EXACT AND HEURISTIC APPROACHES FOR IDENTIFYING DISEASE-ASSOCIATED SNP MOTIFS

GAOFENG HUANG and PETER JEAVONS
Computing Laboratory, University of Oxford
{Gaofeng.Huang, Peter.Jeavons}@comlab.ox.ac.uk

DOMINIC KWIATKOWSKI
Wellcome Trust Centre for Human Genetics, University of Oxford
Dominic.Kwiatkowski@paediatrics.oxford.ac.uk

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 are given to demonstrate that these approaches are sufficiently effective to support ongoing biological research.

1. Introduction

The DNA sequences of different individuals within the same species are highly conserved but not exactly the same. 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. A general research question within the field of human genetics is to ask whether a disease of interest is related to the occurrence of (unknown but) particular SNPs within the human genome. This question is hard to answer in most cases due to the fact that the frequency of SNPs occurring in the human genome is estimated to be 1 SNP per 300 bases, which means that there are about 10 million common SNPs to be investigated. It is currently technically and economically impossible to screen the whole genome of all patients.

Fortunately, the phenomenon of Linkage Disequilibrium (LD) makes some progress possible. Two SNPs are said to be in high Linkage Disequilibrium if their respective alleles are not randomly associated with each other. This allows us to investigate only some SNPs and infer others. To calculate the LD value between two SNP loci, several quantitative LD measures have been proposed, including D' [4], r^2 [3] and λ [9]. 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.

Within this framework there are currently two different typical scenarios for finding the positions of disease-associated SNPs depending on how much *a priori* knowledge we have about these positions. In the first scenario, where we know nothing about where these 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 is most likely to contain disease-associated SNPs. This problem has been the subject of extensive research by statisticians [5,6,7,8]. Some research has also been done by computer scientists, including HPM (Haplotype Pattern Mining) [11], which has had some success. Many of these methods are statistically powerful; they can often locate candidate disease-associated SNPs with a resolution of about 50-100Kb (where a Kb is 1000 nucleotide bases).

In the second scenario, we assume that the disease-associated SNPs we wish to identify are already known to lie within some small region of the genome (50-100Kb). This is the scenario that our biological collaborators are currently investigating. 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 (e.g. lifestyle). 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. If we only examine each single SNP locus, we might not find any significant association between each individual SNP and the disease. However, a particular combination of several SNPs leads to high disease risk. One possible explanation is that this particular combination of several SNPs might be in high LD with another 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).

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. Our biological collaborators currently perform this task manually, and our aim is to develop efficient automated tools to do this job.

The rest of the paper is organized as follows. In Section 2, we formulate the problem as a combinatorial optimization problem. In Section 3 we show this problem to be $\mathcal{NP}$-hard, and develop both exact and heuristic approaches. Computational results are presented in Section 4 for both simulated data and real data. 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 this alphabet is usually adopted by biologists is that almost all (99%) of the SNPs

in the human genome are bi-alleles, for example `A/T`. The standard practice is to 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. The vector in row i of matrix $M^{(x)}$, i.e. the data for individual i, will be denoted $M^{(x)}_i$.

A SNP *motif* is an expression of the form "`--11---2-`", where "-" means "don't care". A convenient sparse representation for such a motif is to use two vectors, a position vector $P = (p_1, p_2, ..., p_k)$, where $1 \leq p_1 < p_2 < ... < p_k \leq m$ and a data vector $D = (d_1, d_2, ..., d_k)$, where each $d_i \in \{1, 2\}$. These 2 vectors specify the motif by requiring that the allele in each position p_i should be d_i; for example, the motif "`--11---2-`" is represented as $P = (3, 4, 8), D = (1, 1, 2)$.

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

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

Using this function, the number of cases (a) and controls (b) that match a motif (P, D) is given by the formula $a = \sum_{i=1}^{n_p} Match((P, D), M^{(p)}_i)$ and $b = \sum_{i=1}^{n_h} Match((P, D), M^{(h)}_i)$ respectively.

