Programming Research Group

EXACT AND HEURISTIC APPROACHES FOR IDENTIFYING DISEASE-ASSOCIATED SNP MOTIFS

Gaofeng Huang
Peter Jeavons
Dominic Kwiatkowski

PRG-RR-05-03

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

Abstract

A Single Nucleotide Polymorphism (SNP) is a small DNA variation which occurs naturally between different individuals of the same species. Some combinations of SNPs in the human genome are known to increase the risk of certain complex genetic diseases. This paper formulates the problem of identifying such disease-associated SNP motifs as a combinatorial optimization problem and shows it to be $\mathcal{NP}$-hard. Both exact and heuristic approaches for this problem are developed and tested on simulated data and real clinical data. Computational results indicate that our algorithms are efficient AI tools which can support ongoing biological research.

Keywords: AI in bioinformatics, combinatorial optimization, genome analysis, association analysis, Single Nucleotide Polymorphism

1 Biological Background

The DNA sequences of different individuals within the same species are highly conserved but not exactly the same. The variations that occur at a particular location within the DNA sequence in different individuals within a population are known as alleles. One common type of variation is called a Single Nucleotide Polymorphism (SNP), where a single position within a DNA sequence is altered from one nucleotide base to another (Figure 1). It is believed that SNPs contain important genetic information that influences how individuals differ in their risk of certain diseases or their response to particular drugs [1].

	SNP1 SNP2 SNP3
	↓ ↓ ↓
Individual 1	...ATGCC**G**ATTAGCGT...
Individual 2	...ATGCCTATTAGCGT...
Individual 3	...ATGCC**G**ATTA**T**CGT...
Individual 4	...AT**A**CCTATTAGCGT...
Individual 5	...ATGCCTATTA**T**CGT...

Figure 1: An example with 3 SNPs (loci)

A general research question within the field of human genetics is to ask whether a disease of interest is related to the occurrence of particular SNPs within the human genome, i.e., whether there are SNPs that can be seen as causative for that disease. This question is extremely hard to answer in most cases due to the fact that the whole human genome contains more than 3000 million nucleotide bases. It is currently technically and economically impossible to screen the whole genome of all patients. Hence, as a first step, several SNP databases are being established to maintain information about common SNPs, such as the dbSNP database (`http://www.ncbi.nlm.nih.gov/projects/SNP/`). However, the frequency of SNPs occurring in the human genome is estimated to be 1 SNP per 300 bases, which means that there are still about 10 million common SNPs to be investigated. To identify the causative SNPs among them is like finding a needle in a hay stack.

Fortunately, the phenomenon of Linkage Disequilibrium (LD) makes some progress possible [2]. Two SNPs occurring at different locations within the genome are said to be in Linkage Disequilibrium if their respective alleles do not associate independently in the population. This is an important property of many pairs of SNPs, which means that we can infer some SNPs by observing others. For example, suppose the whole population contains the 5 major genotypes shown in Figure 1, in this case we would say that SNP1 and SNP2 are in high LD, because if we observe an "`A`" in SNP1, there must be a "`T`" in SNP2, and similarly if we observe a "`G`" in SNP2, we can infer that there must be a "`G`" in SNP1. On the other hand, SNP2 and SNP3 are in low LD, since they seem to be independent. To calculate the LD value between two SNP loci, several quantitative LD measures have been developed, including D' [3] and λ [4]. Further progress could be made with additional information about the LD among multiple SNP loci. Hence, the International HapMap Project (`http://www.hapmap.org/`), which aims to produce a map of all the common SNPs in the human genome, was started in October 2002 and is currently ongoing.

With the aid of Linkage Disequilibrium, it is possible to home in on the causative SNPs for a particular disease by using a sequence of "sampling-analyze-resampling" cycles, and we will now describe the basic principles of this approach.

Under this framework there are two different scenarios depending on how much *a priori* knowledge we have about the causative SNPs. In the first scenario, where we know nothing about where the causative SNPs might be, we have to do a genome-wide scan which samples a small proportion of the SNPs from the whole genome and outputs the estimated region which will probably contain the causative SNPs. This problem is hard, and has been the subject of extensive research by statisticians [5, 6, 7, 8]. Currently, the most popular statistical methods attempt to model possible genealogical trees, and hence estimate the location of the disease causative SNP. Some research in this area has also been done by computer scientists, including HPM (Haplotype Pattern Mining) [9], which has had some success, since Laitinen et al. report in Science [10] an asthma gene located with the aid of HPM. All these methods are statistically powerful, and can often locate candidate causative SNPs with a resolution of about 50-100Kb (where a Kb is 1000 nucleotide bases).

In the second scenario, we assume that the causative SNPs are known to lie within some small region of the genome (50-100Kb). This information may be obtained by genome-wide scanning, or other biological analysis. Even in this case it is still not trivial to determine whether particular SNPs are "significantly" associated with the disease. One reason is that complex genetic diseases are influenced by both genetic and environmental factors; even when people have the same genotype, some of them get the disease while others do not, due to their different living environments, lifestyles, etc. In a word, the "causative" SNPs only affect the *risk* of getting the disease in question. Another difficulty is due to the *combinatorial* nature of disease association with SNPs, which has recently been observed in biological research. For example, with the data shown in Figure 2, if we only examine each single SNP locus, we cannot find any significant association between each individual SNP and the disease. However, if we look at the combination of SNP1 and SNP3, we notice that the combination of "A" and "T" leads to high disease risk. The explanation is that this particular combination of these two SNPs might be in high LD with a third SNP, which is the real causative SNP (but has not been measured - remember that it is not realistic at present to screen a patient for all possible SNPs, even in a restricted region).

