic in many cases. The method can only handle a small number of IBD groups, such as two or three, because of scalability issues. What is more, it does not take LD (linkage disequilibrium) into consideration.

Results: In this work, we developed an efficient method *IBD-Groupon*, which detects group-wise IBD tracts based on pairwise IBD relationships, and it is able to address all the drawbacks aforementioned. To our knowledge, our method is the first practical group-wise IBD tracts detection method that is scalable to very large datasets, for example, hundreds of individuals with thousands of SNPs, and in the meanwhile, it is powerful to detect short IBD tracts. Our method does not need to specify the number of IBD groups, which will be detected automatically. And our method takes LD into consideration, as it is based on pairwise IBD tracts where LD can be easily incorporated.

Contact: dhe@us.ibm.com

1 INTRODUCTION

Two or more alleles are identical by descent (IBD) if they are identical copies of the same ancestral allele. IBD tracts are regions shared between two or more individuals if they inherit identical nucleotide sequences in the regions from common ancestor. IBD detection has been applied to a wide range of applications, such as detecting relatedness of unknown individuals (Albrechtsen *et al*., 2009; Browning and Browning, 2011; Gusev *et al*., 2009; He *et al*., 2013; Purcell *et al*., 2007), reconstructing pedigrees (He *et al*., 2013), phasing haplotypes (Browning and Browning, 2011), disease mapping (Albrechtsen *et al*., 2009; Purcell *et al*., 2007) and so forth.

Recently, many methods (Albrechtsen *et al*., 2009; Browning and Browning, 2011; Gusev *et al*., 2009; Purcell *et al*., 2007) have been developed to detect IBD tracts between pairs of individuals, based on genotype data or haplotypes. Albrechtsen *et al*. (2009) proposed Relate, which uses a continuous time Markov model where the hidden states are the number of alleles shared IBD between pairs of individuals at a given position. Purcell *et al*. (2007) proposed PLINK, which detects extended chromosomal segmental IBD sharing between pairs of distantly related individuals by use of a hidden Markov model (HMM), in which the underlying hidden IBD state is estimated given the observed IBS (identity-by-state) sharing and genome-wide level of relatedness between the pair. Gusev *et al*. (2009) proposed a program GERMLINE based on a dictionary of haplotypes that is used to efficiently discover short exact matches between individuals. These matches are then expanded using dynamic programming to identify long, nearly identical segmental sharing that is indicative of relatedness. Browning and Browning (2011) implemented an algorithm fastIBD in the program BEAGLE, which is based on estimating frequencies of shared haplotypes because a shared common haplotype is unlikely to reflect recent IBD, whereas a shared haplotype that is rare is likely to be identical by descent. A fixed number of haplotype pairs are sampled for each individual from the posterior haplotype distribution. Each sampled haplotype corresponds to a sequence of HMM states. The fastIBD algorithm searches for pairs of sampled haplotypes sharing the same sequence of HMM states for a set of consecutive markers. Overlapping shared haplotype tracts are merged, and the merged shared haplotype tract is a mosaic of pairs of sampled haplotypes. A fastIBD score is calculated for each merged tract, and if the score is below a user-specified threshold, the tract is reported as IBD.

These methods have been shown to be powerful to detect long pairwise IBD tracts. However, they generally suffer from low power to detect relatively short IBD tracts, for example, IBD tracts of length <1 MB. And they all aim to detect pairwise IBDs only. Methods to use group-wise IBD relationships or IBD tracts shared by multiple individuals to increase the power for short IBD tracts detection have been proposed. Hansen *et al*. (2009), Leibon *et al*. (2008) and Thomas *et al*. (2007) have proposed methods to detect group-wise IBD tracts for cases in which some pedigree information or information about IBD sharing is available. Moltke *et al*. (2011) proposed a more general Markov Chain Monte Carlo (MCMC) method where it does not assume any previous information about IBD sharing patterns. Group-wise IBDs are detected directly from unphased genotype data, using an HMM model where the states are possible IBD tracts among the given set of individuals. An MCMC approach is developed to infer relevant information about the parameters of the HMM model. Besides the capability to report group-wise IBD tracts, it is also shown that the MCMC approach has relatively

high power to detect short IBD tracts compared with the pairwise IBD tracts detection methods.

The MCMC method, however, suffers from intensive computation; therefore, it can be only applied to small datasets, for example, 20–30 individuals with 500 SNPs. And the MCMC method takes tens of hours even on these small datasets. It also requires a previous specification of the number of *IBD groups*, or partitions of the individuals where all the individuals in the same partition are IBD with each other. The number of IBD groups needs to be small, such as two or three. This is unrealistic in many cases. What is more, the MCMC method does not take linkage disequilibrium (LD) into account, which is common in the genotype data.

To address all the drawbacks of the MCMC method, we developed a method *IBD-Groupon,* which detects group-wise IBD tracts based on pairwise IBD tracts. We first use Beagle (Browning and Browning, 2011) to detect pairwise IBD tracts, as Beagle allows us to adjust a threshold to detect short IBD tracts, with the side effects of obtaining many false-positive IBD tracts. Then we construct an HMM to select the most likely group-wise IBD tracts to eliminate false-positive IBD tracts, based on the assumption that the length of a false-positive IBD tract is generally smaller than that of the true positives, and thus false positives generally have lower likelihoods. We also developed a depth first search algorithm, as well as a shortest path algorithm to effectively select the possible values of the states of the HMM such that our method is scalable to large datasets. IBD-Groupon does not require any specification of the number of IBD groups and the length of the IBD tracts, which will be detected automatically. Using both simulated and real data, we show that our method is not only much more efficient than the MCMC method but also much more accurate, in that it has much lower false-positive rate. And to our knowledge, it is the first practical method that is able to detect group-wise IBD tracts on large datasets, for example, hundreds of individuals with thousands of SNPs, and in the meanwhile, it does not assume any previous information of the IBD sharing patterns. It also takes LD into consideration, as Beagle considers LD for pairwise IBD detection.

