Haplotype Block Partition with Limited Resources and Applications to Human Chromosome 21 Haplotype Data

Kui Zhang, Fengzhu Sun, Michael S. Waterman, and Ting Chen

Molecular and Computational Biology Program, Department of Biological Sciences, University of Southern California, Los Angeles

Recent studies have shown that the human genome has a haplotype block structure such that it can be decomposed into large blocks with high linkage disequilibrium (LD) and relatively limited haplotype diversity, separated by short regions of low LD. One of the practical implications of this observation is that only a small fraction of all the single-nucleotide polymorphisms (SNPs) (referred as "tag SNPs") can be chosen for mapping genes responsible for human complex diseases, which can significantly reduce genotyping effort, without much loss of power. Algorithms have been developed to partition haplotypes into blocks with the minimum number of tag SNPs for an entire chromosome. In practice, investigators may have limited resources, and only a certain number of SNPs can be genotyped. In the present article, we first formulate this problem as finding a block partition with a fixed number of tag SNPs that can cover the maximal percentage of the whole genome, and we then develop two dynamic programming algorithms to solve this problem. The algorithms are sufficiently flexible to permit knowledge of functional polymorphisms to be considered. We apply the algorithms to a data set of SNPs on human chromosome 21, combining the information of coding and noncoding regions. We study the density of SNPs in intergenic regions, introns, and exons, and we find that the SNP density in intergenic regions is similar to that in introns and is higher than that in exons, results that are consistent with previous studies. We also calculate the distribution of block break points in intergenic regions, genes, exons, and coding regions and do not find any significant differences.

Introduction

The pattern of linkage disequilibrium (LD) plays a central role in genomewide association studies to identify genetic variation responsible for common human diseases. SNP markers are preferred over microsatellite markers for association studies because of their abundance along the human genome (SNPs with minor allele frequency >0.1 occur in ~1 of every 600 bp) (Wang et al. 1998), the low mutation rate, and accessibilities to high-throughput genotyping. However, genotyping a large number of individuals for every SNP is still too expensive to be practical when using current technology.

The number of SNPs required for genomewide association studies depends on the LD pattern. Recent studies (Daly et al. 2001; Johnson et al. 2001; Patil et al. 2001; Dawson et al. 2002; Gabriel et al. 2002) have shown that the human genome can be partitioned into discrete blocks of high LD separated by shorter regions of low LD, such that only a small fraction of characteristic ("tag") SNPs are sufficient to capture most of haplotype structure of the human genome in each block. The haplotype block structure with the corresponding tag SNPs can be extremely useful for association studies in which it is not necessary to genotype all SNPs. A recent simulation study (Zhang et al. 2002*a*) indicated that the genotyping effort could be significantly reduced without much loss of power for association studies.

In a large-scale study of chromosome 21, Patil et al. (2001) identified, by a rodent-human somatic cell hybrid technique, 20 haplotypes consisting of 24,047 SNPs (with at least 10% minor allele frequency) spanning >32.4 Mb. They developed a greedy algorithm to partition the haplotypes into 4,135 haplotype blocks with 4,563 tag SNPs on the basis of two criteria: (1) in each block, at least 80% of the observed haplotypes are represented more than once; and (2) the total number of tag SNPs for distinguishing at least 80% of haplotypes is as small as possible. For the same data, Zhang et al. (2002*b*) reduced the number of blocks and tag SNPs to 2,575 and 3,582, respectively, using a dynamic programming algorithm. Both studies tried to minimize the total number of tag SNPs for the entire chromosome. However, when resources are limited, investigators may not be able to genotype all the tag SNPs and, instead, must restrict the number of tag SNPs used in their studies. An objective of the present article is to prioritize SNPs and corresponding chromosomal regions for genotyping in association studies with limited resources.

Received February 26, 2003; accepted for publication April 10, 2003; electronically published June 10, 2003.

Address for correspondence and reprints: Dr. Ting Chen, Department of Biological Sciences, University of Southern California, 1042 W. 36th Place DRB-290, Los Angeles, CA 90089-1113. E-mail: tingchen@usc.edu

0002-9297/2003/7301-0007$15.00

We first give a mathematical formulation for this problem and then develop two dynamic programming algorithms for haplotype block partitioning to maximize the fraction of the genome covered by a fixed number of tag SNPs.

The goal of association studies is to identify genetic variation responsible for human complex diseases and traits. Thus, it is necessary to know the functional SNPs. In addition, to understand the biological implications of haplotype block structure, we must know whether the results, such as the SNPs at the starting points of long blocks, are associated with biological functions. To address these questions, we first apply our algorithms to a data set of SNPs on human chromosome 21 (Patil et al. 2001) to obtain the corresponding haplotype block partitions. We then search the dbSNP database and human genome resources in the National Center for Biotechnology Information (NCBI) database to identify SNPs located in regions of known biological functions, which are, specifically, genes, exons, coding regions, and nonsynonymous SNPs. Finally, we statistically characterize the relationship between the SNPs at the beginning of the haplotype blocks and the SNPs in these regions to assess the biological implications of the haplotype blocks.

Methods

We first formulate the problem of haplotype block partitioning with limited resources, and then we provide algorithmic solutions. We also calculate the distribution of SNPs along coding and noncoding regions.

Haplotype Block Partitioning with Limited Resources

Assume that we are given K haplotype samples consisting of consecutive SNPs: $s_1,s_2,\ldots,s_n$. For simplicity, the SNPs are referred as $1,2,\ldots,n$. Let $h_1,h_2,\ldots,h_K$ be the K haplotype samples. Each haplotype $h_k,k=1,2,\ldots,K$ can be represented as an n-dimensional vector with the ith component $h_k(i) = 0$, 1, or 2 being the allele of the kth haplotype at the ith SNP locus, where 0 indicates missing data, and 1 and 2 are the two alleles.

Here, we follow the definitions of ambiguous and unambiguous haplotypes and the haplotype blocks proposed by Patil et al. (2001) and used by Zhang et al. (2002*b*). To make the present article self-contained, we summarize the definitions of "ambiguous" and "unambiguous" haplotypes. Consider haplotypes defined by SNPs i and j. Two haplotypes, k and k', are compatible if the alleles for the two haplotypes are the same at the loci with no missing data, that is, $h_k(l) = h_{k'}(l)$ for any $l,i\leqslant l\leqslant j$, and $h_k(l)h_{k'}(l) \neq 0$. A haplotype in a block is ambiguous if it is compatible with two other haplotypes that are themselves incompatible. For example, consider three haplotypes $h_1 = (1,0,0,2)$, $h_2 = (1,1,2,0)$, and $h_3 = (1,1,1,2)$. Haplotype h_1 is compatible with haplotypes h_2 and h_3, but h_2 is not compatible with h_3, because they differ at the third locus. Thus, h_1 is an ambiguous haplotype, whereas h_2 and h_3 are unambiguous haplotypes. In the remainder of the present article, only unambiguous haplotypes will be included in the analysis. Compatible haplotypes will be treated as identical haplotypes.