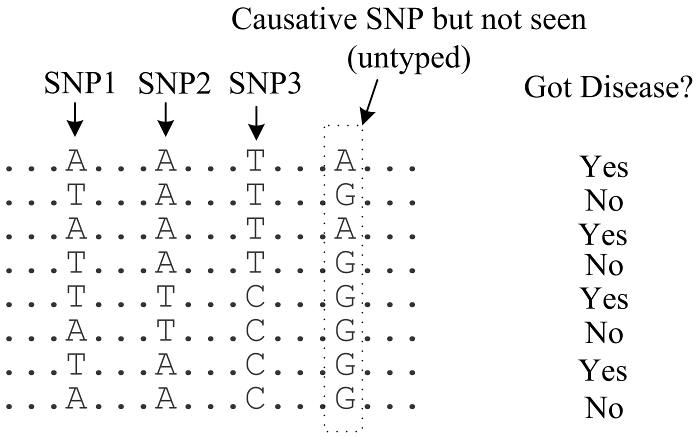

Figure 2: Combinatorial feature of SNPs

In this paper, we will focus on the second scenario in the context of a classical clinical case-control study: given several candidate SNP loci within a small region, and the SNP data observed from both cases (patients) and controls (healthy individuals), the problem is to identify the most significant SNP *combination* (motif) that associates with the disease. At present, this task is generally done manually by experienced biologists. In cooperation with a biological laboratory, our aim is to develop efficient AI tools to do this job.

The rest of the paper is organized as follows. In Section 2, we formulate the problem more precisely as a combinatorial optimization problem. In Section 3 we show this problem to be $\mathcal{NP}$-hard. In Sections 4 and 5, both exact and heuristic approaches are developed. Computational results are presented in Section 6 for both simulated data and real data. And in the final section, we summarise our conclusions and discuss directions for future work.

2 Problem Formulation

Throughout this paper the input SNP data for an individual will be represented in a standard way by a vector of m values over the alphabet $\{0, 1, 2\}$. The reason for using this alphabet is that almost all (99%) of the SNPs in the human genome are bi-alleles, which means that there are only two kinds of bases that occur at each SNP locus, for example `A/T`. Biologists usually use "1" to denote the major allele (i.e. the one with the highest frequency in the human population) and "2" to denote the minor allele.

Some allele values may be missing (unknown) due to experimental reasons, and these are denoted with "0".

The SNP data from n_p patients (cases) and n_h healthy individuals (controls) will be represented by two matrices $M^{(p)}_{n_p \times m}$ and $M^{(h)}_{n_h \times m}$. Each row of such a matrix represents the data for a single individual for the m SNPs under consideration. Figure 3 shows an example. The vector in row i of matrix $M^{(x)}$, i.e. the data for individual i, will be denoted $M^{(x)}_i$.

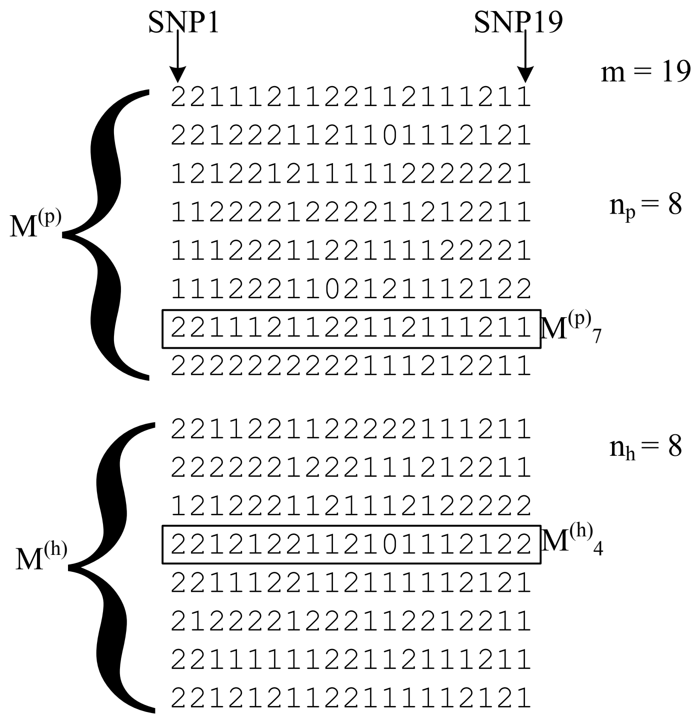

Figure 3: SNP data in Matrix Representation

A SNP *motif* is an expression of the form "`--11---2-`", where "`-`" means "don't care". A motif can be mathematically represented by two vectors, a position vector $P = (p_1, p_2, ..., p_k), 1 \leq p_1 < p_2 < ... < p_k \leq m$ and a data vector $D = (d_1, d_2, ..., d_k), d_i \in \{1, 2\}$, which means that the allele in each position p_i should be d_i. Hence, the motif "`--11---2-`" will be represented as $P = (3, 4, 8), D = (1, 1, 2)$.

