Comprehensive overview of various haplotype block inference methods

Sven Laur

February 17, 2009

Abstract

During last years there has been a breakthrough in genetics and biotechnology. Present technology allows large scale association studies between genotypes and diseases. The most powerful methodology of association studies is the analyse of single nucleotide polymorphisms (SNPs). The estimated number of SNPs in the human genome is 10 million. Therefore, SNP markers provide practical fine-grain map of human genome. Theoretical models and practical experiments suggest that a single strand of DNA is composed from haplotype blocks—sequences of fixed DNA strands. In other words, few suitably chosen SNPs can reveal most of DNA strand and reduce the number of required SNPs for association studies.

The following review gives a introduction to basic concepts and models. We analyse pros and cons of various definitions of haplotype blocks, point out advantages and limitations of inference methods. We cover all three main approaches: methods based on marker pairs, combinatorial methods and models with minimal-description length. We also briefly discuss the optimal marker set (OMS) problem that is a common for all methodologies. Good solution can provide high quality of the end results and lower the measurement costs. However, the complete treatment of OMS is out of our scope.

1 Introduction

The human genome sequence consists of 3 billion DNA base-pairs which are divided into 46 chromosomes. The chromosomes are coupled into pairs holding two copies of genes. The latter makes it difficult to extract one DNA strand (haplotype), since both chromosomes contribute to the measurements. Although asymmetric methods allow to isolate and measure single chromosomes, they are more sophisticated and resource demanding. Thus asymmetric methods are not suitable for large scale studies. Alternative strategy is to measure both chromosomes at the same time and afterwards restore original DNA strands by statistical inference.

The gargantuan number of base-pairs poses a big problem for large scale DNA sequencing. Somewhat surprisingly, 99.9% of base-pairs are identical

among different haplotypes. The natural variation of haplotypes causes multiple differences (polymorphisms) like substitutions, insertions and deletions of base-pairs. The most common are single-nucleotide polymorphisms (SNPs)—local single base-pair differences. The SNP are distributed evenly over the genome and occur approximately after every thousand base-pairs. Therefore, measurements of SNPs reveal local structure of the haplotype. Furthermore, current technology favours SNP measurements and offers an economical method for association studies. The correlation between the occurrence of SNPs and disease symptoms indicates possible causal relationship which can be confirmed or rejected by more elaborate analysis.

One outcome of the The Human Genome Project was approximately 3.7 millions SNPs that were discovered and localised [13]. Another authoritative collection was composed by the International SNP Map Working Group, it contains 1.4 millions of SNPs [21]. These figures are in the good accordance with the theoretical estimates of 4-10 million SNPs [28]. The strong correlation between neighbouring SNPs provides a shortcut to genome analysis. Various estimates state that only 100–500 thousand tag SNPs are sufficient for association studies, because properly chosen markers can identify large haplotype blocks. The main aim of the Haplotype Map project [25] is to find an optimal set of SNPs and the corresponding haplotype blocks so that the whole genome is covered, and publish the corresponding map.

Inference of haplotype blocks consists of three main steps. First haplotypes are extracted from experimental data. Usually biological measurements provide only genotype—a sequence of unordered SNP pairs that is formed by two different haplotypes. The identification of haplotypes is non-trivial task, since there are exponential number of possible solutions. The most common approaches are Clark's [12, 22] and EM-algorithms [16, 17]. Some recent methods are based on Bayesian inference and Monte-Carlo simulation [38, 33]. Some methods combine the haplotype inference with the block identification.

In the next stage haplotype blocks are determined. The three main approaches use different criteria. The most straightforward approach is to use correlation between SNPs to determine block boundaries [18]. An alternative combinatorial approach [34, 47] seeks such a block structure that minimises the number of tag SNPs. Third alternative [31, 2, 19] is based on minimum-description length principle—a model with minimum description is inferred from data.

In the last stage, locations of SNPs that identify the haplotype blocks are determined. The underlying combinatorial problem corresponds to the $\mathcal{NP}$-hard test-set problem. Fortunately, the number of haplotype blocks is often small and thus the exact solution is feasible. Nevertheless, the straightforward exploration of the search space can be quite demanding, especially when haplotype blocks are long. Therefore, additional insights for pruning the search space are essential for efficiency.