A segment of consecutive SNPs can form a block if at least α percent of unambiguous haplotypes are represented more than once in the samples (Patil et al. 2001; Zhang et al. 2002*b*). The tag SNPs are selected on the basis of the measure of haplotype quality in each block. Different measures of block quality have been used, depending on the purpose of a study. For example, Patil et al. (2001) defined the tag SNPs as the minimum subset of SNPs that can distinguish at least α percent of the unambiguous haplotypes. Another measure is based on haplotype diversity (Johnson et al. 2001). We can choose tag SNPs that minimize the number of SNPs that can account for at least β percent of overall haplotype diversity. In the present article, we follow the definition of tag SNPs used by Patil et al. (2001).

Given α and the above-mentioned criterion for defining tag SNPs, Zhang et al. (2002*b*) developed a dynamic programming algorithm for haplotype block partitioning to find a partition with the minimum total number of tag SNPs. For a fixed number of SNPs to be genotyped, we consider haplotype block partitions with some SNPs being excluded. For a set of consecutive SNPs $(s_i,s_{i+1},\ldots,s_j)$, we define the following functions:

- $block(i,\ldots,j) = 1$ if at least $\alpha M(\alpha < 1)$ unambiguous haplotypes defined by the SNPs $s_i,s_{i+1},\ldots,s_j$ are represented more than once, where $M \leqslant K$ is the total number of unambiguous haplotypes defined by the SNPs $s_i,s_{i+1},\ldots s_j$.
- $f(i,\ldots,j)$: the number of tag SNPs within the block. Given a set of disjointed blocks, $B=\{B_1,B_2,\ldots,B_I\}$ and $B_1 \prec \ldots \prec B_I$, where $B_1 \prec B_2$ indicates that the last SNP of B_1 is located before the first SNP of B_2, (if the last SNP of B_1 and the first SNP of B_2 are not consecutive, the interval between them is excluded from this block partition); the total number of tag SNPs for these blocks is defined by $f(B) = \sum_{i=1}^{I} f(B_i)$.
- $L(i,\ldots,j)$: the length of the block. We simply define it as the number of SNPs in this block, $L(i,\ldots,j) = j - i + 1$. We can also define it as the actual length of the genome spanning from the ith SNP to the jth SNP. Given a set of disjointed blocks, $B = \{B_1,B_2,\ldots,B_I\}$, the total length for these blocks is $L(B) = \sum_{i=1}^{I} L(B_i)$.

With a given number of tag SNPs, our goal is to find the haplotype block partition to maximize the total length of the region included. We formulate the problem as follows:

Block partition with a fixed number of tag SNPs (FTS).— Given K haplotypes consisting of n consecutive SNPs and an integer m, find a set of disjointed blocks $B = \{B_1,B_2,\dots,B_1\}$ with $f(B) \leqslant m$ such that $L(B)$ is maximized.

This problem can be converted to an equivalent, "dual" problem as follows:

Block partition with a fixed genome coverage (FGC).—Given a chromosome with length L, K haplotypes consisting of n consecutive SNPs, and $\beta \leqslant 1$, find a set of disjoint blocks $B = \{B_1,B_2,\dots,B_l\}$ with $L(B) \geqslant \beta L$ such that $f(B)$ is minimized.

In the following, we propose a two-dimensional (2D) dynamic programming algorithm for the FTS problem and then a parametric dynamic programming algorithm for the FGC problem.

A 2D Dynamic Programming Algorithm

Let $S(j,k)$ be the maximum length of the genome that is covered by, at most, k tag SNPs for the optimal block partition of the first j SNPs, $j = 1,2,\dots,n$. Set $S(0,k) = 0$ for any $k \geqslant 0$ and $S(0,k) = -\infty$ for any $K < 0$. Then,

$$S(j,k) = \max\ [S(j-1,k)]$$

and

$$S(j,k) = \max\ \{s[i-1,k-f(i,\dots,j)] + L(i,\dots,j)\}$$
$$\text{for all } 1 \leqslant i \leqslant j \text{ where } block(i,\dots,j) = 1\ .$$

Let $B = \{B_1,\dots,B_J\}$ be the set of disjointed blocks for $S(j,k)$, such that $L(B)$ is maximal with the constraint $f(B) \leqslant k$. Then either the last block B_J ends before j, such that $S(j,k) = S(j-1,k)$, or B_J ends exactly at j and starts at some $i^*,1 \leqslant i^* \leqslant j$, such that $S(j,k) = S[i^* - 1,k - f(B)] + L(B_J)$. Using this recursion, we can design a dynamic programming algorithm to compute $S(m,n)$, the maximum length of genome that is covered by m tag SNPs. The optimal block partition B can be found by backtracking the elements of S that contribute to $S(m,n)$.

The space complexity for this algorithm is $O(m \cdot n)$. If we have precomputed the values of $block(\cdot)$, $f(\cdot)$, and $L(\cdot)$, then the time complexity of this algorithm is $O(N \cdot m \cdot n)$, where N is the number of SNPs contained in the largest block, and $N \ll n$ generally. In fact, given a block of k SNPs (i.e., $s_i,\dots,S_{i+k-1}$), the computation time for $L(\cdot)$ is $O(1)$, and the computing time for $block(i,\dots,i+k-1)$ is $O(K^2N)$, because we need to determine whether any two of the K haplotypes are compatible at these k SNPs in the block. In total, there are, at most, $O(nN)$ blocks, which requires $O(K^2N^2n)$ time for computing all values of $block(\cdot)$. As mentioned elsewhere (Zhang et al. 2002*b*), the problem of calculating $f(i,\dots,i+k-1)$ is NP complete, which means that there are no polynomial time algorithms computing $f(\cdot)$ for any input. Theoretically, the time needed for the enumeration method proposed elsewhere (Zhang et al. 2002*b*) is, at most, $O(NK)$, but it is much shorter in practice. Considering the computation of $block(\cdot)$ and $f(\cdot)$, the overall time complexity becomes $O(K^2N^{K+2}n + Nmn)$.