Gusev *et al.* (2011) proposed a method DASH that identifies pairwise IBD tracts using GERMLINE first and then conducts an iterative minimum cut algorithm to identify densely connected components as IBD clusters. The method cuts the haplotypes into windows, and the min-cut algorithm runs recursively until it identifies a subgraph of desired density or a trivial subgraph that contains no edges to be cut. The idea is similar to IBD-Groupon but with the following significant differences: (i) DASH takes the density threshold of the subgraphs, as well as the length of the windows as the parameters of input. IBD-Groupon uses an HMM to determine the cliques and the length of the identified IBD tracts automatically. (ii) DASH identifies the IBD clusters in each window independently. IBD-Groupon builds a global probabilistic framework to combine information cross-windows. Indeed, DASH can be considered as a special case of IBD-Groupon where there is one HMM state for each window but with one value for each state. We show in our experiments that compared with DASH, not only IBD-Groupon requires no parameter tuning but also it produces more accurate results. And the two methods have similar execution time.

2 METHODS

Our method IBD-Groupon builds an HMM based on the pairwise IBD tracts. The HMM creates a state for each loci or SNP (single-nucleotide polymorphism) position. For SNPs across all haplotypes at each position, we first obtain all pairwise IBD relationships of them. As pairwise IBD relationship should be transitive, namely, if allele a and b are IBD, b and c are IBD, then a and c should be IBD, we say there is a *conflict* if a and c are not IBD. Then we generate all possible *global IBD configurations* from these pairwise IBD relationships, which are sets of IBD groups that determine the pairwise IBD relationships for every pair of haplotypes at the loci, and there should be no conflict among these pairwise IBD relationships. There can be multiple global IBD configurations for each loci. For the above example, we can have two configurations: $[[a, b], c]$ and $[[b, c], a]$. In the first configuration, a, b are IBD, and they are not IBD with c. Thus, there is no conflict. Similarly, in the second configuration, b, c are IBD, and they are not IBD with a.

Each global IBD configuration is then considered as a value of the HMM state. The emission probability is computed as the posterior probability of the configuration given all the pairwise IBD relationships for this position. The transition probability is computed according to the change of the IBD status between adjacent SNPs on the same haplotypes. Based on the HMM, we identify the most likely global IBD configurations for all SNP positions, and we concatenate them to obtain group-wise IBD tracts among multiple individuals of the whole genome. To make the HMM practical for large datasets, we developed various of techniques and the details will be given in the next few sections.

We show the flow chart of IBD-Groupon in Figure 1. Given the unphased genotype data, we first phase them using the existing tools, such as Beagle Browning and Browning (2011), and we obtain the pairwise IBD tracts among all the haplotypes where both inbreeding and LD are considered. We define a *haplotype chunk* as a subsequence of a haplotype. Given the pairwise IBD relationships of all the individual haplotypes, we first split the haplotypes into the *maximum IBD chunks,* which are defined as the maximum haplotype chunks where the IBD status between all pairs of haplotypes remains the same. We create one HMM state for each chunk. Next, for each chunk, we construct an *IBD Graph* where the nodes are the chunks from each haplotype, the edges connect the haplotype chunks that are IBD to each other. Once we have such a graph, we search for all maximal cliques and build a bipartite graph where one side is all the haplotype chunks, the other side is all the maximal cliques. An edge indicates the maximal clique *covers* a haplotype chunk (namely, the clique in the IBD graph contains the corresponding haplotype node), and a clique may cover multiple haplotype chunks. As these maximal cliques give configurations of the group-wise IBD status among the individuals, selecting different cliques generally leads to different global configurations, and thus naturally different values of the HMM state (we will give the definition of 'select' later). As the number of possible configurations, we rank the configurations according to the posterior probability of the cliques given the pairwise IBD relationships and select the top- k configurations, where k is small enough for the HMM. Therefore, IBD-Groupon is an approximation algorithm, but we show in our experiments using k as 100 makes significant improvements over the state-of-the art methods. Ranking the configurations requires computing the posterior probabilities of all configurations, which can be expensive and even prohibitive. We convert the problem into a top- k shortest paths problem on a *State Graph* where each shortest path represents one configuration, and the length of shortest paths is proportional to the reverse of the posterior probability. By extending algorithms, such as Dijkstra algorithm (Dijkstra, 1959), we can select the top- k shortest paths directly without exploring all possible paths. And then we take the top- k best