	#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 1. 2×2 contingency table

To measure the significance of a motif, a standard method is to put these numbers into a 2×2 contingency table (see Figure 1) and perform a standard chi-squared test. A convenient formula for calculating the value of the chi-squared statistic is:

$$\chi^2(a, b) = \frac{n(ad - bc)^2}{efn_p n_h} = \frac{n(an_h - bn_p)^2}{(a + b)(n - a - b)n_p n_h}. \quad (1)$$

Our search problem can now be formulated as finding a motif (P^*, D^*) which yields the maximal possible value for χ^2, given the data matrices $M^{(p)}_{n_p \times m}$ and $M^{(h)}_{n_h \times m}$. This problem will be called the SNP motif identification problem.

We believe that this formulation of the problem captures essential aspects of current biological research in this area, including the fact that SNP motifs may be only weakly associated with the occurrence of disease, and yet may be biologically significant. By seeking to maximise a statistical measure of association our formulation is also able to cope with noisy data and missing values.

3. Methods

3.1. $\mathcal{NP}$-hardness

In this section, we will show that the SNP motif identification problem 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 [2], 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 construct a corresponding instance of the SNP motif identification problem by the following procedure:
1. Let the m boolean variables of the MAX-SAT instance correspond to m SNPs.
2. Construct the matrix $M^{(h)}$ by transforming each clause C_i to the vector $M_i^{(h)}$:

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

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

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 (Equation (1)) becomes $\chi^2(1, b) = \dfrac{(n_h - b)}{(1 + b)n_h}$, 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 $x_j^* = true$ if $d_j^* = 2$, and $x_j^* = false$ if $d_j^* = 1$.

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

1. $M_{i,j}^{(h)} = 1$ **and** $d_j^* = 2$**.** In this case $M_{i,j}^{(h)} = 1$ means that clause C_i contains x_j while $d_j^* = 2$ means $x_j^* = true$, so clause C_i is satisfied.

2. $M_{i,j}^{(h)} = 2$ **and** $d_j^* = 1$**.** In this case $M_{i,j}^{(h)} = 2$ 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 of finding a motif (P^*, D^*) with the maximal value for χ^2. Since MAX-SAT is known to be $\mathcal{NP}$-hard, it follows that the SNP motif identification problem is also $\mathcal{NP}$-hard.

3.2. *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 hence 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.

3.2.1. *Search Tree 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.

Each 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 the values for each p_i and d_i can be retrieved by tracing the path from the root node to the given node. To speed up the calculation of χ^2 objective function and allow the search tree to be pruned, we will associate each node with a triple (j, S^+, S^-), where S^+ is the set of lines in $M^{(p)}$ which match motif (P, D), i.e. $S^+ = \{i \mid Match((P, D), M_i^{(p)}) = 1\}$ and $S^- = \{i \mid Match((P, D), M_i^{(h)}) = 1\}$.

The root node is associated with the triple $(j = 0, S^+ = \{1, 2, ..., n_p\}, S^- = \{1, 2, ..., n_h\})$. Given a node with associated triple (j, S^+, S^-), its child node along the branch (p_{j+1}, d_{j+1}), has associated triple $(j + 1, S'^+, S'^-)$, where

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

$$S'^- = \{i \mid i \in S^- \text{ and } M^{(h)}_{i,p_{j+1}} = d_{j+1}\}. \tag{3}$$

Hence, generating a new node with its associated triple, can be done in $O(n_p + n_h)$ time. Since $a = |S'^+|$ and $b = |S'^-|$, we can calculate the χ^2 objective function for that node in $O(n_p + n_h)$ as well. Hence by storing the information in the triples at each node we have reduced the time complexity of the calculations at each node from $O((n_p + n_h)m)$ to $O(n_p + n_h)$.