A Parametric Dynamic Programming Algorithm

For a consecutive set of SNPs $i,\dots,j$, if $block(i,\dots,j) = 1$ and if this block is included in the partition, then $f(i,\dots,j)$ equals the number of tag SNPs. If these SNPs are excluded in the partition, the penalty for this exclusion is defined as $\lambda L(i,\dots,j)$, where λ is the parameter for deletion and $\lambda \geqslant 0$. λ can be regarded as the penalty for each unit length of the excluded regions. Using this scoring scheme, we can score a block partition by $f(B) + \lambda L(E)$, where B represents the included blocks, and E represents the excluded SNPs. Let the scoring function $S(j,\lambda)$ be the minimum score for the optimal block partition of the first j SNPs $(j = 1,2,\dots,n)$ with respect to the deletion parameter λ. Let $S(j,\lambda) = 0$. We can apply the dynamic programming algorithm to obtain $S(j,\lambda)$ by the following recursion:

$$S(j,\lambda) = \min\ [S(i-1,\lambda) + \lambda L(i,\dots,j),1 \leqslant i \leqslant j]$$

and

$$S(j,\lambda) = \min\ [S(i-1,\lambda) + f(i,\dots,j),1 \leqslant i \leqslant j]$$
$$\text{and } block(i,\dots,j) = 1\ .$$

For any given $\lambda \geqslant 0$, the dynamic programming algorithm that uses the above recursion can compute the minimum score.

For any j, if there exists i^* satisfying $1 \leqslant i^* \leqslant j$ and $S(j,\lambda) = S(i^* - 1,\lambda) + \lambda L(i^*,\dots,j)$, then the block $[i^*,\dots,j]$ is included in the partition. Otherwise, there must exist i^* satisfying $1\leqslant i^*\leqslant j$ and $S(j,\lambda) = S(i^* - 1,\lambda) + f(i^*,\dots,j)$, such that the interval $[i^*,\dots,j]$ is excluded from the partition. The penalty for the excluded intervals equals the product of and the total length of these intervals. For any $\lambda \geqslant 0$, $S(n,\lambda)$ equals the sum of the total number of tag SNPs for included blocks and the penalty for excluded intervals. It should be noted that the parametric dynamic programming method is a classical computational tool in sequence alignment, in which the parameters are the weight of matches, mismatches, insertions/deletions, and gaps (Waterman et al. 1992; Gusfield et al. 1994). In the following, we use a method similar to that used by Gusfield et al. (1994) and Waterman et al. (1992), to study the properties of block partitions according to the deletion parameter λ.

Obviously, $S(n,0) = 0$, since all SNPs are excluded from the block partition, and $S(n,\infty)$ equals the minimum number of tag SNPs for the entire genome, because all SNPs are included in the block partition. $S(n,\infty)$ can be obtained by the dynamic programming algorithm described elsewhere (Zhang et al. (2002*b*). For any fixed $\lambda > 0$, the parametric dynamic programming algorithm can compute the optimal solution with included blocks and excluded intervals. Let the length of the included blocks be equal to βL. Then, the number of tag SNPs is $S(n,\lambda) - \lambda(1-\beta)L$. In fact, $S(n,\lambda) - \lambda(1-\beta)L$ must be equal to the minimum number of tag SNPs that is necessary to include at least βL of the genome length. Otherwise, there must exist a partition that needs only $m < S(n,\lambda) - \lambda(1-\beta)L$ tag SNPs to cover at least βL of the genome, and this partition can reduce the score to $m + \lambda(1-\beta)L < S(n,\lambda)$, which contradicts the assumption that $S(n,\lambda)$ is the minimum. Using ideas similar to those discussed by Waterman et al. (1992), it can be shown that $S(n,\lambda)$ has the following properties:

$S(n,\lambda)$ is an increasing, piecewise-linear, and convex function of λ. The right-most linear segment of $S(n,\lambda)$ is constant. The intercept and slope for $S(n,\lambda)$ for each piecewise-linear segment are the total number of tag SNPs and the total length of excluded intervals, respectively.

Waterman et al. (1992) proposed a method to find $S(n,\lambda)$ for all $\lambda \geqslant 0$ efficiently. To make the present article self-contained, a brief description of the idea and a brief sketch of the algorithm are given in this paragraph and the next. Assume that, for an arbitrary λ_i, we obtain $S(n,\lambda_i) = a + \lambda_i b$, where a equals the number of tag SNPs for the included blocks, and b equals the length of the deleted intervals. Define a linear function $s(\lambda) = a + \lambda b$ by a and b. By definition, $S(n,\lambda)$ is minimal. Therefore, $S(\lambda) \leqslant S(n,\lambda)$ for any $\lambda \geqslant 0$, including λ_i where $s(\lambda_i) = (n,\lambda_i)$. Since $S(n,\lambda)$ is piecewise linear, the point $(\lambda_i, S(n,\lambda_i))$ must be located within some linear segment $L_i(\lambda)$ of $S(n,\lambda)$. If $(\lambda_i, S(n,\lambda_i))$ is not located at the end of $L_i(\lambda)$, $s(\lambda)$ should be exactly the line defined by $L_i(\lambda)$, because $s(\lambda) \geqslant L_i(\lambda)$ when λ is $\sim\lambda_i$, and, at the same time, $s(\lambda_i) = L_i(\lambda_i)$. Repeating this idea, we are able to find all such line segments for $S(n,\lambda)$.

The algorithm begins with $S(n,0)$, and $S(n,\infty)$, $S(n,0) = 0$, and the slope of the corresponding line L_0 equals the total length of the genome. $S(n,\infty)$ equals the minimum number of tag SNPs, and the corresponding line L_∞ is horizontal. Let the intersection point for L_∞ and L_0 be (x,y). If $S(n,x) = 0$, then L_0 and L_∞ together define the entire function of $S(n,\lambda)$. Otherwise, $S(n,x) < y$, and the corresponding line L_x for the point $[x,S(n,x)]$ intersects both L_0 and L_∞. We then divide λ into two regions—$[0,x]$ and $[x,\infty]$—and repeat the above procedure for these two regions separately, until all the line segments of $S(n,\lambda)$ are found.