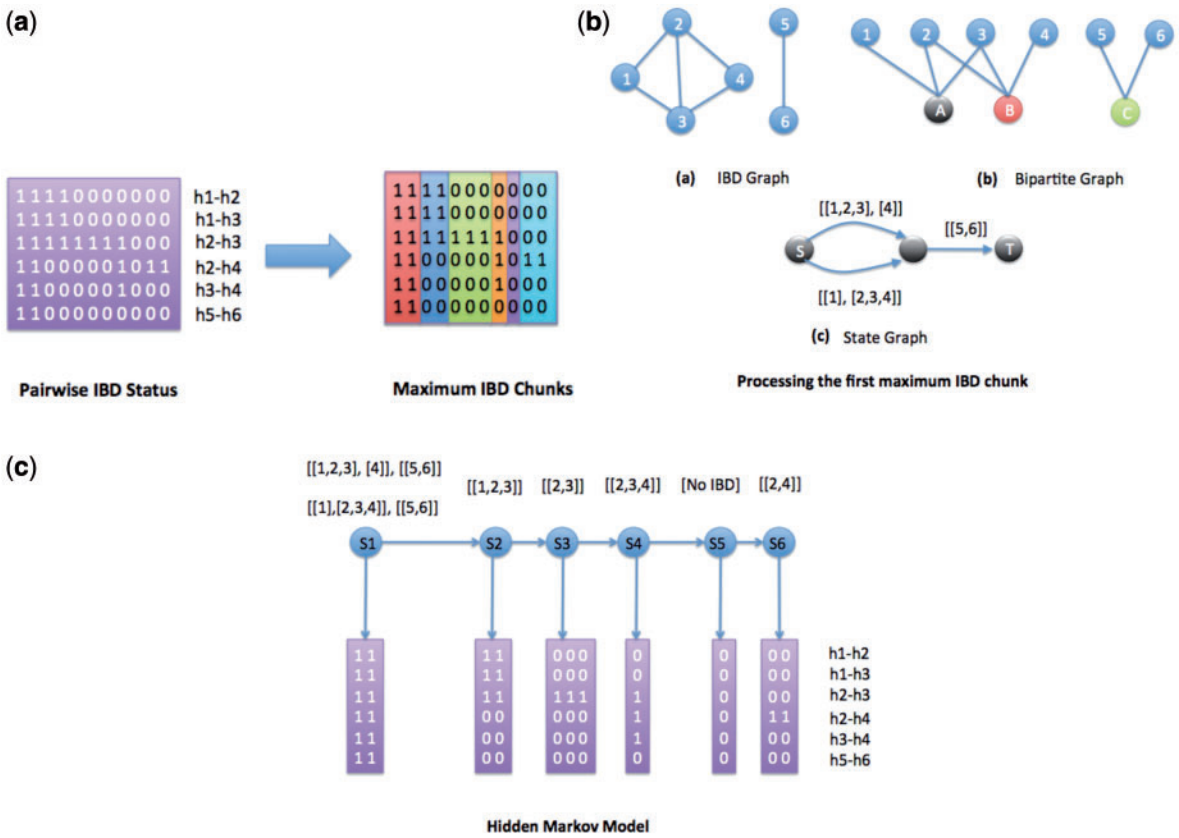

Fig. 1. Flow chart of IBD-Groupon. (**a**) Identify the maximum IBD chunks from pairwise IBD relationships. (**b**) Running example of IBD-Groupon for the first maximum IBD chunk, where haplotypes 1, 2 and 3 are IBD, haplotypes 2, 3 and 4 are IBD and haplotypes 5 and 6 are IBD. (b.a) The IBD graph. (b.b) The bipartite graph where node *A* corresponds to the maximal clique [1,2,3], node *B* corresponds to the maximal clique [2,3,4], node *C* corresponds to the maximal clique [5,6]. (b.c) The state graph. (**c**) The full HMM built from the pairwise IBD relationships. As there are six chunks, there are six states. Only the first state has two values. Each state emits the pairwise IBD relationships for the corresponding chunk

configurations as the values of the HMM state. Next, we explain each step of IBD-Groupon in details.

2.1 Identify maximums IBD chunks

We define a *haplotype chunk* as a subsequence of a haplotype. Given the pairwise IBD relationships of all the individual haplotypes, we first split the haplotypes into the *maximum IBD chunks,* which are defined as the maximum haplotype chunks where the IBD status between all pairs of haplotypes remains the same. As shown in Figure 1a, if we mark positions in the pairwise IBD tracts as 1 and in the non-IBD tracts as 0 (in the example, there are six pairwise IBD relationships, each row is one relationship), we can identify the maximum IBD chunks where there is no switch between 1 and 0 in the chunk. As the IBD status remains the same in the chunk, we do not need to create separate HMM states for each SNP position in the same chunk. Instead, we create one HMM state for each chunk, which significantly reduces the complexity of the HMM.

2.2 IBD graph

Once we have identified the maximum IBD chunks, we create an *IBD Graph* for each chunk, where one node is created for each haplotype rather than one individual, as we would like to handle inbreeding cases where the two haplotypes of the same individual may share certain IBD regions. And we create an edge between a pair of haplotypes if they are IBD for this chunk. An example of an IBD graph is shown in Figure 1b. We can see haplotypes 1, 2 and 3 are IBD for the first chunk, haplotypes 2, 3 and 4 are IBD for the same chunk and haplotypes 5 and 6 are IBD for the same chunk.

Once we build the IBD graph, we apply a maximal clique algorithm (Bron and Kerbosch, 1973) to find all the maximal cliques, which has the worst case complexity $O(3^{h/3})$, where h is the total number of haplotypes. The complexity in reality is much better because group-wise IBD is rare, and the size of groups is usually not big. Each clique indicates one configuration of IBD status for the haplotypes involved. For example, a maximal clique $[h_1, h_2, h_3]$ indicates the three haplotypes are IBD for this chunk. And we consider maximal cliques to identify group-wise IBD tracts among the most number of haplotypes. Therefore, we can obtain different IBD configurations by selecting different cliques (from now on, we will simply use clique to represent maximal clique).

We can obtain a global IBD configuration by *selecting* the maximal cliques. Once we select a maximal clique, all the haplotypes in the clique are IBD with each other. The global IBD configuration then determines the IBD relationship of the haplotypes according to the selected cliques so that all the haplotypes in the same clique are IBD with each other. If all the maximal cliques are independent, namely, they do not overlap, we select all of them and we will have only one global IBD configuration for all haplotypes at this chunk. However, in reality, these cliques may overlap, indicating that the detected pairwise IBD relationships may conflict with each other. As the IBD configuration cannot have any conflicts, once we select a clique, we delete its corresponding nodes and edges in the graph. The remaining cliques will be shrunk by this deletion, as the overlapping haplotypes are deleted, and the conflicts are thus resolved.

Therefore, the order of selection determines the IBD configuration. In the example in Figure 1c in the IBD graph, there are two overlapped maximal cliques C_1 [1,2,3] and C_2 [2,3,4]. We have two way, or two orders to select them, namely, C_1C_2 or C_2C_1. Thus, there are two possible global configurations [[1, 2, 3], 4] (once we have selected C_1, C_2 shrank to [4]) or [1, [2, 3, 4]] (once we have selected C_2, C_1 shrank to [1]). Therefore, given a set of n maximal cliques, we have O(n!) ways to select them, which generates O(n!) orders, each leading to a global configuration.