3.2.2. *Branching Strategy and Pruning Rules.*

We use a Depth-First-Search strategy to minimise the storage requirements. Two pruning rules are applied to reduce the search space:

1. Absorption Rule : When we generate a child node of a node with triple (j, S^+, S^-) following branch $(p_{j+1} = p^*, d_{j+1} = d^*)$, as shown in Figure 2, if the child node has the associated triple $(j+1, S'^+, S'^-)$, where $S'^+ = S^+$ and $S'^- = S^-$, then this child node can be "absorbed", and therefore the whole subtree T' rooted at the child node can be pruned, as we will now explain.

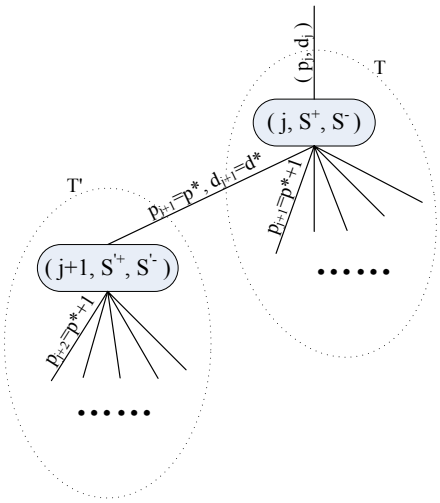

Figure 2. Absorption Rule

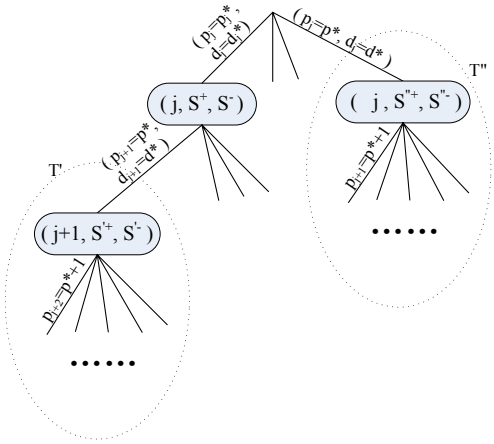

Figure 3. Sibling Rule

Recall the calculation of S'^+ and S'^- in Equations (2) and (3). If $S'^+ = S^+$ and $S'^- = S^-$, then we have that $\forall i \in S^+, M^{(p)}_{i,p^*} = d^*$ and $\forall i \in S^-, M^{(h)}_{i,p^*} = d^*$. In other words, for both cases and controls, all the remaining individuals have the same allele d^* in the SNP locus p^*, so adding position p^* to the motif does not provide any more information. This implies that subtree T' and subtree T in Figure 2 are identical, so we can prune subtree T' and explore subtree T only.

2. Sibling Rule : Suppose that we generate a child node of a node with associated triple (j, S^+, S^-), following branch $(p_{j+1} = p^*, d_{j+1} = d^*)$, and assume that the original node has a sibling node which follows branch $(p_j = p^*, d_j = d^*)$ from its parent, as shown in Figure 3. If the child node has associated triple (j, S'^+, S'^-), and the sibling node has associated triple (j, S''^+, S''^-), where $S''^+ = S'^+$ and $S''^- = S'^-$, then the child node can be pruned, as we will now explain.

It follows from their relative positions in the search tree that the child node and the sibling node 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 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 in the subtree T', matching the same cases and controls, which is 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.

3.3. *Heuristic Approach*

Our heuristic approach includes two stages: a search stage and a refinement stage.

The search stage is adapted from the exact search algorithm described in the previous section by extending the absorption rule. In Figure 2, a child node with triple $(j+1, S'^+, S'^-)$ is absorbed by its parent node with triple (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 with triple $(j+1, S'^+, S'^-)$ will be absorbed by its parent node with triple (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: $|S'^+| \geq 0.95\ |S^+|$ and $|S'^-| \geq 0.95\ |S^-|$.

Using this rule, when we generate the children of a node we prune those branches which do not provide "enough" new information. This pruning is very effective because it makes use of the statistical relationship between different SNP loci, i.e. Linkage Disequilibrium.

