Haplotype Block Partitioning and TagSNP Selection on Human Chromosome 21

Wen-Pei Chen Tso-Ching Lee* Yaw-Ling Lin†

Dept. of Comput. Sci. and Info. Management,
College of Computing and Informatics, Providence University,
200 Chung Chi Road, Shalu, Taichung, Taiwan 433.
E-mail: g9471023@pu.edu.tw, yllin@pu.edu.tw

Abstract

A Single Nucleotide Polymorphism or SNP is a DNA sequence variation occurring when a single nucleotide in the genome differs between members of species. Recent research reveals that SNPs within certain haplotype blocks induce only a few distinct common haplotypes in the majority of the population. The existence of haplotype block structures has serious implications for association-based methods in mapping of disease genes. In this paper, we propose and implement several efficient algorithms for identifying haplotype blocks in the genome. For dealing with missing data, which often appears in real biomedical data, we develop methods that assign missing data to specific alleles such that the resulting diversity is minimized. Our developed system is used for analyzing chromosome 21 haplotype data provided by Patil et al. [11]. In contrast to previous partition methods, the haplotype blocks and tagSNPs identified by our methods are longer and fewer, yet still retaining the desired expressing capability.

Keywords: Diversity, dynamic programming, SNP, haplotype block, tagSNP, missing data.

1 Introduction

SNPs make up 90% of all human genetic variations, and SNPs with a minor allele frequency of $\geq 1\%$ occur every 100 to 300 bases along the human genome. Global pattern of human DNA sequence variation (haplotypes) defined by common single nucleotide polymorphisms (SNPs) have important implications for identifying disease association and human traits. Recent studies have shown that the chromosome recombination only takes places at some narrow hotspots. We use some characteristics of recombination on the analysis of haplotype strings and observe several situations of haplotype patterns to identify some ranges of chromosome with few or even no recombination event occurred. Those ranges are called *haplotype blocks*. It means every haplotype pattern is inherited to descendant completely if the number of haplotype patterns in a haplotype block is few. Each haplotype block, in which the genome is largely made up of regions of low diversity, can be characterized by a small number of SNPs, which are refereed to as *tagSNPs* [7]. This characteristic is very important and useful for medicine or therapy.

Studying on SNP and haplotype blocks not only decrease the cost for detecting inherited diseases but also has many contributions for classifying the race of human and researching on species evolution. Our ultimate goal is to select haplotype block designations that best capture the structure within the data. Unfortunately, a consensus definition for haplotype blocks based on the LD (linkage disequilibrium, also termed allelic associations) structure has not been established thus far. However, a range of operational definitions has been used to identify haplotype-block structures, including LD-based [4, 12], recombination-based [6, 13], information-complexity-based [1, 8, 5] and diversity-based [3, 11, 15] methods. For a diversity-based test, methods can be classified into two categories: those that divide strings of SNPs into blocks on the basis of the decay of LD across block boundaries and those that delineate blocks on the basis of some haplotype-diversity measure within the blocks. Patil et al. [11] defined a haplotype block as a region in which a fraction of percent or more of all the observed haplotypes are represented at least n times or at a given threshold in the sample. They applied the optimization criteria outlined by Zhang et al. [14, 15] and describe a general algorithm that defines block boundaries in a way that minimizes the number of SNPs that are required to identify all the haplotype in a region. Patil et al. have identified a total of 4,563 tagSNPs and a total of 4,135 blocks to define the haplotype structure of human chromosome 21. In each block

*Address: Taichung Veterans General Hospital. 160, Sec. 3, Taichung Kang Road, Taichung, Taiwan 407. E-mail: tclee@vghtc.gov.tw

†Corresponding author. This work is partially supported by grants from the Taichung Veterans General Hospital and Providence Univ. (TCVGH-PU-958101) and by the National Science Council (NSC-95-2221-E-126-007), Taiwan, Republic of China.

they required at least 80% of haplotype must be represented more than once in the block.