The matching function between a motif (P, D) and an individual data vector $M^{(x)}_i$ is defined as follows:

$$Match((P,D), M^{(x)}_i) = \begin{cases} 1\ (true), & \text{if} \quad \forall k, M^{(x)}_{i,p_k} = 0 \ or \ M^{(x)}_{i,p_k} = d_k \\ 0\ (false), & \text{otherwise.} \end{cases}$$

Hence the number, a, of cases that match a motif (P, D) is given by $a = \sum_{i=1}^{n_p} Match((P, D), M^{(p)}_i)$

and the number, b, of controls that match the motif (P, D) is given by $b = \sum_{i=1}^{n_h} Match((P, D), M_i^{(h)})$.

To measure the significance of a motif, a standard method is to put these numbers into a 2×2 contingency table (see Figure 4) and perform a standard chi-squared test [11].

	#Match	#non-Match	Total
#Cases	a	c=n_p−a	n_p
#Controls	b	d=n_h−b	n_h
Total	e=a+b	f=n−e	n=n_h+n_p

Figure 4: 2×2 contingency table

A quick formula for calculating the value of the chi-squared statistic is:

$$\chi^2(a, b) = \frac{n(ad - bc)^2}{efn_pn_h} = \frac{n(an_h - bn_p)^2}{(a + b)(n - a - b)n_pn_h}$$

For details about the chi-squared test, and this quick formula, please refer to [11].

Therefore, our problem is formulated as finding a motif (P^*, D^*) which yields the maximal possible value for χ^2, given the matrices $M_{n_p \times m}^{(p)}$ and $M_{n_h \times m}^{(h)}$. This problem will be called the SNP motif identification problem.

3 $\mathcal{NP}$-hardness

In this section, we will show that the SNP motif identification problem defined in the previous section is $\mathcal{NP}$-hard, by constructing a reduction from the standard MAX-SAT problem.

The MAX-SAT (Maximum Satisfiability) problem is a well-known $\mathcal{NP}$-hard problem [12], which can be stated as follows. Let $x_1, x_2, ..., x_m$ be m boolean variables. A clause C_i is a disjunction of $|C_i|$ literals, i.e. $C_i = \bigvee_{j=1}^{|Ci|} l_{i,j}$, where each literal $l_{i,j}$ is either of the form x_j or $\bar{x}_j$. Given a conjunction of n clauses $\bigwedge_{i=1}^{n} C_i$, the MAX-SAT problem is to find an assignment of boolean values to the variables $x_1, x_2, ..., x_m$, such that the number of satisfied (true) clauses is maximized.

To reduce MAX-SAT to our SNP motif identification problem we proceed as follows. As a first step, given any instance of MAX-SAT, we can construct a corresponding instance of the SNP motif identification problem by the following procedure:

- Let the m boolean variables of the MAX-SAT instance correspond to m SNP loci.

- Construct matrix the $M^{(h)}$ by transforming each clause C_i to the vector $M_i^{(h)}$, as follows:

$$M_{i,j}^{(h)} = \begin{cases} 1 & \text{if } x_j \text{ exists in clause } C_i \\ 2 & \text{if } \bar{x}_j \text{ exists in clause } C_i \\ 0 & \text{otherwise} \end{cases}$$

- Let the matrix $M^{(p)}$ consist of a single row containing only zeros.

For example, the MAX-SAT instance $(x_1 \vee x_2 \vee \bar{x_3}) \wedge (\bar{x_1} \vee x_3 \vee x_4) \wedge (\bar{x_1} \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x_4}) \wedge (\bar{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee \bar{x_2} \vee \bar{x_4})$ will be transformed into:

$$\begin{array}{ll} & \begin{array}{cccc} \text{SNP1} & & & \text{SNP4} \\ \downarrow & & & \downarrow \end{array} \qquad m = 4 \\ M^{(p)} & \left\{ \begin{array}{cccc} 0 & 0 & 0 & 0 \end{array} \right. \qquad n_p = 1 \\ M^{(h)} & \left\{ \begin{array}{cccc} 1 & 1 & 2 & 0 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 2 \end{array} \right. \qquad n_h = 6 \end{array}$$

We will now show that an optimal solution to this artificial instance always corresponds to an optimal solution to the original MAX-SAT instance. Since the matrix $M^{(p)}$ only contains a single line with all zeros, which will match any motif, we will always have the number of matches $a = 1$ for any motif. Therefore, the objective function

$$\chi^2(a,b) = \frac{n(an_h - bn_p)^2}{(a+b)(n-a-b)n_p n_h}$$

becomes

$$\begin{aligned} \chi^2(1,b) &= \frac{(n_h+1)(1 \times n_h - b \times 1)^2}{(1+b)(n_h+1-1-b) \times 1 \times n_h} \\ &= \frac{(n_h+1)}{n_h} \times \frac{(n_h-b)}{(1+b)}, \end{aligned}$$

which is a decreasing function with increasing b. This means that an optimal solution to the artificial instance is one which minimizes b, i.e. a motif (P^*, D^*) which matches the least number of lines in $M^{(h)}$. Note that we can assume that $P^* = (1,2,3,...,m)$ (otherwise, we could add some extra position, which would not increase b). We can transform (P^*, D^*) to a solution X^* of the original MAX-SAT instance by setting:

$$x_j^* = \begin{cases} \text{true} & \text{if } d_j^* = 2 \\ \text{false} & \text{if } d_j^* = 1 \end{cases}$$

If the motif (P^*, D^*) does not match line i in $M^{(h)}$, then there is some position j such that $M^{(h)}_{i,j} \neq 0$ and $M^{(h)}_{i,j} \neq d^*_j$. There are two cases:

1. $M^{(h)}_{i,j} = 1$ and $d^*_j = 2$

   In this case $M^{(h)}_{i,j} = 1$ means that clause C_i contains x_j while $d^*_j = 2$ means $x^*_j = true$, so clause C_i is satisfied.