To find all the intersection points of the piecewise-linear segments of $S(n,\lambda)$, we need to compute $S(n,\lambda)$ for any specific λ efficiently. In the above algorithm, the calculation of $\min_{1\leqslant i\leqslant j}(S(i-1,\lambda) + f(i,\ldots,j))$ depends on the block structure of the haplotypes and is the same as the dynamic programming algorithm (Zhang et al. 2002*b*). If we have precomputed the values of $block(\cdot)$, $f(\cdot)$, and $L(\cdot)$, the parametric algorithm takes $O(Nn^2)$ time, where N is the number of SNPs contained in the largest block. However, if $L(\cdot)$ is an additive function, we can improve the algorithm to $O(Nn)$ time (Waterman et al. 1992; Waterman 1995). Considering the computation of $block(\cdot), f(\cdot)$, the total time for finding $S(n,\lambda)$ is $O(K^2N^{K+2}n + NSn)$, where K is the total number of haplotype samples, and S is the number of segments in $S(n,\lambda)$, which is less than the total number of tag SNPs.

After finding all the line segments of $S(n,\lambda)$, we know the entire function of $S(n,\lambda)$. At each intersection point $[x,S(n,x)]$, several block partitions with different numbers of tag SNPs and lengths of excluded intervals may have the same score. We will choose the right-most one with the maximum number of tag SNPs and the minimum length of excluded intervals. For each line segment between two intersection points, both the total number of tag SNPs and the total length of excluded intervals are constant along this segment, and both are equivalent for the low intersection point between this segment and the previous segment. We can sort the number of tag SNPs in ascending order, according to the deletion parameters at the intersection points. The gaps between these numbers give us information as to how the block partition is affected by the deletion parameter λ.

Results

Source of Data

We test our algorithms on a data set of human chromosome 21 reported by Patil et al. (2001). The data set includes 20 haplotypes of 24,047 SNPs (at least 10% minor allele frequency) spanning >32.4 Mb. These SNPs are located in four contigs. Here, we apply our algorithms to the largest contig, NT_002836, which contains 21,840 SNPs. We search the dbSNP database and the human genome resources in the NCBI database to identify SNPs in genes, introns, exons, and coding regions and nonsynonymous SNPs. We distinguish coding regions from exons, because not all exons are translated into proteins. In fact, about a third of all the exons are untranslated. We then apply the algorithms to partition

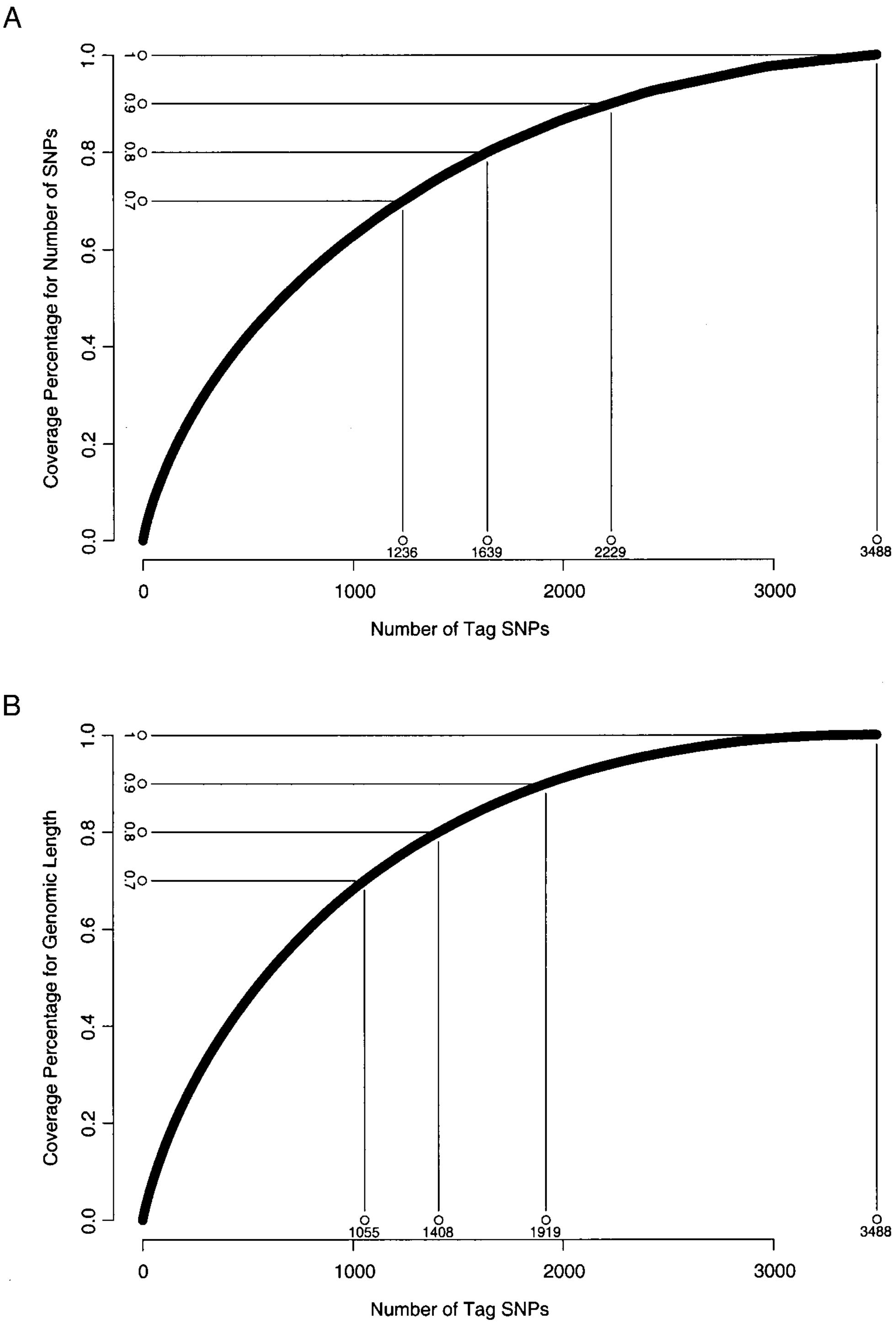

Figure 1 Results of the 2D dynamic programming for block partitioning. *A*, Relationship between the number of tag SNPs and the percentage of the total number of SNPs being included. *B*, Relationship between the number of tag SNPs and the percentage of the actual genomic length being included.