In this paper, we propose diversity functions that can be adapted to measure haplotype block quality. We implement programs according to our dynamic programming algorithms to partition haplotypes into blocks. Our algorithms find segmentations consisting of k blocks such that the total length is maximized. Based on the same criteria adopted by Patil et al., at least 80% of haplotypes appeared more than once in the block, we identify a total of 1,707 blocks, which can be tagged by 4,588 tagSNPs. The number of blocks we identified is 58.7% less than those identified by Patil et al. Furthermore, although the number of tagSNPs discovered by our method is almost the same as theirs, note that, for example, 20% of our tagSNPs (about 900 tagSNPs) is sufficient to tag 50% region of chromosome 21 (more than 16 Mb). Many similar interesting observations are addressed in Section 3.

2 Methods

A *SNP* (Single Nucleotide Polymorphisms) is defined as a position in a chromosome where each one of two (or more) specific nucleotides is observed in at least 10% of the population [11]. The nucleotides involved in a SNP are called *alleles*. A SNP is said *biallelic* if it has only two different alleles. Almost all SNP are biallelic and we will consider exclusively biallelic SNP in this paper.

2.1 Diversity Functions

The result of block partition and the meaning of each haplotype block may be different by using different measuring formula. Here we provide a measuring functions for analysis. In simplicity, we convert haplotype samples into haplotype matrixes by assigned major alleles to 0 and minor alleles to 1. Given an $m \times n$ haplotype matrix A, a block $A(i, j)$ (i, j are the block boundaries) of matrix A is viewed as m haplotype strings; they are partitioned into k groups by merging identical haplotype strings into the same group. The probability p_i of each haplotype pattern s_i, is defined accordingly such that $\sum p_i = 1$. Li [9] proposes a diversity formula defined by

$$\delta_D(S) = 1 - \sum_{p_i \in S} p_i^2. \quad (1)$$

Note that $\delta_D(S)$ is the probability that two haplotype blocks chosen at random from S are different from each other.

We can use the formula to calculate the degree of difference within haplotype strings. Diversity measurement usually reflects the activity of recombination events occurred during the evolutionary process. Generally, haplotype blocks with low diversity indicates conserved regions of genome.

2.2 Common Haplotypes

Two haplotypes are said to be *compatible* if the alleles are identical at all loci for which there are no missing data; otherwise the two haplotypes are said to be *incompatible.* As in Patil et al., we define the ambiguous haplotypes as those haplotypes compatible with at least two haplotypes that are themselves incompatible. It should be noted that when there are no missing data, all of the haplotypes are unambiguous. We define the *common haplotypes* as those haplotypes that are represented more than once in a block. The haplotypes are called *singleton* if they are not compatible with any others.

We are mainly interested in the common haplotypes. Therefore we require that, in the final block partition, a significant fraction of the haplotypes in each block are common haplotypes. Patil et al. require that at least α =70%, 80%, and 90%, respectively, of the unambiguous haplotypes appear more than once. The α is also referred to as the *coverage* of common haplotypes in a block. Ambiguous haplotypes are not included in calculating percent coverage. The coverage of block A can be mathematically formulated as a form of diversity:

$$\delta_S(A) = 1 - \frac{C}{U} = \frac{S}{U}. \quad (2)$$

Here U denotes the number of unambiguous haplotypes, C denotes the number of common haplotypes, and S denotes the number of singleton haplotypes. In other words, Patil et al. [11] require that at most $\delta_S(A) \leq$ 30%, 20%, and 10%. Note that the coverage will not decrease as the length of haplotypes increase.

2.3 Haplotype Blocks Partitioning

When we select haplotype blocks base on the diversity value that calculated by diversity function 1, the diversity value of a candidate block must smaller than the diversity limit D. Denote the *farthest site* $j = L[i]$ for each site i as the largest site j with the diversity $\delta(i, j) \leq D$. We show in [10] that all farthest sites can be determined in time complexity $O(n)$ by using the characteristic of block diversity. Note that δ_D-function is a monotonic non-decreasing function from $[1..n; 1..n]$ to the unit real interval $[0, 1]$; that is, $0 \leq \delta(j', k') \leq \delta(j, k) \leq 1$ whenever $[j', k'] \in [j, k]$.