2. $M^{(h)}_{i,j} = 2$ and $d^*_j = 1$

   In this case $M^{(h)}_{i,j} = 2$ means that clause C_i contains $\bar{x_j}$ while $d^*_j = 1$ means $x^*_j = false$, so clause C_i is again satisfied.

Hence, if motif (P^*, D^*) does NOT match line i in $M^{(h)}$, then clause C_i is satisfied by assignment X^*. A similar argument shows that the converse is also true. Since (P^*, D^*) is a motif which matches the least number of lines in $M^{(h)}$ (minimizes b), this means that the number of lines that motif (P^*, D^*) does NOT match is maximized. Therefore, the number of clauses that assignment X^* satisfies is maximized, i.e. X^* is an optimal solution for the original MAX-SAT problem.

In this way, we have reduced the MAX-SAT problem to our problem. Since MAX-SAT is known to be $\mathcal{NP}$-hard, our problem of finding a motif (P^*, D^*) with the maximal value for χ^2 is also $\mathcal{NP}$-hard.

4 Exact Algorithm

A straight-forward exhaustive search algorithm needs to explore $O(3^m)$ motifs, and for each motif it takes $O((n_p + n_h)m)$ time to test matching and compute a and b. In this section, we will develop an effective exact algorithm using a branch-and-prune tree search technique, which dramatically reduces the search space.

4.1 State Representation

We will search for the motif (P, D) in a sequence of steps. At step j, we enumerate the possible choices of p_j and d_j. Figure 5 illustrates the hypothetical search tree.

Each state (node) at level j in the tree determines a motif $(P', D'), P' = (p_1, p_2, ..., p_j), D' = (d_1, d_2, ..., d_j)$, where each p_i and d_i can be retrieved through the path from the node to the root node. We represent each state as a tuple (j, S^+, S^-), where S^+ is the set of lines in $M^{(p)}$ which match motif (P', D'), i.e.

$$S^+ = \{\ \ i \ \ |\ \ Match((P', D'), M^{(p)}_i) = 1, i = 1, 2, ..., n_p\},$$

and similarly

$$S^- = \{\ \ i \ \ |\ \ Match((P', D'), M^{(h)}_i) = 1, i = 1, 2, ..., n_h\}.$$

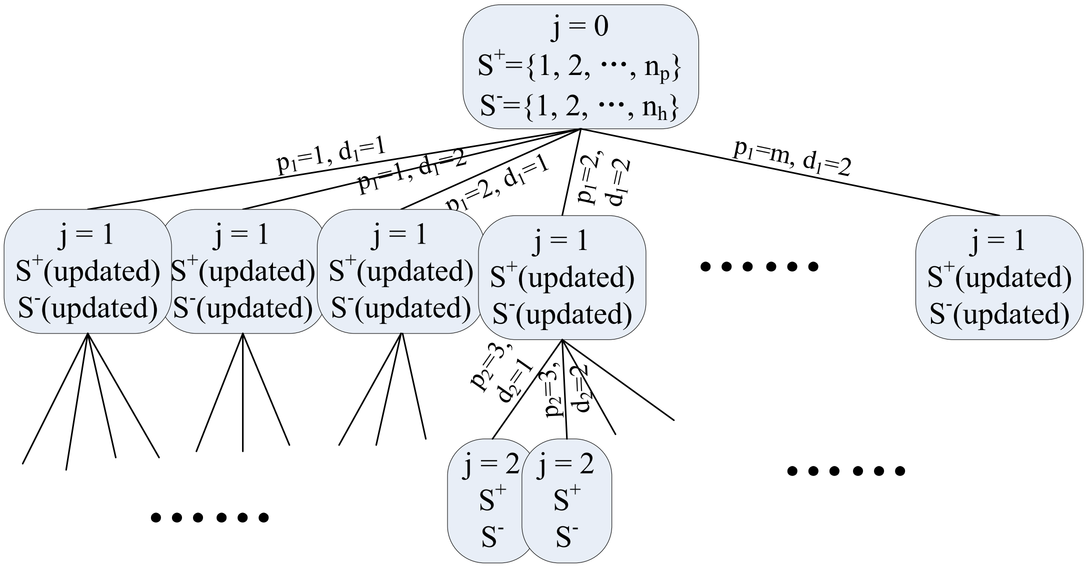

Figure 5: hypothetical search tree

The root node is $(j = 0, S^+ = \{1, 2, ..., n_p\}, S^- = \{1, 2, ..., n_h\})$. Given a state (j, S^+, S^-), its child state $(j + 1, S'^+, S'^-)$, along the branch (p_{j+1}, d_{j+1}), can be generated by

$$S'^+ \;=\; \{\ i\ \ |i \in S^+ \text{ and } M^{(p)}_{i,p_{j+1}} = d_{j+1}\} \tag{1}$$

$$S'^- \;=\; \{\ i\ \ |i \in S^- \text{ and } M^{(h)}_{i,p_{j+1}} = d_{j+1}\} \tag{2}$$

Hence, generating a new state, i.e. a new motif, takes $O(n_p + n_h)$ time. Since $a = |S'^+|$ and $b = |S'^-|$, we can calculate the χ^2 objective function in $O(n_p + n_h)$ as well. Hence we have reduced the time complexity of the matching test for each individual motif from $O((n_p + n_h)m)$ to $O(n_p + n_h)$.