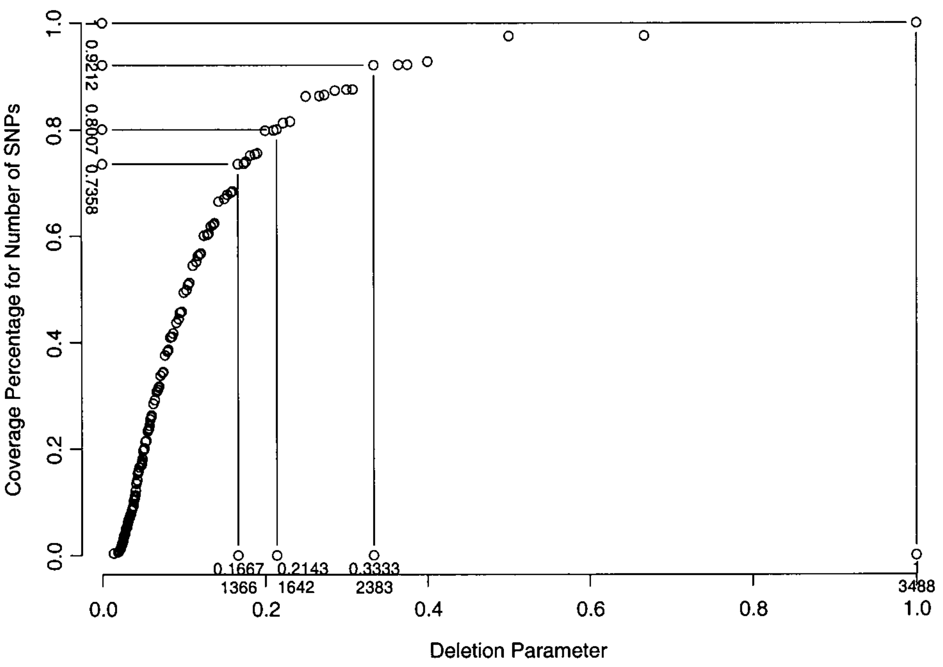

Figure 2 Relationship between the deletion parameter (λ) and the corresponding coverage of the total number of SNPs at the intersection points of piecewise-linear segments of the score function $S(n,\lambda)$, using the parametric dynamic programming algorithm. The penalty for a set of excluded consecutive SNPs is chosen as the product of the deletion parameter λ and the number of SNPs.

the haplotypes into blocks that include all the SNPs in coding regions.

Haplotype Blocks with Limited Resources

The parameters used in the algorithms are set as follows. The number of tag SNPs for a block, $f(\cdot)$, is defined by the minimum number of SNPs that can distinguish at least $\alpha = 80\%$ of unambiguous haplotypes, and $f(\cdot) = 1$ when there is one haplotype with ⩾80% frequency. $L(i,\ldots,j)$ is set as either the number of SNPs in included blocks or the genome length of these blocks. Figure 1*A* shows the relationship between the number of tag SNPs and the ratio of the number of SNPs in included blocks over the total number of SNPs, using the 2D dynamic programming algorithm. The minimum number of tag SNPs that can cover 70%, 80%, 90%, and 100% of all SNPs are also shown in figure 1*A*. A total of 3,488 tag SNPs can cover 100% of SNPs, and this number becomes 2,229 for coverage of 90% of SNPs and 1,639 for coverage of 80% of SNPs. Figure 1*B* shows the relationship between the number of tag SNPs and the ratio of the length of the sequence of included blocks over the length of the whole sequence, as well as the minimum number of tag SNPs for covering 70%, 80%, 90%, and 100% of the genome sequence.

We also implement the parametric dynamic programming algorithm and test it on the same data set. We set $L(\cdot)$ to be the number of SNPs, $L(i,\ldots,j) = j - i + 1$. Figure 2 shows the relationship between the percentages of the total number of SNPs being included and the deletion parameter λ at the intersection points of the piecewise-linear segments. Figure 3 shows the relationship between the percentages of the total number of SNPs being included and the number of tag SNPs required at the intersection points of the piecewise-linear segments. When figure 3 is compared with figure 1, the corresponding numbers of tag SNPs shown in each figure give comparable percentages of the total number of SNPs included. The scoring function with respect to the deletion parameter with the right-most segment of each intersection point is shown in figure 4. The slopes of the lines connecting two adjacent intersection points in figures 3 and 4 are very useful for selecting tag SNPs for genotyping.

SNPs at Coding and Noncoding Regions

We searched the dbSNP database and the human genome resources of the NCBI database to identify SNPs in genes, introns, exons, coding regions, and nonsynonymous SNPs. We distinguish coding regions from exons because not all exons are translated into proteins.

Instead of using the contig NT_002836, which is not mapped in the human genome at the NCBI database, we use the mapped contig NT_011512, which contains the

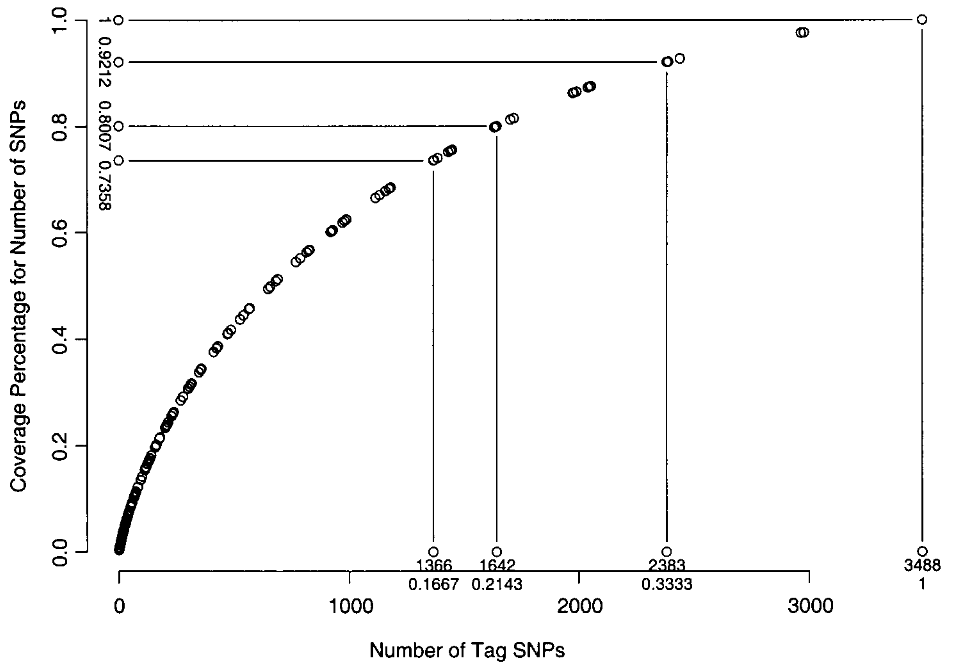

Figure 3 Relationship between the number of tag SNPs and the corresponding percentage of the total number of SNPs being included at the intersection points of piecewise-linear segments of the score function $S(n,\lambda)$, using the parametric dynamic programming algorithm. The penalty for a set of excluded consecutive SNPs is chosen as the product of deletion parameter λ and the number of SNPs.