Given a haplotype matrix A and a diversity upper limit D, let $S = \{B_1, B_2, \ldots, B_k\}$ be a segmentation of A with $\delta(B) \leq D$ for each $B \in S$. The *length* of S is the total length of all block in S; i.e., $\ell(S) = |B_1| + |B_2| + \cdots + |B_k|$. Our objective is to find a segmentation consists of k feasible blocks such that the total length $\ell(S)$ is maximized. Given A and D, first we consider the most general form of the problem and define the block length evaluation function

$$f(k, i, j) = \max\{\, \ell(S) \mid S \text{ a feasible segmentation of}$$

```
LowDiv(R)
Input: Haplotype matrix A with missing data.
Output: Haplotype matrix A with low diversity by assigning missing data to 0 or 1.
1 Sort k groups without missing data in S = ⟨s_1, s_2, ..., s_k⟩.
    ▷  s_i's are listed in decreasing order of the number of haplotype strings.
2 Sort all rows with missing data in T = ⟨t_1, t_2, ..., t_|T|⟩.
    ▷  t_i's are listed in increasing order of the number of missing data.
3 for i ← 1 to |T| do        ▷ visit all strings in T.
4     for j ← 1 to k do      ▷ visit all groups in S.
5         if Compatible(t_i, s_j) then
6             t_i ← Consolidate(t_i, s_j)
7             |s_j| ← |s_j| + 1
8             remove t_i from T and leave this for loop
9     if ∀s_j ∈ S is not compatible with t_i then
10        {s_{k+1}} ⋃ t_i          ▷ add t_i into set S be a new group.
11        |s_{k+1}| ← 1
```

Figure 1: Assigning missed data with values to obtain matrix of low diversity.

$$A(i,j) \text{ with } k \text{ blocks}\}$$

Note that the k-LONGEST-BLOCKS [10] asks to find the value $f(k,1,n)$. After the *left farthest sites*, $L[j]$'s, are calculated, the answer can be found in $O(nk)$ time after the preprocessing. It can be verified that

$$\begin{aligned} f(k,1,j) &= \max\{f(k,1,j-1), \\ &\quad f(k-1,1,L[j]-1)+j-L[j]+1\} \end{aligned}$$

That is, the k-th block of the maximal segment S in $[1,j]$ either does not include site j; otherwise, the block $[L[j],j]$ must be the last block of S. Note that $f(k,1,j)$ can be determined in $O(1)$ time when $f(k-1,1,\cdot)$'s and $f(k,1,1..(j-1))$'s are ready. It follows that $f(k,1,\cdot)$'s can be calculated from $f(k-1,1,\cdot)$'s, totally in $O(n)$ time. Thus a computation ordering from $f(1,1,\cdot)$'s, $f(2,1,\cdot)$'s, ..., to $f(k,1,\cdot)$'s leads to the result of $O(nk)$ time.

2.4 TagSNPs Selection

For each block, we want to minimize the number of SNPs that uniquely distinguish at least 80% of the unambiguous haplotypes in the block. Those SNPs can be thought of as a signature of the haplotype block partition. They are referred to as *tagSNPs* that are able to capture most of the haplotype diversity, and therefore, could potentially capture most of the information for association between a trait and the marker loci [2].

Our strategy for selecting the tagSNPs in haplotype blocks is as the following. First, the common haplotypes are grouped into k distinct patterns in each block. After the missing data are assigned, as explained in the next subsection, we decide the least number of groups needed such that haplotypes in these groups contain at least 80% of the unambiguous haplotypes in the block. Finally, we select a loci set which consist of minimum number of SNPs on the haplotypes such that each pattern can be uniquely distinguish; exhaustive searching method can be used very efficiently since the number of tagSNPs needed for each block is usually modest in the situation. The exhaustive searching algorithm enumerates next r-combination in lexicographic order to generate the next candidate tagSNP loci set until each pattern can be uniquely distinguish.