4.2 Branching Strategy and Pruning Rules

We use a Depth-First-Search strategy due to memory usage consideration. Several pruning rules are applied to reduce the search space:

Ordering Rule

When we generate the child nodes of node (j, S^+, S^-), we only consider those branches (p_{j+1}, d_{j+1}) with $p_{j+1} > p_j$. This is implied by our motif definition, i.e. the position vector $P = (p_1, p_2, ..., p_k)$ has $1 \leq p_1 < p_2 < ... < p_k \leq m$.

Absorption Rule

When we generate a child node $(j + 1, S'^+, S'^-)$ of node (j, S^+, S^-) following branch $(p_{j+1} = p^*, d_{j+1} = d^*)$, as shown in Figure 6, if $S'^+ = S^+$ and $S'^- = S^-$, the child node $(j + 1, S'^+, S'^-)$ can be "absorbed", and therefore the whole subtree T' rooted at node $(j + 1, S'^+, S'^-)$ can be pruned, as we will now explain.

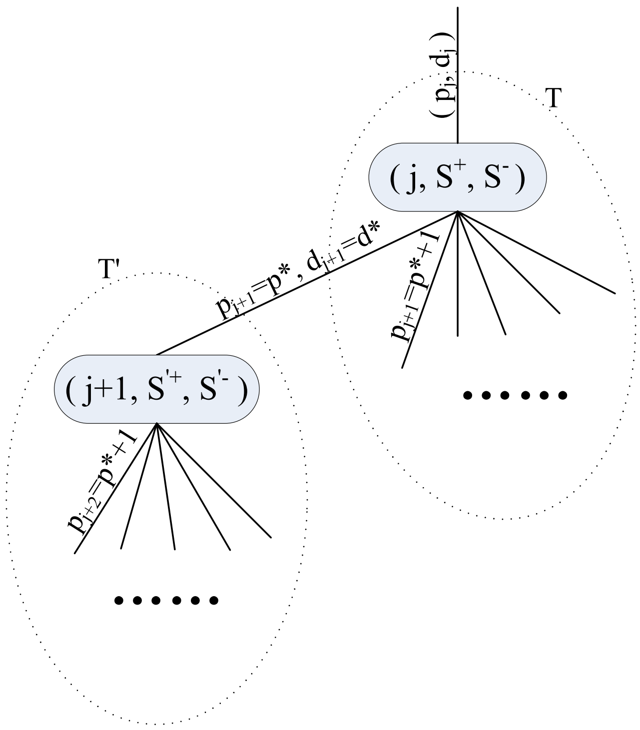

Figure 6: Absorption Rule

Recall the calculation of S'^{+} and S'^{-} in Equations (1) and (2). If $S'^{+} = S^{+}$ and $S'^{-} = S^{-}$, they imply $\forall i \in S^{+}, M^{(p)}_{i,p^*} = d^*$ and $\forall i \in S^{-}, M^{(h)}_{i,p^*} = d^*$, that is to say, for both cases or controls, all the remaining individuals have the same allele d^* in the SNP locus p^*. Therefore, adding position p^* does not provide any more information. In other words, subtree T' and subtree T in Figure 6 are identical, so we can prune subtree T' and explore subtree T only.

As a concrete example, consider the following SNP data:

$$
\begin{array}{ccccccc}
2 & 2 & 1 & 2 & 2 & 2 & 1 \\
1 & 2 & 2 & 1 & 1 & 2 & 1 \\
2 & 1 & 1 & 2 & 2 & 1 & 2 \\
1 & 2 & 1 & 1 & 2 & 1 & 1 \\
2 & 2 & 1 & 2 & 1 & 2 & 1 \\
2 & 1 & 1 & 2 & 2 & 2 & 2 \\
1 & 2 & 2 & 2 & 2 & 2 & 1
\end{array}
$$

Motif “`1------`” and “`12-----`” match the same set of lines; motif “`1----2-`” and “`12---2-`” also match the same set of lines; motif “`1---2--`” and “`12--2--` match the same set of lines as well. These are all because after restricting the first SNP locus to be

"1", all of the remaining lines have allele "2" in the second SNP locus; further restricting the second SNP locus to be "2" does not provide more information. Therefore, the whole subtree can be pruned.

Sibling Rule

Suppose that we generate a child node $(j+1, S'^{+}, S'^{-})$ of node (j, S^{+}, S^{-}), following branch $(p_{j+1} = p^*, d_{j+1} = d^*)$, and node (j, S^{+}, S^{-}) has a sibling node (j, S''^{+}, S''^{-}) which follows branch $(p_j = p^*, d_j = d^*)$ from its parent, as shown in Figure 7. Node (j, S''^{+}, S''^{-}) can be pruned if $S''^{+} = S'^{+}$ and $S''^{-} = S'^{-}$, as we will now explain.