Since there are so many different approaches, we try to formalise, analyse and compare them. Each method has its own advantages and there is no perfect haplotype block definition. There are just too many different and often opposite

goals. However, the minimal number of tag SNPs seems to be the most compelling for large scale studies, even if the inferred haplotypes are not biologically justified.

For practical reasons, the method should be relatively robust and handle missing data without additional complications, otherwise unavoidable measurement errors make the method unstable or too complicated. Another important efficiency issue is locality. Although the method should fit model globally, the quality of end result should not drop drastically, if the haplotype is divided into separated regions. More sophisticated methods have at least quadratic complexity, therefore divide-and-conquer strategy reduces significantly the complexity and allows to parallelise calculations.

2 Basic concepts and notations

The human genome is altered by two biological phenomenas: mutations and a meiotic crossover. The effect of mutations is usually local and brings diversity to the human genome. Since the mutation rate is small, these differences propagate through the population. But on the same time the meiotic crossover swaps "random" blocks of parental haplotypes and thus decreases the correlation between different DNA markers. The decay in structure is reversed by bottleneck events when population descend form a small group of individuals. Strong positive selection causes also similar patterns in DNA. In both cases, a haplotype pattern becomes dominant. Afterwards decay continues and the dominant region are broken into haplotype blocks.

We define the haplotype as a sequence of consecutive SNP marker values. Let the number of markers be n and the number of investigated haplotypes m. Then a haplotype segmentation $\mathcal{B} = \big((s_1, e_1), (s_2, e_2), \ldots, (s_r, e_r)\big)$ consists of all start and endpoints of blocks. The block itself is a consecutive sequence of indices $[a, b] = \{a, a_1, \ldots, b\}$. In principle, the segmentation might be incomplete, but the blocks must be non-overlapping and non-empty $s_1 \leq e_1 < s_2 \ldots < s_r \leq e_r$. Each segmentation induces the haplotype blocks. Let us denote the haplotype by $\mathbf{h}$ and the whole dataset as a $m \times n$ matrix H. Then each block $[s_k, e_k]$ will contain q_k different haplotype blocks $\mathbf{a}_s^{(k)}$.

There is no biological reason why the block structure of the haplotype must be unique. For example, consider two sub-populations that have different segmentations $\mathcal{B}_1$ and $\mathcal{B}_2$. Then a reasonable outcome from the segmentation algorithm is the intersection of both partitions. Small neighbouring blocks in the sub-populations are differently correlated depending on the original structure. Therefore, large differences among the transition frequencies

$$\Pr\left[\, \mathbf{a}_t^{(k)} \middle| \mathbf{a}_s^{(k-1)} \,\right], \qquad t = 1, \ldots, q_k, \quad s = 1, \ldots, q_{k-1}$$

suggest existence of different sub-segmentations. In our example $\Pr[\, D_2 | D_1 \,] = 1$ and $\Pr[\, A_2 | D_1 \,] = \Pr[\, B_2 | D_1 \,] = \Pr[\, C_2 | D_1 \,] = 0$ provide strong evidence of two different haplotype segmentations. Of course, the implication holds only if the algorithm returns approximately the intersection of sub-segmentations.

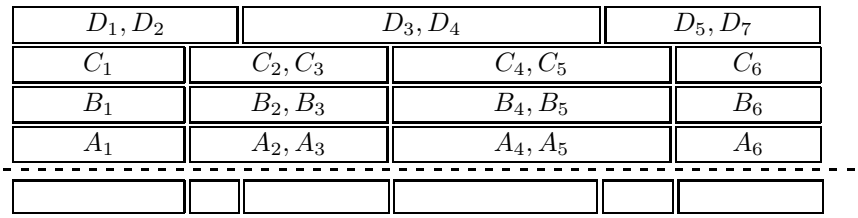

Figure 1: Intersection of two different haplotype segmentations

The simplest statistical indicator of the block structure is joint frequency distribution of two markers. The standard normalised association measure for two markers A and B with two possible alleles is $D' = D/D_{max}$, where