As we can see, the number of possible global IBD configurations, or values for the HMM state, can be very large, and the HMM may not be scalable to such large number of values. Therefore, we would like to select the top- k 'best' configurations where k is a reasonably large number, such as 100, where the HMM is still scalable, and consider these configurations only. We determine the quality of the configurations by the posterior probability of each configuration given the pairwise IBD relationships, or the emission probability of the corresponding HMM state. Given a global IBD configuration and the pairwise IBD relationships for all individual haplotypes, each associated with a probability, the emission probability for the global IBD configuration can be computed as follows:

$$emission(GC) = \prod_{[h_i, h_j] \in GC} P(h_i, h_j) \quad (1)$$

where GC denotes global IBD configuration, $[h_i, h_j]$ is a pair of haplotypes that are IBD in the global configuration, and $P(h_i, h_j)$ is the probability of the pairwise IBD relationships between the two haplotypes. Notice $P(h_i, h_j)$ can never be 0, as h_i, h_j can be IBD in the global configuration only when $P(h_i, h_j)$ is greater than a threshold.

In this work, we use Beagle (Browning and Browning, 2011) to detect pairwise IBD relationships, and Beagle reports a value for each pairwise IBD relationship. However, the value approximates the frequency of the corresponding IBD tracts in the population, which can be interpreted as the probability of IBS for the region or the probability that the regions are identical by random chance. Thus, the smaller the value is, the more likely the IBD tract is true, as the probability of IBS is small. Therefore, this value is not the probability of IBD, and we need to convert it into an IBD probability. As the smaller the value is, the more likely the IBD tract is true, we compute the inverse of the value, and we normalize the new value by the maximal inversed value from all the IBD tracts for the chunk so that all the inversed values are within the range of [0, 1]. The normalized inversed values are then used as the probability of IBD tracts. The reason that we do normalization by the maximal inversed value is because Beagle is powerful in detecting IBD tracts of length >1 MB. The IBD tracts with the maximal inversed value are of length generally >1 MB. For IBD of such length reported by Beagle, we simply consider the IBD probability is 1.

To obtain the top- k configurations, a naïve way is to compute the posterior probability of each configuration and then rank them. The complexity is $O(n!log(n!))$, which can be very large, where n is the number of maximal cliques for the chunk. To avoid a full enumeration of all possible configurations, we construct a *state graph* and convert the problem into a top- k shortest path problem on the state graph where full exploration of all paths can be avoided. More details will be given in the next few sections.

2.3 Bipartite graph

Given all the maximal cliques of an IBD graph, we build a bipartite graph where the nodes on one side are the individual haplotypes (we call them *hap-nodes*); the nodes on the other side are the maximal cliques (we call them *clique-nodes*). We create an edge between a hap-node and a clique-node if the clique in the IBD graph contains the haplotype (we say the clique *covers* the haplotype). Thus, both the hap-nodes and clique-nodes may be associated with multiple edges.

When the cliques do not overlap, we simply select all the cliques, and there is a unique global IBD configuration. When the cliques do overlap, we want to identify the overlapped cliques first. This can be achieved by detecting all the connected components on the bipartite graph. The corresponding cliques of the clique-nodes in the same connected components overlap with each other, and we can treat each connected components independently and then combine the configurations from each component.

For each connected component, we still need to enumerate all possible configurations. However, the number is generally much smaller than that for all possible global configurations when all cliques are considered. We further developed an efficient pruning strategy to stop the enumeration as early as possible based on our observation that the maximal cliques in those connected components generally overlap with each other significantly. Once we select a clique, we remove the clique-node and all the hap-nodes it covers. Therefore, after selecting just a few cliques, the bipartite graph for this component is empty, and we do not need to select any of the remaining cliques. Alternatively, the configuration remains the same no matter how we select the remaining cliques. We consider the problem as a search problem on a search tree where the nodes are the cliques and we conducted a depth first search algorithm. A path on the tree determines the order of selection for the clique-nodes, and a subtree is pruned if all the hap-nodes are removed due to the selection of the clique-nodes along the path.

The algorithm is shown in Algorithm 1. On line 2, we rank the cliques first according to the size because we would like to select large cliques first. Selection of large cliques removes more hap-nodes, and thus usually leads to a faster termination of the search. On line 3 and 4, we maintain two records for the search, a set *Available* recording the available, or unselected cliques, a set *cc* recording the current selected cliques, or the current configuration. On line 6, *min*(*Available*) returns the minimum index of the available cliques as the smaller the index is, the larger the clique is and we give higher priority to larger cliques. For every selection, we need to update the two sets *Available* and *cc*, as shown on lines 8 and 9. We also need to remove the hap-nodes covered by the selected cliques as shown on line 7. On line 10, after every selection, we check whether there are still hap-nodes remaining. If yes, we keep on the depth first search. Otherwise, we add the current configuration as one of the unique configurations. Then we do back track and update the two sets *Available* and *cc* accordingly. The whole depth first search ends until no more clique nodes can be selected. The worst case complexity is $O(t!)$ where t is the number of maximal cliques in the connected components. However, as the maximal cliques overlap with each other significantly, in reality, the depth first search terminates fast.

Algorithm 1. Enumerate configurations for a connected component

Require: A bipartite graph G containing k clique-nodes
Ensure: A set of unique configurations C

1: $C \leftarrow \{\}$
2: Rank the clique-nodes of G according to the size of the corresponding cliques as $CN_1, CN_2, \ldots, CN_k$
3: $Available \leftarrow \{1, 2, \ldots, k\}$
4: $cc \leftarrow \{\}$
5: **while** Search does not end **do**
6: Do depth first search and select CN_i where $i \leftarrow min(Available)$
7: Remove the hap-nodes covered by CN_i
8: $Available \leftarrow Available - \{i\}$
9: $cc \leftarrow cc + \{CN_i\}$
10: **if** size(hap-nodes) == 0 **then**
11: $C \leftarrow C + \{cc\}$

12: Back track
13: $Available \leftarrow Available + \{i\}$
14: $cc \leftarrow cc - \{CN_i\}$
15: **end if**
16: **end while**
17: output C

Once we have generated all possible configurations for each connected component, we need to combine them. Assuming we have k connected components with the number of configurations as $n_1, n_2, \ldots, n_k$, respectively. The total number of global configuration is $\prod_{i=1}^{k} n_i$. To compute the posterior probability for all of them and then rank them can be time consuming. To select the top- k best configurations directly, we construct a *State Graph* and convert the problem into a top- k shortest path problem where full exploration of all paths can be avoided.