contig NT_002836, to search the positions of the SNPs. On July 18, 2002, the length of the contig NT_011512 was 28,512,199 bp. There are 380 mapped genes and 38,083 mapped SNPs in the contig NT_011512 in the NCBI database. Among the 380 genes, 357 have both exon information and coding information, and the rest are either pseudogenes or unconfirmed genes. We therefore included only these 357 genes in our analysis. The total lengths of the genes, the exons, and the coding regions are 15,425,073, 486,409, and 315,574 bp, respectively, corresponding to 54.10%, 1.71%, and 1.11%, respectively, of contig NT_011512. Among the 21,840 SNPs used by Patil et al. (2001), 20,503 are mapped to the dbSNP database and the human genome resources of the NCBI database, and 38 have ambiguous map positions (at least two positions in one contig), which were excluded from further analysis.

Table 1 shows the numbers of SNPs in genes, exons, and coding regions, together with the number of nonsynonymous SNPs. Table 2 shows the density of SNPs (expressed as the average number of SNPs per kilobase and its 95% CI) in intergenic regions, genes, introns, exons, and coding regions, based on the data of Patil. et al. (2001). The last row gives the P values of the observed data under the null hypothesis that SNPs are uniformly distributed along the chromosome. Table 2 shows that the densities of SNPs in intergenic regions, genes, introns, and coding regions are similar to the average density along the chromosome. However, the density in exons is somewhat lower than the average ($P = .1$). This observation is consistent with previous studies on SNPs in coding and noncoding regions. The reason for the relatively high density of SNPs in genes is that most of the gene regions are in introns. However, it is surprising that the density in coding regions is somewhat higher than that in exons. Figure 5 shows histograms for the number of genes with different numbers of SNPs in genes, exons, and coding regions and the number of nonsynonymous SNPs.

We partition the haplotypes into blocks, using the dynamic programming algorithm (Zhang et al. 2002*b*),

Table 1

Numbers of SNPs in Different Regions of the Genome

	No. of SNPs in	
Type of SNPs	NT_002836[a]	NT_011512[b]
Mapped	20,503	38,089
Ambiguously mapped	38	117
In genes	11,055	20,814
In introns	10,736	20,041
In exons	319	773
In coding regions	221	476
Nonsynonymous	144	278

[a] Patil et al. 2001.
[b] NCBI database.

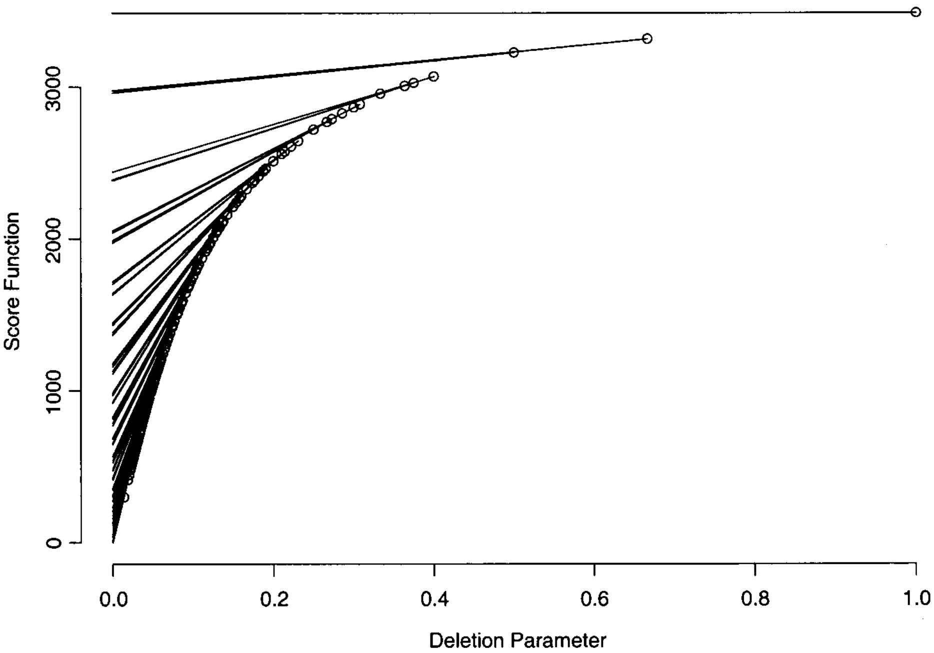

Figure 4 Scoring function $S(n,\lambda)$ with respect to the deletion parameter λ. The penalty for a set of excluded consecutive SNPs is chosen as the product of deletion parameter λ and the number of SNPs.

and test whether the starting SNPs of the blocks are evenly distributed along the chromosome. The number of tag SNPs in each block, measured by $f(\cdot)$, is defined as the minimum number of SNPs that can distinguish at least 80% of unambiguous haplotypes. We obtain a total of 2,182 blocks. The number of starting SNPs in different regions is given in table 3. We assess the association between the SNPs in the beginning of the haplotype blocks and in regions with known biological functions, such as exons. Differences in distribution of the starting SNPs along the chromsome were not statistically signficant.

To investigate the pattern of LD in the haplotype blocks obtained above, we plot a histogram of the number of blocks with different numbers of distinct haplotypes (fig. 6). We find that most of the blocks contain seven or fewer distinct haplotypes; only 213 blocks (~10%) contain eight or more distinct haplotypes, and 1,204 (55%) contain four or fewer distinct haplotypes. A small number of haplotypes within a block indicates a strong LD signal.

One of the advantages of the algorithms used in the present study is that functions of SNPs, such as whether they are in coding or noncoding regions, can be incorporated into the algorithms. In an association study, investigators may put more weight on SNPs within coding regions than on those in noncoding regions. One method is to add higher penalty to the SNPs in coding regions. In the parametric dynamic programming algorithm, we define the length of a set of consecutive SNPs $i,\ldots,j$,$L(i,\ldots,j)$ as follows:

$$L(i,\ldots,j) = (j - i + 1 - n_c) + T^{*}n_c \ ,$$

where n_c is the number of SNPs in coding regions and T is a relatively large positive number. We implemented the parametric dynamic programming algorithm, using this definition of $L(\cdot)$, to obtain the corresponding block partitions. We set $T = 1{,}000$. The number of tag SNPs for a block, $f(\cdot)$, is defined by the minimum number of SNPs that can distinguish at $\alpha \geqslant 80\%$ of unambiguous haplotypes. We let $f(\cdot) = 1$ if a single haplotype has a

Table 2

Densities of SNPs and Their CIs in Different Regions of the Genome for the Contig NT_002836

	Density of SNPs in				
	Intergenic Regions	Genes	Introns	Exons	Coding Regions
SNPs/kb	.722	.717	.719	.656	.700
95% CI	(.71–.73)	(.71–.73)	(.71–.73)	(.65–.79)	(.63–.81)
P^{a}	.60	.60	.93	.10	.69

NOTE.—Contig NT_002836 is described by Patil et al. (2001).