$$D = \Pr[\, A = 1, B = 1\,] - \Pr[\, A = 1\,]\Pr[\, B = 1\,]$$

and

$$\begin{aligned} D_{max} &= \min\{\Pr[\, A = 1\,]\Pr[\, B \neq 1\,], \Pr[\, A \neq 1\,]\Pr[\, B = 1\,]\}, \text{ when } D > 0, \\ D_{max} &= \min\{\Pr[\, A = 1\,]\Pr[\, B = 1\,], \Pr[\, A \neq 1\,]\Pr[\, B \neq 1\,]\}, \text{ when } D < 0. \end{aligned}$$

The generalisation to markers with more than two alleles is a weighted average

$$D' = \sum_{i,j} \Pr[\, A = i\,]\Pr[\, B = j\,] \left|D'_{i,j}\right|,$$

where $D'_{i,j}$ is pairwise measure that is

$$D_{i,j} = \Pr[\, A = i, B = j\,] - \Pr[\, A = i\,]\Pr[\, B = j\,].$$

Alternative measure for di-allelic markers is a correlation

$$r = \frac{\Pr[\, A = 1, B = 1\,] - \Pr[\, A = 1\,]\Pr[\, B = 1\,]}{\sqrt{\Pr[\, A = 1\,]\Pr[\, A \neq 1\,]\Pr[\, B = 1\,]\Pr[\, B \neq 1\,]}}.$$

The most sophisticated method for determination linkage disequilibrium of two markers is the exact Fisher's test. Let f_{ij} be the haplotype counts with alleles $A = i$ and $B = j$ and r_i and c_j marginal counts. Then the combinatorial probability of observed joint frequency F is

$$\Pr[\, F \mid r_1, \ldots, r_m, c_1, \ldots, c_n\,] = \frac{r_1! \cdots r_m! c_1! \cdots c_n!}{N! \prod_{i,j} f_{ij}!},$$

where N is a number of observations. For each possible assignment of f_{ij} that is consistent with marginal distributions conditional probability is calculated. All probabilities that are less than $\Pr[\, F \mid r_1, \ldots, c_n\,]$ are summed up[1]. The sum is a p-value that characterises the significance of linkage disequilibrium. However, the enumerations over all possible assignments quickly comes infeasible and thus Monte-Carlo simulation methods are used.

[1]To define p-value, we need a meaningful linear ordering of matrices F and in principle some other ordering may be more suitable.

3 Methods based on marker correlation

Intuitively, the correlation between markers reveals the information about haplotype blocks. Markers that are always in the same block have high correlation and the markers from different haplotype blocks should have low correlation. The latter is not always true, for example if two neighbouring blocks are highly correlated then the marker correlation is also high. Therefore, the decision whether the markers are connected or separated is not a clear cut.

We start from the simplistic framework and extend it further to the methodology used in [18]. First we define two types of marker pairs.

Definition 1. Let D' be the linkage disequilibrium between markers s and t. Then the pair s, t is strongly connected if $\beta < D'$ and strongly separated if $D' < \alpha$. All pairs that are either strongly connected or separated are informative, others are non-informative.

Ideally, the block should consist of contiguous region strongly connected markers. But mutations and genotyping errors make the demand unrealistic.

Definition 2. The haplotype block is a maximal contiguous region of markers such that over 95% of all informative pairs are strongly connected.

Note that this definition does not guarantee proper segmentation, since the blocks may overlap. Consider a haplotype that consist of three blocks 150, 3 and 150 markers wide. For simplicity, let $D' = 1$ for the marker in the same block and $D' = 0$ for others. Then the middle block can be merged to both sides, because 450 of inter-block pairs is less than 5% of 1,1628 pairs.

The ambiguity around long blocks is unavoidable, therefore conflict handling mechanism must be provided. In case of overlap, the intersection must be assigned to the closer neighbour or recursively divided such way that large blocks do not merge smaller ones. In principle, percentage of strongly connected markers may be low in some sub-blocks of a valid block. Therefore, the best segmentation requires global search and quadratic computational complexity. A more efficient alternative is to fix the size of the look-ahead buffer, that is to allow only a fixed number of invalid sub-blocks during the iterative widening of blocks. This strategy reduces complexity to $\mathcal{O}(ln)$, where l is the maximum length of haplotype block.