2.4 State graph

We build a *State Graph,* which is used to generate the final k values for the HMM state of each chunk. Given there are t connected components $cc_1, cc_2, \ldots, cc_t$ with the number of configurations as $n_1, n_2, \ldots, n_t$, respectively. We create $t-1$ nodes $N_1, N_2, \ldots, N_{t-1}$, and also one start node S and one sink node T. Then we create n_1 edges from S to node N_1, where each edge corresponds to one of the n_1 configurations for cc_1. As all the edges are directed, we can only choose one of these edges to go from S to N_1, which corresponds to the fact that for each connected component, each time we can only select one configuration among n_1 configurations. Between node N_i and node N_{i+1}, we create n_{i+1} edges where each edge corresponds to one of the n_{i+1} configurations for cc_{i+1}. Finally, between node N_{t-1} and the sink node T, we create n_t edges where each edge corresponds to one of the n_t configurations for the connected component cc_t. Therefore, the state graph is a DAG (directed-acyclic-graph).

Thus, a global configuration is essentially a path from S to T. If we associate each edge with the emission probability of the corresponding configuration, we can obtain the posterior probability of the global configuration, or the path, by multiplying all the emission probabilities along the path. To convert the problem into a shortest path problem, instead of the emission probability, we associate each edge with a distance, which is the log value of the inverse of the emission probability. Thus, the higher the emission probability, the smaller the distance is. And instead of multiplying the distances, we now sum them and we turn the problem into a top- k shortest path problem, which can be solved efficiently by extending algorithms, such as Dijkstra algorithm (Dijkstra, 1959). We simply run Dijkstra from S to T and terminate the algorithm only after the top- k shortest paths are identified. Full exploration of all paths can be avoided. The worst case complexity is $O(kt(m + t\log(t)))$, where t and m are the number of nodes and edges, respectively, in the state graph. In reality, the state graph contains usually a small number of nodes and edges, as group-wise IBD is rare, leading to a small number of configurations in the connected components.

2.5 Generate HMM states

Once we obtain the top- k global configurations for all the chunks, we construct an HMM model where k values are created for each chunk. Notice for many chunks, the total number of possible values is indeed smaller than k. The emission probability of each chunk can be computed using Equation (1). The transition probability between a value i and a value j between adjacent states is computed by the following formula:

$$T_{i,j} = P_{IBD->IBD}^{N_{IBD->IBD}} \times P_{IBD->NoIBD}^{N_{IBD->NoIBD}} \times P_{NoIBD->IBD}^{N_{NoIBD->IBD}} \times P_{NoIBD->NoIBD}^{N_{NoIBD->NoIBD}}$$

where $P_{X->Y}$ denotes the transition probability from status X to status Y, and X, Y are either status 'IBD' or 'no IBD', $N_{X->Y}$ denotes the number of transitions from status X to status Y between all pairwise IBD relationships of the two adjacent states. As each value of a state is a configuration of all pairwise IBD relationships, including IBD and no IBD, we need to consider all four possible status transitions.

The probability of transition between IBD and no IBD status can be measured from the pairwise IBD relationships as the follows:

$$P_{IBD->IBD} = \sum_{1\le i,j\le m} \frac{N_{IBD->IBD}(h_i,h_j)}{N_{IBD->IBD}(h_i,h_j) + N_{IBD->NoIBD}(h_i,h_j)}$$

$$P_{IBD->NoIBD} = \sum_{1\le i,j\le m} \frac{N_{IBD->NoIBD}(h_i,h_j)}{N_{IBD->IBD}(h_i,h_j) + N_{IBD->NoIBD}(h_i,h_j)}$$

$$P_{NoIBD->IBD} = \sum_{1\le i,j\le m} \frac{N_{NoIBD->IBD}(h_i,h_j)}{N_{NoIBD->IBD}(h_i,h_j) + N_{NoIBD->NoIBD}(h_i,h_j)}$$

$$P_{NoIBD->NoIBD} = \sum_{1\le i,j\le m} \frac{N_{NoIBD->NoIBD}(h_i,h_j)}{N_{NoIBD->IBD}(h_i,h_j) + N_{NoIBD->NoIBD}(h_i,h_j)}$$

where h_i, h_j are the i-th and j-th haplotypes, $N_{X->Y}(h_i,h_j)$ is the number of transitions from status X to status Y between all pairs of adjacent alleles in h_i, h_j, and X, Y are either status 'IBD' or 'no IBD'. We can also consider the weighted version of these probabilities, namely, we weight $N_{IBD->IBD}(h_i,h_j)$ by its corresponding IBD probability. But as the short IBDs generally have low IBD probability, the effects of weights can be indeed ignored. Therefore, we simply use the aforementioned formulas.

Given all the states, the emission probabilities of each state and the transition probabilities between adjacent states, we can apply Verterbi algorithm (Viterbi, 1967) to find the most likely path of the states, where each state corresponds to a global IBD configuration for the corresponding chunk, and we concatenate all these configurations to obtain a complete IBD configuration for all the haplotypes. The complexity of HMM is $O(c \times k^2)$ where c is the number of IBD chunks, and k is the number of values to be saved for each state. In reality, lots of states have much smaller number of values; therefore, the HMM is generally efficient.

3 EXPERIMENTAL RESULTS

We compare the performance of IBD-Groupon with the MCMC method proposed in Moltke *et al.* (2011) and DASH (Gusev *et al.*, 2011). All the experiments are done on a 2.4 GHz Intel dual core machine with 4 GB memory.