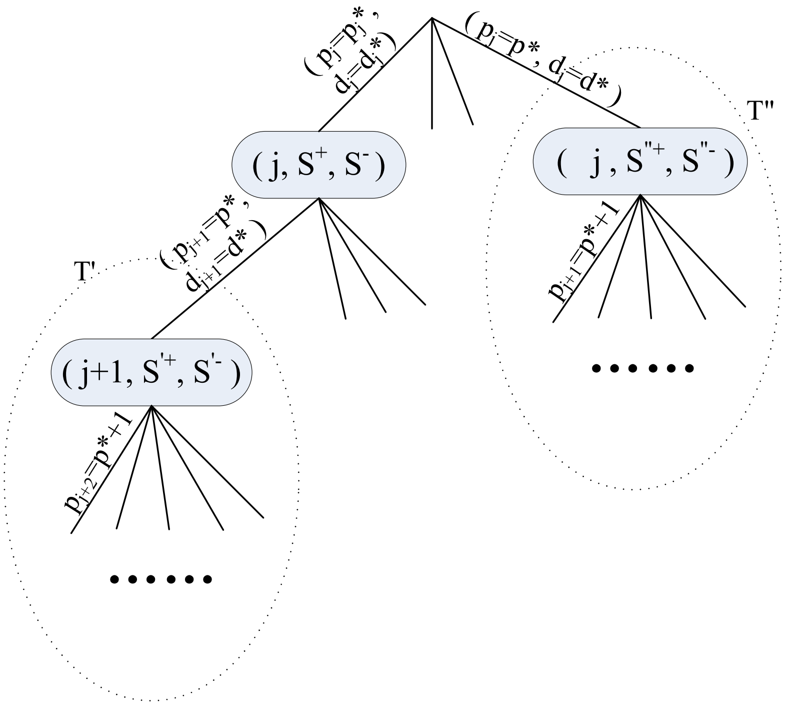

Figure 7: Sibling Rule

It is clear that node $(j+1, S'^{+}, S'^{-})$ and node (j, S''^{+}, S''^{-}) only differ in one SNP locus: p_j^*. The equations $S''^{+} = S'^{+}$ and $S''^{-} = S'^{-}$ imply that adding position p_j^* does not provide more information. In other words, for any node (k, S^{*+}, S^{*-}) in the subtree T'' obtained by following the path $(p_j = p^*, d_j = d^*), (p_{j+1} = p_{j+1}^*, d_{j+1} = d_{j+1}^*), ..., (p_k = p_k^*, d_k = d_k^*)$, there exists a corresponding node $(k+1, S^{*+}, S^{*-})$ in the subtree T' obtained by following the path $(p_j = p_j^*, d_j = d_j^*), (p_{j+1} = p^*, d_{j+1} = d^*), (p_{j+2} = p_{j+1}^*, d_{j+2} = d_{j+1}^*), ..., (p_{k+1} = p_k^*, d_{k+1} = d_k^*)$, and vice versa. Hence, subtree T' and subtree T'' are identical, and we only need to explore one of them.

5 Heuristic Approach

Our heuristic approach includes two stages: a search stage and a refinement stage. In the search stage, we try to get a list of candidate motifs with high χ^2 value. Each of these motifs is refined in the refinement stage and the most significant one is returned as our final result.

5.1 Search Stage

The search stage is adapted from the exact search algorithm described in the previous section by extending the absorption rule. In Figure 6, a child node $(j+1, S'^+, S'^-)$ is absorbed by its parent node (j, S^+, S^-) if $S'^+ = S^+$ and $S'^- = S^-$, because adding position p^* does not provide more information. Now we extend this rule: a child node $(j+1, S'^+, S'^-)$ is absorbed by its parent node (j, S^+, S^-) if S'^+ is highly similar to S^+ and S'^- is highly similar to S^-. Keeping in mind that $S'^+ \subset S^+$ and $S'^- \subset S^-$, the similarity conditions can be formulated as follows:

$$\begin{aligned} |S'^+| &\geq 0.95|S^+| \\ |S'^-| &\geq 0.95|S^-| \end{aligned}$$

Using this rule, when we generate the children of node (j, S^+, S^-) we prune those branches which do not provide "enough" information. This pruning is very effective because it makes use of the relationship between SNP loci, i.e. Linkage Disequilibrium. Although we cannot guarantee to find the optimal motif this way, this approach will find a list of candidate motifs with high χ^2 value very quickly.

5.2 Refinement Stage

To refine a motif in the candidate list obtained from the search stage, we try to put back those SNP loci which are ignored (pruned) during the search stage. We illustrate this idea through an example. Suppose that we have a candidate motif "`--2--12-`", i.e. $P = (3,6,7), D = (2,1,2)$, so its path to the root node in the search tree is $(p_1 = 3, d_1 = 2), (p_2 = 6, d_2 = 1), (p_3 = 7, d_3 = 2)$. And we have pruned some branches in this path during the search stage, for example $(p'_1 = 8, d'_1 = 1), (p'_2 = 5, d'_2 = 1)$. Now we try to put back these two SNP loci, i.e. we try to find the combination within these 5 SNP loci that gives the highest χ^2 value, which again can be done by our exact search algorithm. Suppose here we find the combination of $(p_1 = 3, d_1 = 2), (p'_2 = 5, d'_2 = 1), (p_2 = 6, d_2 = 1), (p'_1 = 8, d'_1 = 1)$ gives the highest χ^2 value, so we refine the candidate motif "`--2--12-`" to be a more significant one "`--2-11-1`".