Robustness The robustness of the method is determined by parameters α and β. Different errors occurring in the genotyping phase can drastically change LD measures. We refer here robustness results of di-allelic markers [1].

Let A and B be actual and A' and B' observed marker values. Two common error mechanisms during the genotyping are symmetrical and one-sided error. The former postulates that both error types are equiviprobable, although different markers can have different error probabilities μ and ν. One-sided error mechanism excludes one error type, that is $\Pr[A' = 0 \mid A = 1] = \Pr[B' = 0 \mid B = 1] = 0$ and the other errors have probabilities μ and ν. The

	Symmetric		One-sided		Symmetric		One-sided	
$\Pr[B=1]$	D'_T	D'_E	D'_T	D'_E	D'_T	D'_E	D'_T	D'_E
	$\Pr[A=1]=0.90$				$\Pr[A=1]=0.50$			
0.90	1.00	0.67	1.00	0.70	1.00	0.64	1.00	0.70
	0.50	0.33	0.50	0.35	0.50	0.32	0.50	0.35
0.50	1.00	0.64	1.00	0.70	1.00	0.88	1.00	0.94
	0.50	0.32	0.50	0.35	0.50	0.44	0.50	0.47

Table 1: The difference between true value D'_T and estimated value D'_E under symmetrical and one-sided error rate 3% [1].

first order error estimate is obtained by differentiating D' and substituting probability differences by expected differences. For example, in the symmetrical model the difference of $\Pr[A=1,B=1]$ and $\Pr[A'=1,B'=1]$ is

$$\Delta\Pr[11] = -(\mu+\nu)\Pr[11] + \mu\Pr[01] + \nu\Pr[10].$$

Table 1 shows that value of D' changes drastically under modest error rate 3% and makes the segmentation sensitive to errors. It is common to all linkage disequilibrium measures, although some of the are more robust. Of course, proper choice of thresholds makes the method more robust, but in the same time increases the number of non-informative marker pairs.

The sub-segmentations have similar effect on D' as errors. Table 2 illustrates the case with two sub-populations. In the first $D'=1$ and in the second markers are independent. Distributions that are symmetric are also less sensitive and for dominant alleles D' does not change much. In other cases the markers will be probably separated and the resulting segmentation is close to intersection of sub-segmentations.

Further enhancements The value of D is essentially a point estimate and known to fluctuate upwards. The use of confidence intervals makes the inference statistically more justified. This leads to methodology [18], where one-sided confidence intervals are used instead of D'. Confidence intervals of D' are obtained by bootstrap methods, since there is no explicit formulas. Still the method is sensitive to errors, since the erratic distribution causes erratic confidence intervals. Another shortcoming is subjectivity—different thresholds give different results. On the other hand, the running-time is almost linear $\mathcal{O}(lmn)$ and locality is quite well preserved.

4 Four-gamete test

The four-gamete test [26] provides indisputable evidence of recombination events under infinite site model. The infinite site model allows only one mutation per

First distribution	Rate	Second distribution $(0.5, 0.5)$	$(0.9, 0.9)$	$(0.9, 0.1)$	$(0.1, 0.9)$	$(0.1, 0.1)$
$(0.9, 0.9)$	10%	0.79	0.90	0.99	0.99	0.94
	30%	0.56	0.70	0.95	0.95	0.88
$(0.9, 0.5)$	10%	0.64	0.83	0.97	0.17	0.94
	30%	0.32	0.56	0.92	-0.25	0.79
$(0.9, 0.1)$	10%	-0.04	0.50	0.90	-0.33	0.50
	30%	-0.16	0.20	0.70	-0.57	0.20
$(0.5, 0.5)$	10%	0.90	0.96	1.00	1.00	0.96
	30%	0.70	0.79	0.98	0.98	0.89
$(0.5, 0.1)$	10%	0.64	0.89	0.98	0.17	0.83
	30%	0.32	0.79	0.92	-0.25	0.56

Table 2: Linkage disequilibrium D' of two sub-populations the first with $D' = 1$ and the second $D' = 0$. Distributions are fixed by marginal frequencies $\Pr[A = 1]$ and $\Pr[B = 1]$.