2.5 Dealing with Missing Data

In real biomedical data, some SNPs may be missing, and we may fail to distinguish two distinct haplotypes due to the ambiguity caused by missing data. The haplotype matrix A is generalized to an $m \times n$ matrix of m observations over n markers (sites) such that $A_{ij} \in \{0,1,3\}$; $A_{ij} = 3$ when the j-th SNP site of observation i is a missing data. One way to deal with the missing data when trying to determine the diversity of submatrix $A(j,k)$ is to set missing values of A_{ij}'s to either 0 or 1, such that the resulting diversity of $A(j,k)$ is minimized. However, as we have shown in [10] that the *minimum diversity problem* is NP-complete.

Given a haplotype matrix A with missing data, we say two rows i,j of A are *different* if there exists a column k such that $\{A_{ik}, A_{jk}\} = \{0,1\}$; in other words, it is impossible to assign missing data of A to make the two rows identical. Two rows are *compatible* if they are not different. Our method for dealing with missing data is as the following. The main idea is to assign the missing data $A_{ij} = 3$ to 0 or 1 in interval $[x,y]$ such that $\delta(A(x,y))$ is minimized. By reducing the patterns of haplotype as few as possible, the assigned haplotype patterns would lead to small diversity score.

Our method consists of three phases: partition phase, search phase, and assignment phase. First, we partition $A(x,y)$ into two sets. The set of strings with miss-

ing data is called T, and those strings without missing data are called S. Strings in S can be grouped into k patterns (distinct strings), $\langle s_1, \ldots, s_k \rangle$; they are arranged in order of decreasing number of haplotype strings, $\langle p_1, \ldots, p_k \rangle$ such that $p_1 \geq p_2 \geq \ldots \geq p_k$. The strings in T are ordered in increasing the number of missing site, $\langle t_1, \ldots, t_n \rangle$.

In search phase, we try to find the compatible strings from set S and T. For each $t_i \in T$, $i = 1, \ldots, n$, we find a string $s_j \in S$ such that t_i and s_j are compatible and p_j is maximized before invoking the assignment phase. If there is not any string compatible with t_i, t_i is added into set S and becomes a new group s_{k+1}. In third phase, we assign the missing data of string t_i to the corresponding value in string s_j. Repeat the second and third phases above until all string $t_i \in T$ are be classified into S. These steps aim to increase fewest haplotype patterns and to assign more counts into groups with larger counts. The heuristic algorithm is shown in Figure 1. Note that missing data may still exist in the haplotype matrix after the missing data process; e.g., some strings are never compatible with the others or the values of one specific column are all missing data. It's worthy noting that the situations do not influence the computation of diversity. Because two strings are always different no matter how we assign the missing data and the values of one specific column with all missing data are the same.

3 Experimental Results

We apply our dynamic programming algorithm to the haplotype data for chromosome 21 provided by Patil et al. [11]. The data contain 20 haplotype samples and each contains 24,047 SNPs spanning 32.4 Mb of chromosome 21. The minor allele frequency at each marker locus is at least 10%. Using our algorithm with the same criteria as in Patil et al. with coverage = 80%, a total of 4,588 tagSNPs and a total of 1,707 haplotype patterns blocks are identified. In contrast, Patil et al. identified a total of 4,563 tagSNPs and a total of 4,135 blocks. Our dynamic programming algorithm reduces the number of blocks by 58.71%. The properties of blocks are showed in Tables 1. Our analysis system discovers a total of 564 blocks containing more than 15 SNPs per block. The blocks with more than 15 SNPs account for 33% of all of blocks. The average number of SNPs for all of the blocks is 14.09. The largest block contains 135 common SNPs, which is longer than the largest block (containing 114 SNPs) identified by Patil et al..