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

4. Results

We have 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. We set the time limit of running each single testcase to be 5 minutes.

4.1. *Simulated Data with Realistic Linkage Disequilibrium*

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. To provide a suitable test set for our computational tools we therefore had to develop a novel way to generate simulated data with realistic Linkage Disequilibrium between SNP at different loci. The procedure is briefly described as follows:

Step 1: Obtain raw data from the HapMap Project[10]. The HapMap raw data (release 14) contains 60 unrelated Caucasian (CEU) individuals, 60 Yoruba individuals, 45 unrelated individuals from Tokyo (Japan) and 45 unrelated individuals from Beijing (China). We randomly select $m+1$ consecutive SNP loci.

Step 2: Estimate haplotype frequency using a popular program SNPHAP [1]

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. We first randomly assume one SNP locus $p*$ among the $m+1$ loci to be the disease causative SNP. Then we assign a disease risk $r_1 = 0.001$ and $r_2 = 0.01$ respectively to allele "1" and "2" in SNP locus $p*$, and randomly simulate whether each individual gets the disease or not according to that risk. Finally we remove the column of the disease causative SNP, locus $p*$, and only use the remaining m SNP loci.

Step 5: Simulate a clinical scenario. A classical case-control study samples the same number of cases and controls. Here we consider all individuals in our simulated population sample who get the disease to be cases. Then we randomly sample the same number of healthy individuals from the whole population to be controls.

4.1.1. *Exact Search vs. Heuristic*

We first carried out experiments with the number of SNPs, m, set to 7 different values ranging from 20 to 50. For each value of m, we did 100 simulations to generate 100 testcases from 10 different genomic regions. Both exact and heuristic approaches were then tested on these $7 \times 100 = 700$ testcases. The results are reported in Table 1. Each row presents the average result over the 100 testcases with the same m. The first column is the average number of nodes that the exact search algorithm explores. The second column is the average CPU time that the exact search algorithm takes. The third column is the number of testcases solved within 5 minutes. Similarly, the fourth, fifth and sixth columns are for the heuristic approach. The last column shows for how many testcases the solution that the heuristic returns is optimal, i.e. the same solution as the exact algorithm returns.

Table 1. Results for Simulated Data ($7 \times 100 = 700$ testcases in total)

	Exact Search			Heuristic			
m	average nodes	average time(sec)	#solved	average nodes	average time(sec)	#solved	#optimal
20	16988.0	0.62	100	7110.3	0.24	100	100
25	47166.3	3.34	100	16533.1	0.88	100	100
30	161205.8	21.54	97	41637.1	4.13	100	$\geq$97
35	-	-	85	77899.7	11.23	100	$\geq$85
40	-	-	-	124126.4	29.16	100	-
45	-	-	-	187257.0	67.38	97	-
50	-	-	-	-	-	83	-

These results show that our exact algorithm using a branch-and-prune strategy is considerably more efficient than a brute-force approach: the number of nodes explored is much smaller than 3^m, as shown in Table 1, because a large number are pruned. The results also indicate that our heuristic method can provide high quality motifs using less time. In fact, for all of the 382 testcases that we can compare, the motifs that the heuristic method provides are always optimal.

4.1.2. *Are the Motifs Found Real Signals?*

More specifically, what if the disease is not associated with any SNP? Under our simulation framework, We simulated this "no association" scenario by setting disease risk $r_1 = r_2 = 0.005$. We did 100 simulations to generate 100 testcases with $m = 40$ SNPs to test our heuristic approach. We compare the results with the 100 testcases ($m = 40$) from previous experiments. Figure 4 shows the histogram

for both scenarios (wider bars for the association scenario and narrower bars for non-association). The x-axis represents the χ^2 value of the motif we found, and the y-axis represents the number of testcases.