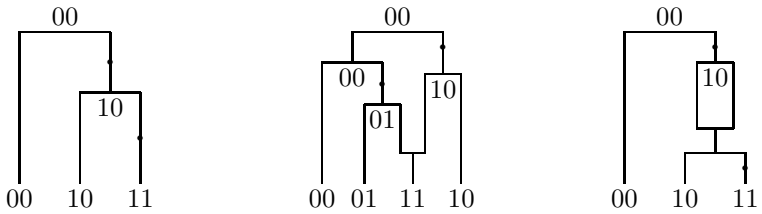

Figure 2: Genealogy graphs: no recombination, detectable recombination, undetectable recombination

base-pair. Since the probability of single mutation of a base-pair is very small, the model is well-justified.

The mutation history can be captured in the binary genealogy tree. The leaves of the tree are observed haplotypes and the other vertices are intermediate haplotype forms. Each intermediate vertex has two children, only one of them can contain a new mutation. Consider now region of m SNPs. If no recombination have occurred, the number of different haplotype patterns is $m+1$, because a mutation can happen only in one subtree.

Recombination joins mutations of different parents and the resulting genealogy graph can have more than $m + 1$ leaves. Since each recombination generates one new vertex in the graph, the maximum number of different patterns is $m + 1 + r$, where r is the number of recombinations. In two marker case four different marker pairs indicate the recombination. This is called four-gamete test (FGT). But every recombination is not captured by FGT, see figure 2.

The four-gamete test provides a neat definition of haplotype block [39] that does not contain any subjectively adjustable parameters.

Definition 3. The haplotype block is a maximal contiguous region of markers that pass the FGT.

The FGT can handle only di-allelic markers and not three or four alleles. The main advantage of the FGT segmentation algorithm is low complexity. First, FGT between two markers takes $\mathcal{O}(m)$ comparisons and the overall complexity is $\mathcal{O}(lmn)$, where l is the length of the largest block.

In principle, block borders depend on starting point, but the difference propagation over a long region is unlikely. These differences can be detected by ignoring iteratively starting markers in the first block. If all runs indicate a same block boundary, then the segmentation after it is unique. Therefore, the algorithm can be run in parallel on several subregions without altering the resulting segmentation.

The failing FGT test between neighbouring markers forces strong block borders. If the population consist from several sub-segmentations, then all strong borders of sub-segmentations will also appear in the global segmentation. In other words, the summary segmentation is close to intersection.

Generally, the FGT method produces more blocks than the other methods. For example, the FGT method found 75 blocks [2]in 5q31 data [14], whereas the evidence suggest existence of of 10–12 blocks. The excessive borders are caused by the working principle of FGT, since markers are separated if there is an evidence of a recombination. In other words, the FGT finds probable recombination borders, but these might be sub-optimal for the haplotype map. Moreover, the infinite site model is not always adequate—mutations can happen more than once.

Another major drawback is sensitivity to genotyping errors. Clearly, a single measuring error can force a new border. Thresholds for marker pairs can make the method more robust, but this requires a larger sample. Also, missing data must be ignored, since imputation methods lead either to erratic borders or are useless.

5 Combinatorial methods

Combinatorial methods have easily interpretable objective. They seek through all possible complete segmentations and output a segmentation with minimal number of required tag SNPs. The utility is evident even if the segmentation has no biological grounding. The methodology was successively introduced through series of articles [34, 45, 47, 49, 48].

First, we describe general dynamic programming strategy. The main idea behind search algorithm is universal and has been used in MDL methods [31, 2]. Algorithm 1 is applicable when the cost function is additive

$$f(\mathcal{B}) = \sum_{i=1}^{r} f(s_i, e_i), \qquad \mathcal{B} = \big((s_1, e_1), \ldots, (s_r, e_r)\big).$$

For the fixed last block $[s, n]$, the best segmentation must have minimal cost over all possible segmentations of region $[1, s-1]$. The algorithm starts dynamically building best segmentations of prefix regions. Let S_j be the minimal cost of the prefix $[1, j]$. Then Algorithm 1 computes iteratively

$$S_{j+1} = \min_{i=1,\ldots,j} S_i + f(i, j).$$

Stored minimising values I_j allow to reconstruct the minimising solution by backtracking.