3.1 Simulated data

In Moltke *et al.* (2011), it is shown that for long IBD regions, pairwise IBD detection methods, such as Beagle, GERMLINE, Relate, PLINK generally perform well. The MCMC method is superior to the other methods when the IBD regions shared by multiple individuals are short. Thus, in our experiments, we only focus on detecting short group-wise IBD tracts. We use the same simulation used in Moltke *et al.* (2011), where we randomly generate 104 individuals each with 2 haplotypes, each of length 8 MB with 200 evenly distributed SNPs. Then we set one haplotype of the second individual to be IBD with one haplotype of the third individual for SNPs with index [1, 112], and we set the same haplotype of the second individual to be IBD with one haplotype of the fourth individual for SNPs with index [88, 200]. All the other haplotype regions of all individuals are randomly generated and are not IBD.

Table 1. Performance of MCMC versus IBD-Groupon on simulated data

Method	True positive	Average true positive length	No. of false positive	Average false positive length
MCMC	0.9	33.6	16	16.9
IBD-Groupon (t = 10E-8)	0	0	0	0
IBD-Groupon (t = 10E-7)	0.3	27.8	0	0
IBD-Groupon (t = 10E-6)	0.55	27	0	0
IBD-Groupon (t = 10E-5)	0.6	27	0	0
IBD-Groupon (t = 10E-4)	0.75	26	1	13
IBD-Groupon (t = 10E-3)	0.9	22	8	14

Thus, the group-wise IBD regions are shared by individuals 2, 3 and 4 for SNPs with index [88, 112] of length 25. Moltke *et al*. (2011) showed that the pairwise IBD detection methods are generally not able to detect such short IBD regions. As we use Beagle to detect the pairwise IBD, and by using different threshold, Beagle is able to detect IBD regions with different sizes, we first vary the threshold of Beagle to generate pairwise IBD tracts and run IBD-Groupon on top of the results. All the results are the average performance of 100 randomly generated datasets.

We consider a true positive, as that the true group-wise IBD region is detected, and all the other detected group-wise IBD regions (the number of haplotypes involved needs to be no less than three) are false-positive regions. We also compute the average lengths of the detected true positive group-wise IBD regions and the false-positive group-wise IBD regions. We show the results in Table 1. We can see that the MCMC method achieves high-TP ratio, as reported in the work Moltke *et al*. (2011). However, it also reports relatively large number of FP regions. Also the average length of the TP regions, 33.6, deviates from the ground-truth value 25 a lot. For IBD-Groupon, using the default threshold (10E-8), the group-wise IBD region was never detected, which is also consistent with the findings of Moltke *et al*. (2011), which used the default threshold. However, as we increase the threshold, shorter IBDs are detected and the TP ratio increases as well. The length of the TP regions is much more consistent with the ground-truth value 25 compared with that of the MCMC method. We also observe that when threshold 10E-3 is used, IBD-Groupon achieves the same detection power as MCMC does, but it also suffers from false positives. However, the number of FP is still much smaller than that of MCMC. And the length of FP is generally smaller than that of TP. Threshold 10E-4 seems to achieve the best balance between TP and FP, and thus in our future experiments, we use threshold 10E-4 for Beagle.

Moltke *et al*., (2011) reported that the running time of MCMC is generally much longer than that of other methods. For the simulated data, as we only used 200 SNPs, the running times of MCMC and IBD-Groupon generally do not differ too much. Both of them finished in a few seconds. But as we will show later, when we use a larger dataset, the running time of MCMC increases dramatically.

Table 2. Performance of MCMC on real data

IBD group size	No. of reported IBDs	Average num of reported IBD locus
3	144	123
4	237	75
5	246	52

3.2 Real data

We simulate a pedigree using HapMap according to a real pedigree. The pedigree consists of 184 extant individuals, or individuals of the most current generation, and a family of size 456 with three generations. We select random individuals as ancestral individuals and mate them according to the pedigree to generate the extant individuals. We also make sure the selected individuals do not have parent–child relationship.

We conduct our experiments mainly on chromosome 22, which contains 6159 SNPs. We first consider nine individuals, as MCMC method gets very slow for larger dataset. MCMC runs for a long time even if we only consider a small number of SNPs. For example, for 500 SNPs, MCMC finished in ~50 min. IBD-Groupon, on the contrary, finishes in seconds. As there is a clear advantage of efficiency of IBD-Groupon over MCMC, we did not run MCMC for larger datasets.

For the nine individuals, individuals 2, 3 and 4, individuals 5, 6 and 7 and individuals 8 and 9 are siblings, respectively, according to the pedigree. Therefore, we should expect group-wise IBDs for them. As MCMC finishes fast enough only for 500 SNPs, we consider only the first 500 SNPs of all the haplotypes for the nine individuals. For IBD group size 3, we observe the two groups $\{2, 3, 4\}$ and $\{7, 8, 9\}$ have the longest IBD regions, which are consistent with the fact that the individuals $\{2, 3, 4\}$ and $\{7, 8, 9\}$ are siblings. However, the MCMC program reports lots of false positives. As can be seen in Table 2, besides the two true sibling group-wise IBDs, there are lots of false-positive group-wise IBDs reported, for all group sizes 3, 4 and 5. And even for the two true sibling groups, the length of IBD regions is much higher than expected.

We next run IBD-Groupon on the whole chromosome 22 for all the nine individuals, for 6159 SNPs. We use the threshold 10E-4. We set the top values to be saved for each state as 50 because none of the state has >50 values for all the chunks. IBD-Groupon finished in 9 s and it reports group-wise IBDs of size 3 only for the two sibling groups 2, 3 and 4 and 5, 6 and 7. It indeed never reported any group-wise IBDs that are not consistent with the true pedigree structure, namely, it did not report any false positives.