[a] Calculated under the null hypothesis that the SNPs are uniformly distributed along the chromosome.

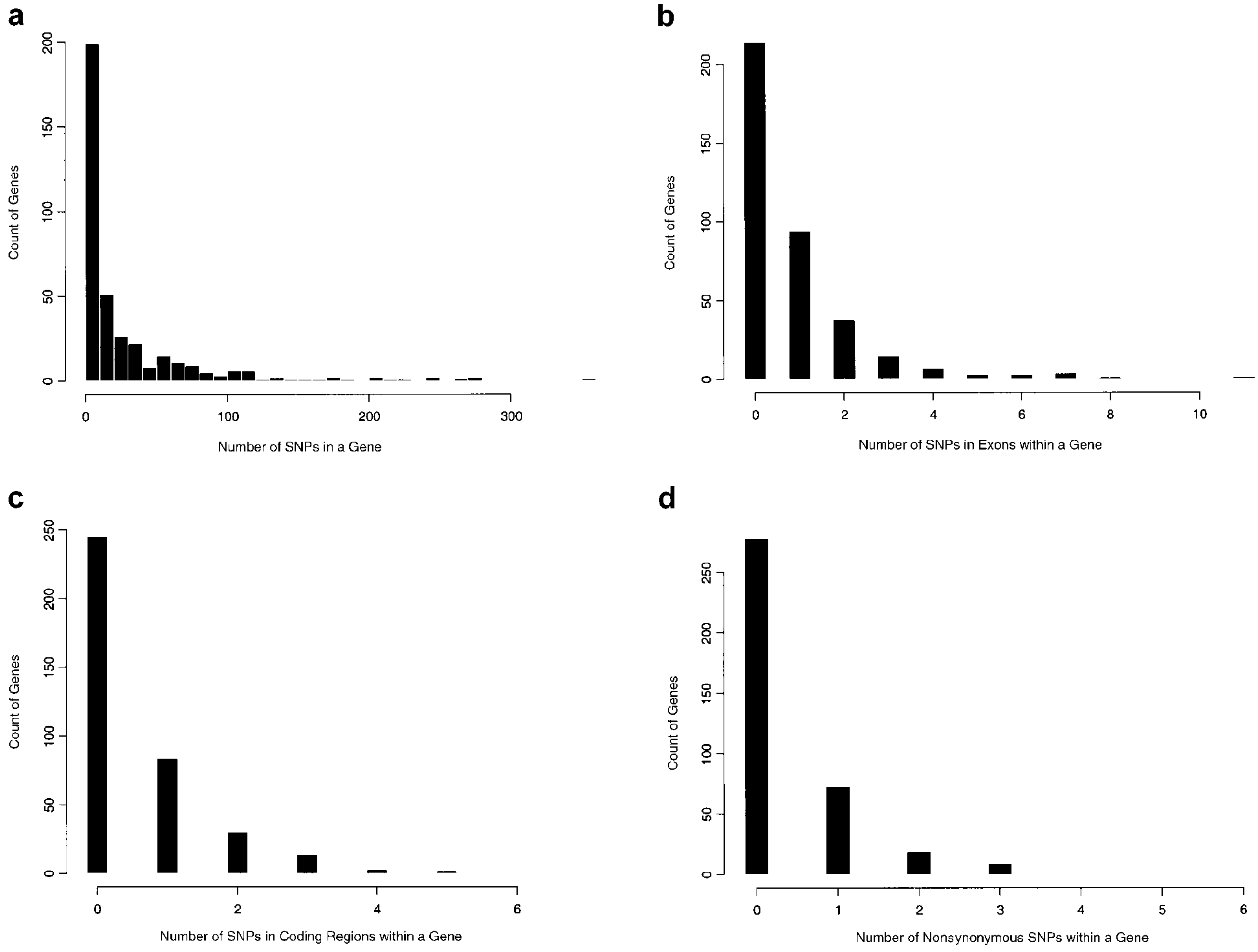

Figure 5 The histograms for the number of genes according to the number of SNPs in genes (*a*), the number of SNPs in exons (*b*), the number of SNPs in coding regions (*c*), and the number of non-synonymous SNPs (*d*).

frequency ⩾80%. For example, we obtain 596 blocks with 1,081 tag SNPs representing 14,048 SNPs when the deletion parameter α is set at 0.140. All 221 SNPs in the coding regions are included in this block partition.

Table 3

The Number of Starting SNPs in Blocks and the Average Number of Starting SNPs per Megabase in Various Regions of the Genome

	No. of SNPs in			
	Intergenic Regions	Genes	Exons	Coding Regions
Starting SNPs	1,025	1,157	39	25
Starting SNPs/Mb	78	75	80	79
P^a	.49	.49	.82	.89

[a] Calculated under the null hypothesis that the SNPs are uniformly distributed along the chromosome.

Discussion

Several recent studies have suggested that the human genome can be divided into blocks with high LD within each block. Because of this feature, a relatively small fraction of SNPs can capture most of the haplotypes in each block. Previously, Zhang et al. (2002*b*) developed a dynamic programming algorithm for haplotype block partition to minimize the total number of tag SNPs across the whole genome. However, in a genetic study with limited resources, investigators may wish to genotype only a fixed number of SNPs. The problem then becomes how to choose tag SNPs and the corresponding genomic regions to maximize the genome coverage with the fixed number of tag SNPs. In the present article, we formulate the problem of finding the haplotype blocks by use of a restricted number of tag SNPs. We provide