Lexicographic ordering and multicomponent cost functions allow to achieve objectives with different priorities. For example, to find segmentation with minimum number of SNPs that has minimum number of blocks, we just define $f(i, j) = (\kappa, 1)$, where κ is the minimum number of markers that identify the haplotype block. The lexicographic ordering assures that the optimal segmentation achieves the goal.

Algorithm 1: Generic dynamical programming algorithm

Input: Haplotype data and cost function $f(\cdot, \cdot)$.

Output: Segmentation with minimal cost $f(\mathcal{B}) = f(s_1, e_1) + \cdots + f(s_r, e_r)$.

```
S_0 ← 0
for j = 1 to n do
    S_j ← +∞
    I for i = 1 to j − 1 do
        if S_j > S_{i−1} + f(i, j) then
            I_j ← i − 1
            S_j ← S_{i−1} + f(i, j)
▲▲▲ Restoring the segmentation ▲▲▲
k ← n
B = ∅
while i > 0 do
    Add the pair (I_k + 1, i) to B
    k ← I_k
return B
```

The heart of the algorithm is the cost function f. To define f properly, we must specify what are the haplotype blocks and when is haplotype identified by marker values. As ordinary haplotype data contains missing marker values, there must be a simple procedure for missing data. There are two different approaches combinatorial and statistical. The combinatorial method [34, 47] divides haplotypes into different classes and excludes ambiguous haplotypes.

Definition 4. Let $\mathbf{a} = a_1 \ldots a_l$ and $\mathbf{b} = b_1 \ldots b_l$ be haplotypes. Then haplotypes are compatible $\mathbf{a} \sim \mathbf{b}$ if $a_i = b_i$ for all pairs without missing values. The

haplotype $\mathbf{b}$ is ambiguous if there are two non-compatible haplotypes $\mathbf{a}$ and $\mathbf{c}$ such that $\mathbf{a} \sim \mathbf{b} \sim \mathbf{c}$.

It is easy to see that a set of non-ambiguous haplotypes consists of non-intersecting compatible sets and certain haplotype block identifies each set. One can define corresponding haplotype blocks for each region $[s, e]$, but this leads to over-fitting. For example, consider the dataset with 200 haplotypes that covers over 10,000 markers. Approximately 200 markers are required to identify each haplotype and thus without constraints the optimal partition will be trivial. Because of that a valid haplotype block must have support over certain threshold and a valid block certain coverage.

Definition 5. Support $\text{supp}(\mathbf{a}_s)$ is the number of non-ambiguous haplotypes that are compatible with haplotype block $\mathbf{a}_s$. The haplotype block $\mathbf{a}_s$ is valid if $\mathbf{a}_s$ is among the observed data and $\text{supp}(\mathbf{a}_s) \geq \tau$. The block is valid if summary support of valid haplotype blocks covers over $\alpha\%$ observations.

Finding out the smallest set of ambiguous haplotypes can be tricky in theory. In practice, it is sufficient to start eliminating haplotypes with highest count of missing values and continue until all ambiguouties are resolved.

Definition 6. An indicator set of $[i, j]$ is the set of markers that identify haplotype blocks with summary support over $\alpha\%$ of all haplotypes[2]. Let $\kappa(i, j)$ denote the minimal size of the indicator set.