Figure 2-a shows the relation between blocks number and the percentage of genome region (common SNPs) covered by the total blocks length. Note that only a few blocks are needed to cover a wide range of genome region. For example, 503 blocks (about 29% of blocks in figure 2-b) suffices to cover 70% of genome region (about 16,830 common SNPs); the average length of each block covered 33 common SNPs. Figure 3-a shows us how many tagSNPs are required when blocks cover certain percent of genome region. According to experimental results, when blocks cover 70 percent of genome region, we just required 1,633 tagSNPs (about 35.6% of tagSNPs in Figure 3-b) to capture the majority of information about haplotypes. This also indicates that our method identifies only a few tagSNPs to capture the most of genome region information. Table 2 shows some examples of this case. Figure 4-a shows the blocks cover percentage in the genome genion, in contrast to percentage of common SNPs coverage by each tagSNP on average. Note that as the total blocks coverage in the genome region increase, fewer common SNPs are covered by each tagSNP on average. Figure 4-b shows SNP numbers covered per tagSNP for each 10 percent of genome region covered. It is interesting to note that our method discovers the marginal utility of tagSNPs decreases as the genome region covered increases. Figure 4-c shows that the relationship between the percentage of genome region to number of tagSNPs needed to cover.

Furthermore, we examine the influence of common haplotype coverage, α, on the block patterns. The coverages with 70%, 80%, and 90% are examined. When the required coverage is 90%, the total number of blocks increases to 3,132. The total number of tagSNPs required to distinguish these blocks increases to 6,310. The length of the largest block decreases to 92 SNPs. When the coverage is decreased to 70%, the total number of blocks decreases to 1,136 with the largest block containing 183 common SNPs, and the total number of SNPs required to distinguish these blocks decreases to 3,925. Some of our primary results have been incorporated into our web-based system, and the system is freely accessible at `http://bioinfo.cs.pu.edu.tw/~hap/chr21.html`.

4 Conclusion

In this paper, we develop haplotype block-partition system according to our dynamic programming method where we require the total block length is maximized. By using appropriate diversity function, the block selection problem can be viewed as finding a segmentation of given haplotype matrix such that the diversities of chosen blocks satisfy certain value constraint. Compared with Patil et al.'s results, our method identifies longer blocks and the numbers of blocks is reduced by 57.7% for the haplotype data on chromosome 21. Our method discovers that only a few blocks is sufficient to cover a wide range of genome region, and it requires just a few tagSNPs to capture the most of genome region information.

Instead of genotyping all of the SNP markers on the chromosome, one may wish to use only the genotype information on the tagSNP. Only about 19.1% (4,588)

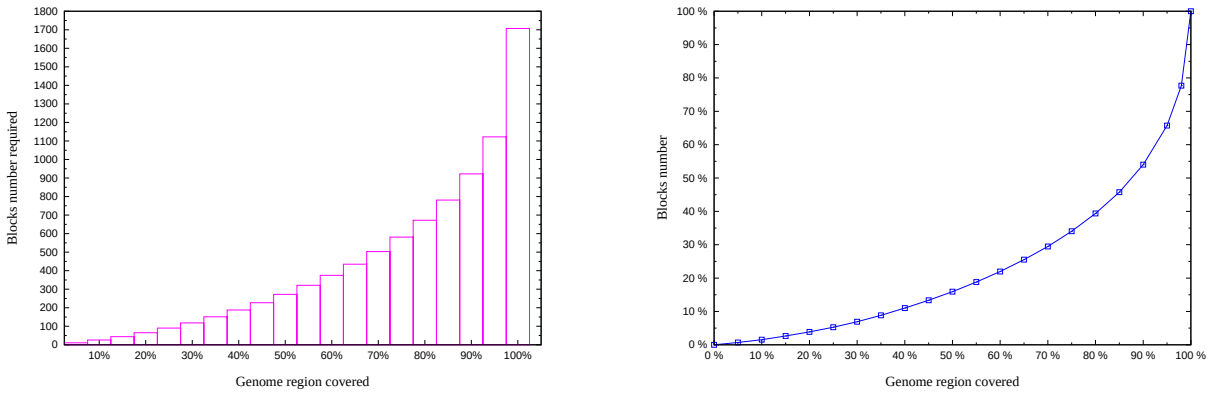

Figure 2: (a) The percentage of genome region covered by the blocks number, and (b) the percentage of blocks number on the chromosome 21.