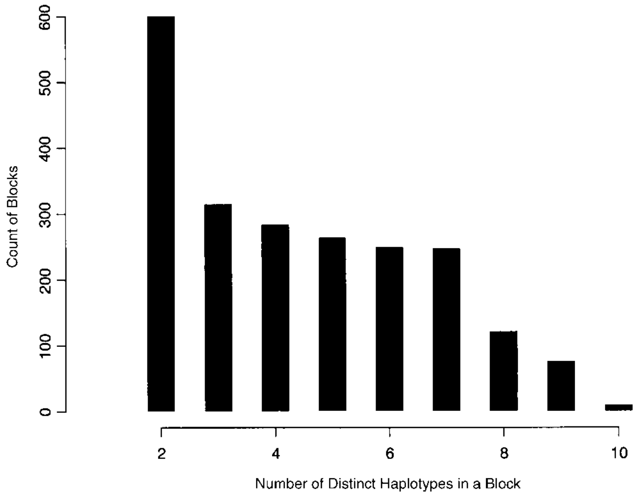

Figure 6 Histogram for the number of blocks according to the number of distinct haplotypes in a block. The blocks are obtained on the basis of a data set of human chromosome 21 from Patil et al. 2001, using the dynamic programming algorithm (Zhang et al. 2002*b*). The number of tag SNPs in each block, measured by $f(\cdot)$, is defined as the minimum number of SNPs that can distinguish at least 80% of unambiguous haplotypes.

two dynamic program algorithms to solve this problem. One of the advantages of the algorithms is that properties of SNPs, such as whether they are in coding or noncoding regions, can be incorporated into the algorithms, whereas, in an association study, investigators may put more weight on SNPs within coding regions than on those in noncoding regions.

As an initial step to understand the biological implications of haplotype blocks, we characterize the relationship between the starting SNPs of haplotype blocks from the optimal block partition and the SNPs in regions with known biological functions. We find that the starting SNPs of haplotype blocks are evenly distributed in genes, exons, and coding regions.

We apply the algorithms for haplotype block partition with limited resources to a contig of a data set of SNPs on human chromosome 21. In this example, we require that all the SNPs in coding regions be selected. The algorithms developed in the present study are flexible enough to allow investigators to decide the weights for SNPs with different functions. In the present article, we use the fraction of haplotypes represented by the tag SNPs as a quality measure. Other quality measures, such as haplotype diversity (Johnson et al. 2001), can be easily incorporated into the programs. The output from the algorithms can guide investigators to SNPs for genotyping, to maximize the success of association studies.

The algorithms developed in the present article are based on haplotype data. Although laboratory techniques, such as allele-specific long-range PCR (Michlataos-Beloin et al. 1996) or diploid-to-haploid conversion (Douglas et al. 2001), have been used to determine haplotypes in diploid individuals, these approaches are technically difficult, labor intensive, and expensive. Most of the time, it is unrealistic to do a large-scale study across the whole genome, as was done by Patil et al. (2001). For these reasons, multiple sets of large-scale genotype data rather than haplotype data are being generated. Thus, it is important to develop methods to extract haplotype block information from genotype data directly. The dynamic programming algorithms developed in the present study, combined with methods for haplotype inference (e.g., Qin et al. 2002) can be used to achieve this objective.

Acknowledgments

We learned of the problem of haplotype block partition with limited resources from Dr. David Cox and Mr. Bruce Moxon when three of us (F.S., M.S.W., and T.C) visited Perlegen Sciences in 2002. The input of Dr. Cox and Mr. Moxon to this project is greatly appreciated. This research was partially supported by National Institutes of Health grants DK53392 and R01-RR16522-01, by National Science Foundation grant EIA-0112934, and by the University of Southern California. We thank two anonymous reviewers for their detailed suggestions, which helped us to improve our presentation.

Electronic-Database Information

URLs for data presented herein are as follows:

dbSNP, http://www.ncbi.nlm.nih.gov/SNP/
HapBlock, http://hto.usc.edu/~msms/HapBlock
National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov/genome/guide/human/

References

Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES (2001) High-resolution haplotype structure in the human genome. Nat Genet 29:229–232

Dawson E, Abecasis GR, Bumpstead S, Chen Y, Hunt S, Beare DM, Pabial J, et al (2002) A first-generation linkage disequilibrium map of chromosome 22. Nature 418:544–548

Douglas JA, Boehnke M, Gillanders E, Trent JM, Gruber SB (2001) Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies. Nat Genet 28:361–364

Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Liu-Cordero SN, Rotimi C, Adeyemo A, Cooper R, Ward R, Lander ES, Daly MJ, Altshuler D (2002) The structure of haplotype blocks in the human genome. Science 296:2225–2229

Gusfield D, Balasubramanian K, Naor D (1994) Parametric optimization of sequence alignment. Algorithmica 12:312–326

Johnson GCL, Esposito L, Barratt BJ, Smith AN, Heward J, Di Genova G, Ueda H, Cordell HJ, Eaves IA, Dudbridge F, Twells RCJ, Payne F, Hughes W, Nutland S, Stevens H, Carr P, Tuomilehto-Wolf E, Tuomilehto J, Gough SCL, Clayton DG, Todd JA (2001) Haplotype tagging for the identification of common disease genes. Nat Genet 29:233–237

Michlataos-Beloin S, Tishkoff SA, Bentley KL, Kidd KK, Ruano G (1996) Molecular haplotyping of genetic markers 10 kb apart by allelic-specific long-range PCR. Nucleic Acids Res 24:4841–4843

Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, Kautzer CR, Lee DH, Marjoribanks C, McDonough DP, Nguyen BTN, Norris MC, Sheehan JB, Shen NP, Stern D, Stokowski RP, Thomas DJ, Trulson MO, Vyas KR, Frazer KA, Fodor SPA, Cox DR (2001) Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science 294:1719–1723

Qin Z, Niu T, Liu J (2002) Partitioning-ligation–expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. Am J Hum Genet 71: 1242–1247

Wang DG, Fan JB, Siao CJ, Berno A, Young P, Sapolsky R, Ghandour G, et al (1998) Large-scale identification, mapping and genotyping of single-nucleotide polymorphisms in the human genome. Science 280:1077–1082

Waterman MS (1995) Introduction to computational biology: maps, sequences and genomes. Chapman & Hall/CRC Press, Boca Raton, FL

Waterman MS, Eggert M, Lander EL (1992) Parametric sequence comparisons. Proc Natl Acad Sci USA 89:6090–6093

Zhang K, Calabrese P, Nordborg M, Sun F (2002*a*) Haplotype block structure and its applications in association studies: power and study design. Am J Hum Genet 71:1386–1394

Zhang K, Deng M, Chen T, Waterman MS, Sun F (2002*b*) A dynamic programming algorithm for haplotype block partitioning. Proc Natl Acad Sci USA 99:7335–7339