The definition allows misclassification, because some haplotypes are unambiguous. An alternative definition is based on haplotype diversity [11]. Then identified haplotypes must capture variation of the block. As the resulting segmentations are similar [47], we omit details. Due to the over-fitting constraint, the explicit form of the cost function is

$$f(i,j) = \begin{cases} \big(\kappa(i,j), 1\big), & \text{if } [i,j] \text{ is a valid block,} \\ (+\infty, 1), & \text{otherwise.} \end{cases}$$

Finding the minimal marker set of a block is $\mathcal{NP}$-hard task, since for zero-one markers it coincides with minimum test-set problem. Fortunately, if the number of valid haplotypes is small, the enumeration over all possible variants is feasible. Let the block length be l and number of haplotypes q. Then at most $q - 1$ markers will identify haplotype blocks and we must test all subsets with size less than q. The resulting complexity is polynomial in l

$$\binom{l}{1} + \binom{l}{2} + \cdots + \binom{l}{q-1} = \mathcal{O}(l^{q-1})$$

However, the number of variants grows drastically, when l is big. For example $l = 100$ and $q = 6$ yields $7.9 \cdot 10^7$ variants. Thus big blocks with relatively few haplotypes can cause algorithm to slow down.

The task of finding the minimal indicator set can be restated as set-cover problem [6][3]. Each marker separates several haplotype blocks. Let $\mathcal{I}$ be the

[2]Some sources [34, 47] require that $\alpha\%$ of unambiguous haplotypes are covered.

[3]The citation gives wrong impression

set of all different haplotype pairs then we can define $|I| \times l$ zero-one matrix A, where

$$a_{ij} = \begin{cases} 1, & \text{if the haplotype pair } i \text{ has different values of the } j\text{th marker,} \\ 0, & \text{otherwise.} \end{cases}$$

A proper marker set must separate all haplotype pairs. The corresponding constraint $A\boldsymbol{x} \geq \mathbf{1}$, $x_j \in \{0, 1\}$ is linear. Since the cost function $c = \mathbf{1} \cdot \boldsymbol{x}$ is also linear, we get a classical setting of set-cover problem, which can be tackled with generic integer linear programming methods [37, 43] and other more specific exact and approximate methods (see [20, 10, 32]. Another source for optimisation is to use overlappings between blocks. More precisely, if $[a, b] \subseteq [c, d]$ then $\kappa(a, b) \leq \kappa(c, d)$.

The statistical inference methods for haplotype blocks specify first a probabilistic measurement model. The haplotype blocks are inferred via maximum likelihood or minimum description length principle. One possible approach is to specify error mechanism as multivariate Bernoulli distribution and choose model with highest likelihood $\mathcal{L}_0$. But this causes over-fitting, therefore we must either limit possible number of haplotype blocks or use penalised likelihood. However, this could lead to a model far from reality and it is reasonable to consider all blocks with likelihood $\mathcal{L} < \alpha\mathcal{L}_0$ invalid.

The approach is very close to MDL methods and same techniques like EM-algorithm and clustering can be used for determination of haplotype blocks. However, we still get segmentation that requires minimal number of tag SNPs.

Robustness Genotyping errors consist of missing and incorrect markers. Elimination of ambiguous haplotypes solves the problem with missing values. True measurement errors can cause erratic and hide rare haplotypes. The latter is unavoidable, whereas the haplotype support threshold can eliminate most of errors. The threshold τ should be higher than μm, where μ is error probability of a single marker. On the other hand, too high τ will filter out rare haplotype blocks. Due to the small sample size current studies [34, 47] use the lowest threshold $\tau = 2$.

The threshold of summary cover α quantifies how far is the discrete model from true distribution. High values of α make the method more sensible to errors and increase the number of tag markers. But low values allow large deviations and make the haplotype model less powerful (see [45] for additional details).

Surprisingly enough, a sub-population may require more tag SNPs. First some rare haplotypes, refuted globally by threshold τ, may be frequent enough in the sub-population. Secondly, misclassified haplotype blocks may be unevenly represented and thus the global segmentation may be invalid.

Locality The quadratic complexity of dynamic programming algorithm favours divide-and-conquer technique. When the haplotypes are divided into k equal regions, the overall running-time reduces k times. More importantly, we can quantify the maximum number of additional markers.