To show the efficiency of IBD-Groupon with respect to the number of values to be saved for each state and the number of individuals involved, we first fix the number of individuals involved as 50 and vary the number of values to be saved for each state as 10, 30, 50, 70 and 100. The running time of IBD-Groupon is shown in Figure 2a. Next we fix the number of top values to be saved as 100 and vary the number of individuals as

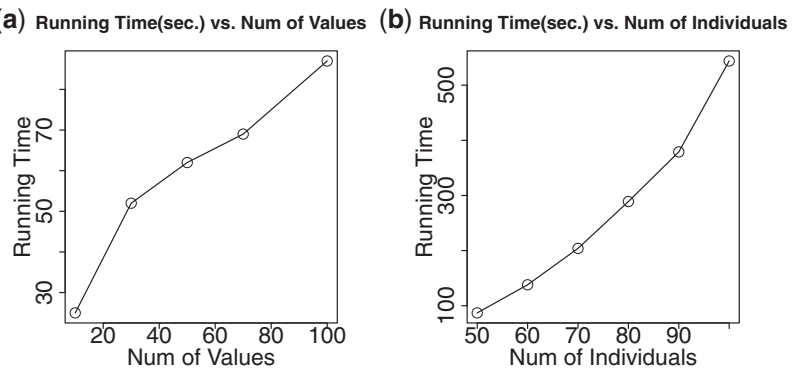

Fig. 2. (**a**) Running time (sec.) of IBD-Groupon for different number of values for each state using 50 individuals. (**b**) Running time (sec.) of IBD-Groupon for different number of individuals using 100 top values

50, 60, 70, 80, 90 and 100. The running time of IBD-Groupon is shown in Figure 2b. We can see that IBD-Groupon is scalable to both parameters, and although we did not show it, obviously IBD-Groupon has the potential to handle much larger datasets.

We also calculate the precision and recall of group-wise IBD regions reported by IBD-Groupon in Table 3 for different number of individuals and save the top-100 values for each state. We define *true positive* as if some group-wise IBD tracts for a sibling group in the pedigree are reported. If the group-wise IBD is not consistent with any of the sibling groups in the pedigree, we call it *false positive*. Notice we only use sibling groups in the pedigree to judge whether the group-wise IBD tracts are true positive or false positive. The false-positive IBD tracts indeed may not be really false positive, as the IBD tracts may be between cousins rather than siblings. We did not distinguish such IBD tracts from false positives as the number of possible relationships could be exponential with the depth and size of the pedigree. Therefore, we just simply use sibling groups instead of considering all family groups with all possible relationships. The precision is the overall precision for different IBD group sizes, and the recall is for IBD groups of different sizes as 2, 3 and 4. The precision is measured as $\frac{\text{reported true positives}}{\text{reported true positives} + \text{reported false positives}}$, and the recall is measured as $\frac{\text{reported true positives}}{\text{all true positives}}$. As we can see the precision drops as the number of individuals increases, as more false positive IBD tracts are reported. We should expect a higher precision if we use all family groups with all possible relationships. The recall for all group sizes is generally good.

To show the effects of the Beagle threshold, we also show the performance of IBD-Groupon using Beagle threshold as 10E-8 in Table 4. We can see that when using a smaller threshold, namely, being stricter on the IBD tracts, the precision of IBD-Groupon increases significantly as most of the reported IBD tracts are true positives. However, the recall drops significantly, as short pairwise IBD tracts are not reported. Thus, when the number of individuals is small, we should use a higher Beagle threshold as both precision and recall are good. When the number of individuals is big, we have to choose between precision and recall. For higher precision, we should use lower threshold and for higher recall, we should use higher threshold.

We also compared Beagle with IBD-Groupon directly to show that even for pairwise IBD tracts, Beagle has low power to detect short IBD tracts. As Beagle only reports pairwise IBD tracts, we only consider pairwise IBD. The results are shown in Table 5. For Beagle, we use the default threshold 10E-8. For IBD-Groupon, we use threshold 10E-4. As Beagle is shown to be reliable for long IBD tracts, we consider IBD tracts with length greater than a threshold, and we vary the length threshold as 100, 200, 300 and 400 SNPs. As we consider the recalls on true positives of all possible lengths, whose number is always fixed, the larger the threshold is, in general, the lower the recalls are. As shown in the table, Beagle has high precision, as it only reports long IBD tracts that are mostly likely true positives. The precision of IBD-Groupon is low when very short IBD tracts are considered (with

Table 3. Performance of IBD-Groupon on real data with respect to different number of individuals

No. of individuals	Precision	Recall (2) (%)	Recall (3) (%)	Recall (4) (%)
10	95	100	100	NA
30	70	100	80	100
50	65	100	83	100
70	51	100	83	100
90	43	100	75	75

Note: The top-100 values are saved for each state. The precision is for all IBD group sizes. The recall is for different IBD group sizes. Beagle threshold = 10E-4.

Table 4. Performance of IBD-Groupon on real data with respect to different number of individuals

Num. of individuals	Precision (%)	Recall (2) (%)	Recall (3) (%)	Recall (4) (%)
10	100	100	100	NA
30	96.4	50	60	0
50	96.8	87.5	66	0
70	87	83	66	0
90	85	86	62	0

Note: The top-100 values are saved for each state. The precision is for all IBD group sizes. The recall is for different IBD group sizes. Beagle threshold = 10E-8.

threshold 100 SNPs). However, IBD-Groupon has much higher recall than Beagle does, as Beagle is not able to detect short IBD tracts (with threshold 100 and 200 SNPs). When the IBD tracts are very long (with threshold 300 and 400 SNPs), both Beagle and IBD-Groupon have the same low recalls, as short IBD tracts are not considered by both of them.

To further evaluate the accuracy of the reported group-wise IBDs, we calculate the average group-wise IBD tract length for groups of sizes as 2, 3 and 4, and we consider 100 individuals and save the top-100 values for each state. It is shown in Donnelly (1983) that the length of IBD tracts follows an exponential distribution $exp(Mr)$, where M is the number of meioses between two individuals, and r is the recombination rate with value as 10^{-8} for the whole genome. Therefore, if we consider 6159 SNPs only, the recombination rate becomes $3 \times 10^9/6159 \times 10^{-8}$. For simplicity, we only consider the group-wise IBD tracts reported by IBD-Groupon for siblings, namely, the numbers of meioses between them are all two. For a group of n siblings, their expected IBD tract length between either paternal or maternal haplotypes is $\frac{1/n}{\frac{3\times10^9}{6159}\times10^{-8}}$. As for each group of siblings, there are two types of haplotypes, one for paternal and one for maternal, the expected IBD tract length should be doubled. We consider the *length* of the IBD tracts simply as the number of SNPs in the IBD tracts. We show the length of IBD tracts reported by IBD-Groupon for different IBD group sizes and that of the expected in Table 6. As we can see, the reported group-wise IBD tract lengths are consistent with those of the expected, indicating that our method is accurate. What is more, most of the group-wise IBD tracts of group size 3 and 4 would not be detected by exiting pairwise IBD tract detection methods, as they are much shorter than the pairwise IBD tracts.