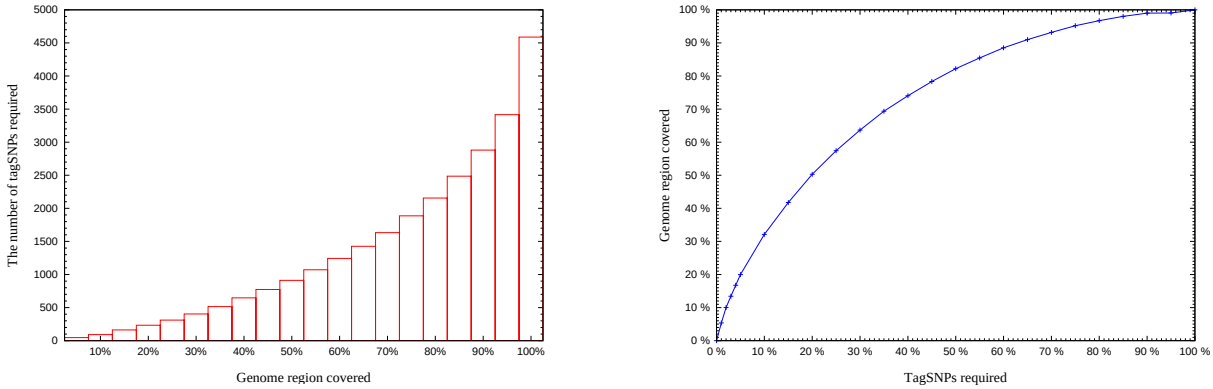

Figure 3: (a) The tagSNPs required versus the percentage of genome region covered. (b) The percentage of tagSNPs required versus the percentage of genome region covered.

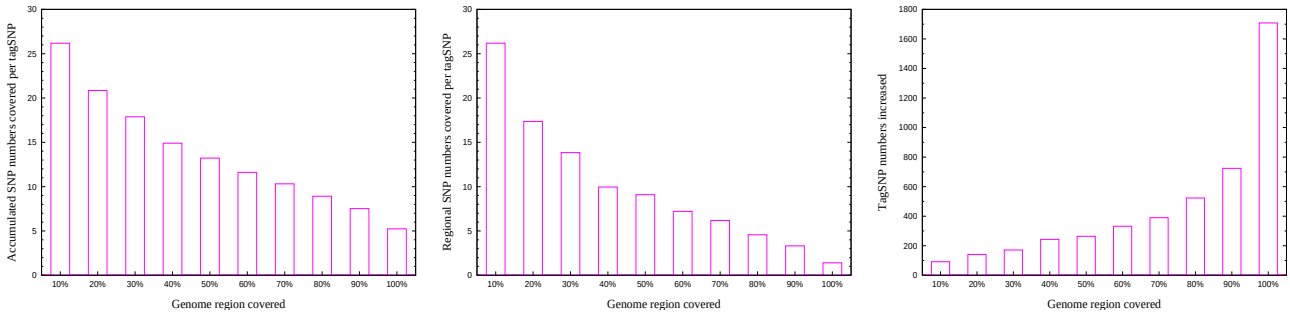

Figure 4: (a) The percentage of genome region covered versus the number of SNPs covered by each tagSNP on average. (b) SNP numbers covered per tagSNP for each 10 percent of genome region covered. (c) The number of tagSNPs increases whenever the genome region covered increases by 10 percent.