In the refinement stage, all motifs in the candidate list obtained from the search stage are refined, and the most significant motif obtained is returned as the final result.

6 Computational Results

We tested our algorithms on both simulated and real clinical data. All experiments were performed on an IBM Pentium 1.5GHz laptop with 512MB of memory.

6.1 Simulated Data

At present the best model for Linkage Disequilibrium between SNP loci is still unclear, but it is unreasonable to simply generate random SNP data without attempting to model the Linkage Disequilibrium. We therefore use a new way to generate the simulated data, suggested by the third author of this paper, based on experience with biological data. The basic idea is to make use of the real SNP data from the HapMap Project [1]. The procedure is as follows:

- **Step 1: Obtain raw data from the HapMap Project (`http://www.hapmap.org/`).**

  The HapMap raw data (release 14) contains 60 unrelated Caucasian (CEU) individuals. We randomly select $m + 1$ consecutive SNP loci from chromosome 20.

- **Step 2: Estimate haplotype frequency using SNPHAP [13].**

  For simplicity of problem description in this paper we haven't used the terms "haplotype" and "genotype"; instead we have used the general term "SNP data". In fact, in this paper we only deal with SNP haplotype data, i.e. a vector of $\{0, 1, 2\}$ for each individual, but the raw data obtained in Step 1 are genotype data. (We will explain this distinction further in the final section as a direction of our future work.)

  In order to generate simulated haplotype data in the form that we need we use a popular program, SNPHAP [13], to infer SNP haplotype frequency from the raw genotype data obtained in Step 1. A sample output from SNPHAP is shown below :

haplotype SNP data	estimated proportion in population
1111111121111111111	0.118307
1221121111111111111	0.050000
1111111121111111112	0.047272
1221221111211112111	0.036585
1112111111111112112	0.033333
1121121121111211111	0.028571
1121121121111111111	0.028571
1111211121211112111	0.025000
1221121121111211122	0.025000
1111111111111111112	0.025000

- **Step 3: Generate a population of 100000 individuals according to the frequency table obtained from SNPHAP**.
- **Step 4: Simulate an unknown disease and remove the causative SNP**

  Among the $m + 1$ SNP loci in our simulated data, we randomly assume one SNP locus $p*$ to be the disease causative SNP. Then for each individual, if there is an allele "1" in SNP locus $p*$, we assign a risk r_a of having the disease; if there is an allele "2" in SNP locus $p*$, we assign the risk r_b of having the disease. In our experiment, we set $r_a = 0.01, r_b = 0.02$. Now we simulate whether each individual gets the disease or not by making a random assignment weighted according to the individual's assigned disease risk.

  After that, we remove the column of the disease causative SNP, locus $p*$, and only use the remaining m SNP loci. This simulates a typical scenario, where the disease causative SNP locus is not directly measured by the researcher.
- **Step 5: Sampling cases and controls**

  Finally, we simulate a clinical scenario. A classical case-control study samples the same number of cases and controls. Here we consider all individuals who get the disease to be cases. Then we randomly sample the same number of healthy individuals from the whole population to be controls.

The major advantage of our simulation is that the Linkage Disequilibrium between SNP loci is realistic and natural.

We carried out experiments with $m = 20, 25, 30, 35, 40, 45, 50, 55, 60$. For each m, we did 10 simulations to generates 10 testcases. Both exact and heuristic approaches were tested on these $9 \times 10 = 90$ testcases. Computational results are reported in Table 1. Each row presents the average result for the 10 testcases with the same m. The first column is the average number of nodes (states) that the exact search algorithm explores. The second column is the average CPU time in seconds that the exact search algorithm takes. Similarly, the third and forth columns are for the heuristic approach. The last column shows for how many testcases (among the 10 testcases with the same m) the solution that the heuristic returns is optimal, i.e. the same solution as the exact algorithm returns.

Recall that a straight-forward exhaustive search algorithm would need to explore 3^m nodes. These results provide convincing evidence that our branch-and-prune algorithm is considerably more efficient than this brute-force approach. As shown in Table 1, a large number of nodes are pruned, and it can deal with the testcases with $m \leq 45$ within one minute.

The results also indicate that our heuristic method can provide high quality motifs using less time. As a matter of fact, for all the 60 testcases which we can compare, the motifs that the heuristic method provides are always optimal.

6.2 Real Clinical Data

Finally, we obtained two real clinical testcases from the biological laboratory. (Due to privacy policy and copyright reasons, we are not allowed to publish the full details of

Table 1: Results for Simulated Data ($9 \times 10 = 90$ testcases in total)

	Exact Search		Heuristic		
m	average nodes	average time (sec)	average nodes	average time (sec)	#optimal
20	448.9	0.26	401.9	0.39	10
25	3893.1	1.51	2472.6	1.25	10
30	6700.6	3.77	4078.6	2.89	10
35	16516.5	5.89	7571.7	3.93	10
40	45969.5	16.55	15732.5	8.10	10
45	94535.1	36.25	26705.5	12.81	10
50	-	>1min	45732.6	18.15	-
55	-	>1min	78155.2	33.90	-
60	-	>1min	108977.7	54.77	-

these datasets.)