Finally, we compare the performance of DASH (Gusev *et al.*, 2011) and IBD-Groupon for the same 90 individuals. As DASH also requires pairwise IBD tracts as input, we use Beagle to generate the pairwise IBD tracts for both DASH and IBD-Groupon for a fair comparison. We also vary the Beagle threshold as 10E-8 and 10E-4. As DASH has a few parameters (subgraph density, window size and so forth) to tune, we tuned these parameter and showed the best results. DASH finished in 1 s for each set of parameters, whereas IBD-Groupon finished in a few seconds. As we can see in Table 7, IBD-Groupon achieves better precision and recalls in general for both thresholds, indicating using a global probabilistic framework, such as HMM, which is able to generate more accurate results. We also observe that both precision and recall drop for DASH when we increase the threshold of Beagle to produce shorter IBD tracts, indicating that DASH targets mainly on long group-wise IBD tracts, and it lacks of an effective strategy to identify the true short group-wise IBD tracts when many false positives are present. IBD-Groupon on the contrary has much better performance to detect short group-wise IBD tracts, which is a benefit of HMM. IBD-Groupon also has the benefits of parameter free, as the HMM determines the size of the IBD groups and the length of the group-wise IBD tracts.

Table 5. Performance of IBD-Groupon on real data with respect to IBD tracts of different lengths

IBD tracts length (SNPs)	Beagle precision (%)	IBD-Groupon precision (%)	Beagle recall (%)	IBD-Groupon recall (%)
≥ 100	90	34	87	100
≥ 200	90	83	73	93
≥ 300	90	88	67	67
≥ 400	91	86	47	47

Note: The top-100 values are saved for each state. We only consider pairwise IBD tracts. Threshold for Beagle is 10E-8 and for IBD-Groupon is 10E-4.

Table 6. Expected IBD length versus IBD length reported by IBD-Groupon on real data for 100 individuals and top-100 values are saved for each state

	2	3	4
Expected length	236	133	100
Reported length	205	137	102

Note: Length of the IBDs is simply the number of SNPs in the IBDs.

4 CONCLUSION

In this work, we proposed an efficient method IBD-Groupon to detect group-wise IBDs in multiple individuals simultaneously based on pairwise IBD relationships. An HMM is created, and the values of the states are efficiently generated using depth first search and shortest path algorithms. Our method is the first practical method to handle large data sets with hundreds of individuals and thousands of SNPs, and in the meanwhile, has good performance for both long and short group-wise IBD tracts. It is shown that our method is not only much more efficient but also reports much fewer false positives than the

Table 7. Performance of IBD-Groupon on real data with respect to different number of individuals

	DASH				IBD-Groupon			
Threshold	Precision (%)	Recall (2) (%)	Recall (3) (%)	Recall (4) (%)	Precision (%)	Recall (2) (%)	Recall (3) (%)	Recall (4) (%)
10E-8	63	62	32	25	85	86	62	NA
10E-4	43	50	30	13	43	100	75	75

Note: The top-100 values are saved for each state. The precision is for all IBD group sizes. The recall is for different IBD group sizes. Beagle threshold = 10E-8. No. of individuals = 90.

state-of-the-art methods. It also takes LD into consideration, as LD is considered when the pairwise IBD tracts are detected. A possible future work is to apply this method to help reconstruct pedigree, where the group-wise IBD tracts contain much more family relationship information than the pairwise IBD tracts do.

Conflict of Interest: none declared.

REFERENCES

Albrechtsen,A. *et al.* (2009) Relatedness mapping and tracts of relatedness for genome-wide data in the presence of linkage disequilibrium. *Genet. Epidemiol.*, **33**, 266–274.

Bron,C. and Kerbosch,J. (1973) Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, **16**, 575–577.

Browning,B. and Browning,S. (2011) A fast, powerful method for detecting identity by descent. *Am. J. Hum. Genet.*, **88**, 173–182.

Dijkstra,E. (1959) A note on two problems in connection with graphs. *Numerische Mathematik*, **1**, 269–271.

Donnelly,K. (1983) The probability that related individuals share some section of genome identical by descent. *Theor. Popul. Biol.*, **23**, 34–63.

Gusev,A. *et al.* (2009) Whole population, genome-wide mapping of hidden relatedness. *Genome Res.*, **19**, 318–326.

Gusev,A. *et al.* (2011) Dash: a method for identical-by-descent haplotype mapping uncovers association with recent variation. *Am. J. Hum. Genet.*, **88**, 706–717.

Hansen,T. *et al.* (2009) A common greenlandic inuit BRCA1 ring domain founder mutation. *Breast Cancer Res. Treat.*, **115**, 69–76.

He,D. *et al.* (2013) IPED: Inheritance path based pedigree reconstruction algorithm using genotype data. In: *Research in Computational Molecular Biology, 2013*. pp. 75–87. Springer, Berlin/Heidelberg.

Leibon,G. *et al.* (2008) A SNP streak model for the identification of genetic regions identical-by-descent. *Stat. Appl. Genet. Mol. Biol.*, **7**, 1–17.

Moltke,I. *et al.* (2011) A method for detecting IBD regions simultaneously in multiple individuals with applications to disease genetics. *Genome Res.*, **21**, 1168–1180.

Purcell,S. *et al.* (2007) Plink: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, **81**, 559–575.

Thomas,A. *et al.* (2007) Shared genomic segment analysis. Mapping disease predisposition genes in extended pedigrees using SNP genotype assays. *Ann. Hum. Genet.*, **72**, 279–287.

Viterbi,A. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, **13**, 260–269.