Common SNPs/block	No. of block	Length	Avg. length	All blocks(%)	Common SNPs(%)
< 15	1143	8029	7.02	66.96	33.39
15 to 30	412	8603	20.88	24.14	35.78
> 30	152	7415	48.78	8.90	30.83
Total	1707	24047	14.09	100.00	100.00
Max. Block	135				

Table 1: The properties of haplotype blocks defined by the dynamic programming algorithm with 80% coverage.

TagSNPs required (%)	1	2	3	4	5	10	15	20
Genome region covered (%)	5.36	10.02	13.43	16.75	19.98	32.13	41.78	50.29

Table 2: The relation between the percentage of tagSNPs required and genome region covered.

of all of the SNPs (24,047) can account for 80% of the common haplotypes in each block. Thus, studying the tagSNPs can dramatically reduce the time and effort for genotyping, without losing much haplotype information. In fact, the result of block partition and the meaning of each haplotype block may be different by using different measuring formula, so we propose a diversity function to measure the block quality. We also provide our algorithms for dealing with missing data whin haplotype matrix. Haplotype diversity is widely used in population genetics studies, for this reason we develop a web tool that can be applied to analyze and visualize the diversity of haplotypes.

References

[1] Eric C. Anderson and John Novembre. Finding haplotype block boundaries by using the minimum-description-length principle. *Am. J. of Human Genetics*, 73:336–354, 2003.

[2] JM Chapman, JD Cooper, JA Todd, and DG Clayton. Detecting disease associations due to linkage disequilibrium using haplotype tags: a class of tests and the determinants of statistical power. *Hum Hered.*, 56(1-3):18–31, Nov 2003.

[3] D. Clayton. Choosing a set of haplotype tagging SNPs from a larger set of diallelic loci. *Nature Genetics*, 29(2), 2001.

[4] S. B. Gabriel, S. F. Schaffner, H. Nguyen, et al. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225–2229, 2002.

[5] G. Greenspan and D. Geiger. Model-based inference of haplotype block variation. In *Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, 2003.

[6] R. R. Hudson and N. L. Kaplan. Statistical properties of the number of recombination events in the history of a sample of dna sequences. *Genetics*, 111:147–164, 1985.

[7] G. C. Johnson, L. Esposito, B. J. Barratt, et al. Haplotype tagging for the identification of common disease genes. *Nat Genet.*, 29(2):233 – 7, Oct 2001.

[8] M. Koivisto, M. Perola, R. Varilo, W. Hennah, J. Ekelund, M. Lukk, L. Peltonen, E. Ukkonen, and H. Mannila. An mdl method for finding haplotype blocks and for estimating the strength of haplotype block boundaries. In *8th Pacific Symposium on Biocomputing (PSB)*, pages 502–513, 2003.

[9] W.H. Li and D. Graur. *Fundamentals of Molecular Evolution*. Sinauer Associates, Inc, 1991.

[10] Yaw-Ling Lin and Wei-Shun Su. Identifying long haplotype blocks with low diversity. In *Proceedings of the 23rd Workshop on Combinatorial Mathematics and Computation Theory*, pages 151–159, Changhua,Taiwan, Apr 28-29, 2006.

[11] N. Patil, A. J. Berno, D. A. Hinds, et al. Blocks of limited haplotype diversity revealed by high resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[12] J.D. Wall and J.K Pritchard. Haplotype blocks and linkage disequilibrium in the human genome. *Nature Reviews Genetics*, 4(8):587–597, 2003.

[13] N. Wang, J.M. Akey, K. Zhang, R. Chakraborty, and L. Jin. Distribution of recombination crossovers and the origin of haplotype blocks: the interplay of population history, recombination, and mutation. *Am. J. Human Genetics*, 71:1227–1234, 2002.

[14] K. Zhang, M. Deng, T. Chen, M.S. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. In *The National Academy of Sciences*, volume 99, pages 7335–7339, 2002.

[15] K. Zhang, Z.S. Qin, J.S. Liu, T. Chen T, M.S. Waterman, and F. Sun. Haplotype block partitioning and tag snp selection using genotype data and their applications to association studies. *Genome Res.*, 14(5):908–916, 2004.