The first clinical testcase contains $m = 19$ SNP loci, $n_p = 1362$ patients and $n_h = 914$ controls. Table 2 shows the computational results. As mentioned before, currently the identification of disease-associated SNP motifs is done manually by experienced biologists. The first row of Table 2 shows the result that we obtained from the biological laboratory. The second and third rows are results provided by our algorithms.

Table 2: Real Clinical Data 1

	Motif	#cases	#controls	χ^2 value
Manual Result	`11-2----2--------2-`	93	113	19.69
Exact Search	`11-2----2-11-----21`	89	112	22.22
Heuristic	`11-2----2-11-----21`	89	112	22.22

As shown in Table 2, the results are very similar, which indicates the success of our algorithms in some sense.

The second clinical testcase is one of the newest datasets which the laboratory is currently working on. It contains $m = 37$ SNP loci, $n_p = 1146$ patients and $n_h = 1152$ controls. Computational results are reported in Table 3. This testcase is much "harder", and the exact search algorithm does not terminate within 3 minutes. Hence only the result of the heuristic is shown here.

As shown in Table 3, we find a much more significant motif than the one found manually. This result has been sent back to the biological laboratory for further investigation.

Table 3: Real Clinical Data 2

	Motif	#cases	#controls	χ^2 value
Manual Result	`-----------------------2-12---------`	96	50	13.19
Heuristic	`--1--12----2----------222-----2------`	132	70	21.22

7 Conclusions and Future Work

In this paper, we studied the problem of identifying disease-associated SNP motifs. We formulated it as a combinatorial optimization problem and showed it to be $\mathcal{NP}$-hard. Both exact and heuristic approaches for this problem were developed. We conducted extensive experiments on both simulated and real data. Computational results indicate that our algorithms are efficient AI tools which can support ongoing biological research.

One possible direction of our future work is to develop algorithms directly dealing with genotype SNP data. So far in this paper we only deal with so-called haplotype SNP data, but there are, in fact, two copies of (almost) all chromosomes in each individual: one from the father and one from the mother. One limitation of current technology is that we can determine the two alleles in each SNP locus, but we don't know which one is from which chromosome. For example, for three SNP loci, the data we could get through biological experiments is something like "11 12 12", which is called unphased genotype data. We don't know whether it is the combination of haplotype "1 1 1" and "1 2 2", or haplotype "1 1 2" and "1 2 1" (or one of the other possible combinations). To infer haplotype data from genotype data is another very active research topic [13, 14, 15]. In this paper, we use the SNPHAP program [13] as a preprocessor to obtain haplotype data. Our plan for the future is to develop algorithms which can deal directly with the unphased genotype SNP data.

References

[1] The International HapMap Consortium: The International HapMap Project. Nature **426** (2003) 789–796

[2] Goldstein, D.B., Weale, M.E.: Population genomics: Linkage disequilibrium holds the key. Curr. Biol. **11** (2001) 576–579

[3] Lewontin, R.C.: The interaction of selection and linkage. I. general considerations; heterotic models. Genetics **49** (1964) 49–67

[4] Terwilliger, J.D.: A powerful likelihood method for the analysis of linkage disequilibrium between trait loci and one or more polymorphic marker loci. Am. J. Human Genet. **56** (1995) 777–787

[5] McPeek, M.S., Strahs, A.: Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping. Am. J. Hum. Genet. **65** (1999) 858C875

[6] Liu, J.S., Sabatti, C., Teng, J., Keats, B.J.B., Risch, N.: Bayesian analysis of haplotypes for linkage disequilibrium mapping. Genome Research **11** (2001) 1716C1724

[7] Morris, A.P., Whittaker, J.C., Balding, D.J.: Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. Am. J. Hum. Genet. **70** (2002) 686C707

[8] Lu, X., Niu, T., Liu, J.S.: Haplotype information and linkage disequilibrium mapping for single nucleotide polymorphisms. Genome Research (2003) 2112C2117

[9] Toivonen, H.T.T., Onkamo, P., Vasko, K., Ollikainen, V., Sevon, P., Mannila, H., Herr, M., Kere, J.: Data mining applied to linkage disequilibrium mapping. American Journal of Human Genetics **67** (2000) 133–145

[10] Laitinen, T., Polvi, A., Rydman, P., Vendelin, J., Pulkkinen, V., Salmikangas, P., Rehn, M., Pirskanen, A., Rautanen, A., Zucchelli, M., Gullsten, H., Leino, M., Alenius, H.: Characterization of a common susceptibility locus for asthma-related traits. Science (2004) 300–304

[11] Kirkwood, B.R.: Chapter 13. In: Essentials of Medical Statistics, ISBN 0-632-01052-5. Blackwell Scientific Publications (1988) 87–90

[12] Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman and Company, New York (1979)

[13] Clayton, D.: SNPHAP : A program for estimating frequencies of large haplotypes of SNPs. `http://www-gene.cimr.cam.ac.uk/clayton/software/snphap.txt` (2002)

[14] Stephens, M., Smith, N.J., Donnelly, P.: A new statistical method for haplotype reconstruction from population data. Am. J. Hum. Genet. (2001) 978–989

[15] Niu, T., Qin, Z.S., X, X., Liu, J.S.: Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. Am. J. Hum. Genet. (2002) 157–169