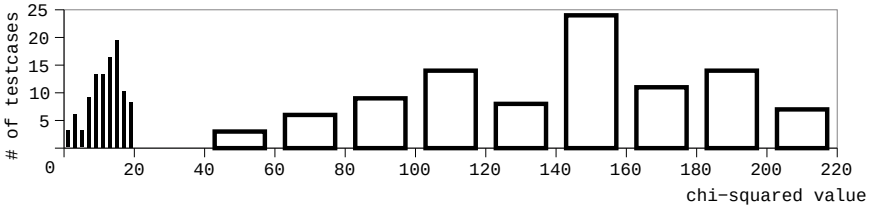

Figure 4. Histogram (Non-association vs. association)

Figure 4 shows that if the disease is not associated with any SNP, then the χ^2 value of the motif that our algorithm returns is very low (less than 20). On the other hand, if the disease has a causative SNP, even though this causative SNP is not directly observed, our algorithm finds a motif with much higher χ^2 value (above 40). This indicates that if our algorithm find motifs with high χ^2 value, then the motifs are very likely to be real biological signals.

4.2. *Clinical Data*

Finally, we obtained a real clinical testcase from the biological laboratory. (Due to privacy policy and copyright restrictions, we cannot give full details of the clinical background to this dataset.) This 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 was obtained in the biological laboratory using this manual process.

As shown in Table 2, the motifs obtained by our methods are very similar to those obtained by the current labour-intensive manual process. For this particular dataset the resulting χ^2 value (22.22) is not high enough to conclude with confidence that the SNP motif is significantly associated with the disease. However, we have successfully automated the process of finding the best possible motif, and this approach can now be used to support ongoing biological research.

Table 2. Real Clinical Data

	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

5. 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 and tested on both simulated and real data. The results demonstrate that these computational approaches can support ongoing biological research.

For simplicity of problem description in this paper we haven't made the distinction between "haplotype" data and "genotype" data. In fact, our approach deals with SNP haplotype data. To infer haplotype data from genotype data is another very active research topic. In this paper, we have used the SNPHAP program [1] as a preprocessor to obtain estimated haplotype data. Our plan for the next stage of this research is to develop algorithms which can deal directly with the unphased genotype SNP data to identify significant SNP motifs.

References

1. D. Clayton. SNPHAP : A program for estimating frequencies of large haplotypes of SNPs. *http://www-gene.cimr.cam.ac.uk/clayton/software/snphap.txt*, 2002.
2. M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman and Company, New York, 1979.
3. W. G. Hill and A. Robertson. Linkage disequilibrium in finite populations. *Theoret. Appl. Genet.*, 38:226–231, 1968.
4. R. C. Lewontin. The interaction of selection and linkage. I. general considerations; heterotic models. *Genetics*, 49:49–67, 1964.
5. J S Liu, C Sabatti, J Teng, B J B Keats, and N Risch. Bayesian analysis of haplotypes for linkage disequilibrium mapping. *Genome Research*, 11:1716–1724, 2001.
6. X. Lu, T. Niu, and Jun S. Liu. Haplotype information and linkage disequilibrium mapping for single nucleotide polymorphisms. *Genome Research*, 13:2112–2117, 2003.
7. M. S. McPeek and A. Strahs. Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping. *Am. J. Hum. Genet.*, 65:858–875, 1999.
8. A P Morris, J C Whittaker, and D J Balding. Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *Am. J. Hum. Genet.*, 70:686–707, 2002.
9. J. D. Terwilliger. A powerful likelihood method for the analysis of linkage disequilibrium between trait loci and one or more polymorphic marker loci. *Am. J. Human Genet.*, 56:777–787, 1995.
10. The International HapMap Consortium. The International HapMap Project. *Nature*, 426:789–796, December 2003.
11. Hannu T. T. Toivonen, P. Onkamo, K. Vasko, V. Ollikainen, P. Sevon, H. Mannila, Mathias Herr, and Juha Kere. Data mining applied to linkage disequilibrium mapping. *American Journal of Human Genetics*, 67:133–145, 2000.