Lemma 1. *Let u be the number of tag SNPs of the optimal segmentation and u' the number of tag SNPs of the optimal segmentation with the border after the kth marker. Then*

$$u' - u \leq \max_{(i,j)\in\mathcal{I}} \kappa(i,k) + \kappa(k+1,j) - \kappa(i,j) \leq q_{max},$$

where q_{max} is the maximal number of haplotypes and

$$\mathcal{I} = \{(i,j) \mid i \leq k < j \text{ and a block } [i,j] \text{ is valid and indivisible}\}.$$

Proof. Consider the optimal segmentation. If k collides with the border then $u' = u$. Otherwise, we get plausible segmentation by adding border after k. The inequality is just the upper bound of possible growth. Clearly, the block must be indivisible that is further splitting will increase the number of required markers. □

Alternative settings The number of tag SNPs is usually limited so a more natural objective is to maximise genome coverage with fixed number of markers. We need another additive cost function $\ell(\mathcal{B}) = \ell(s_1, e_1) + \cdots + \ell(s_r, e_r)$, where ℓ quantifies the span of a block. The simplest form is $\ell(i,j) = j - i + 1$, but more complex ones quantify the block span in base-pairs or even penalise introns, exons and other control structures differently.

Definition 7. The fixed tag SNP problem [48] is following. Find a segmentation $\mathcal{B}$ such that $f(B) \leq u$ and maximises $\ell(\mathcal{B})$.

Let S_{jk} be the maximal span of the segmentation in region $[1,j]$ that has at most k tag markers. The segmentation can have two possible configurations. In the first the last block does not contain the end marker and thus $S_{jk} \geq S_{j-1,k}$. Alternatively, the last block is $[i,j]$ and thus $S_{jk} \geq S_{i-1,k-f(i,j)} + \ell(i,j)$. This leads to the recursive equation

$$S_{j,k} = \max\left\{S_{j-1,k}, \max_{i<j}\big(S_{i-1,k-f(i,j)} + \ell(i,j)\big)\right\},$$

where $f(i,j) = \infty$ for invalid blocks. Natural boundary conditions are

$$S_{0,k} = \begin{cases} 0, & \text{if } k \geq 0, \\ -\infty, & \text{if } k < 0. \end{cases}$$

The complexity of a dynamic programming is $\mathcal{O}(lnu)$, where l be the maximal length of the valid block. In other words, the running-time is magnitudes longer than in the simple setting.

Another compelling task is to find partition that minimises the number of tag marker with respect to the fixed span.

Definition 8. The fixed genome coverage problem [48] is following. Find segmentation $\mathcal{B}$ such that $\ell(\mathcal{B}) \geq \ell_0$ and $f(\mathcal{B})$ is minimal.

Here again, the last block of the optimal partition may be included or excluded. As a result, a straightforward algorithm requires $n \times \ell_0$ matrix for storing intermediate results, since the algorithm must keep track of excluded markers. It is natural to allow ℓ_0 be a fraction of n, but this leads to cubic complexity $\mathcal{O}(n^3)$.

Zhang [48] used an alternative parametric cost function to reduce complexity. Let $\mathcal{B}$ and $\mathcal{E}$ be the sub-segmentation of included and excluded blocks and the parameter λ exclusion cost. Then the cost of the segmentation is $f^* = f(\mathcal{B}) + \lambda\ell(\mathcal{E})$. Parametric sequence alignment uses analogous cost function and therefore many algorithms and theoretical results coincide with results [42, 23].

Note that optimal segmentation $\mathcal{B}, \mathcal{E}$ has minimal value of $f(\mathcal{B})$ over all segmentations that exclude less than $\ell(\mathcal{E})$. Clearly, if $\ell(\mathcal{E}') \leq \ell(\mathcal{E})$ and $f(\mathcal{B}') < f(\mathcal{B})$, we get a contradiction $f(\mathcal{B}') + \lambda\ell(\mathcal{E}') < f(\mathcal{B}) + \lambda\ell(\mathcal{E})$. Moreover, the segmentations $\mathcal{B}_1, \ldots, \mathcal{B}_k$ that correspond to the growing λ have a special structure. The span of excluded regions decreases strictly and the number of tag markers increases strictly in the sequence. Shortly put, the sequence provides solutions for all fixed genome coverage problems.

For each λ we have dynamic programming problem

$$S_j = \max\left\{\max_{i<j}\bigl(S_{i-1} + \lambda\ell(i,j)\bigr), \max_{i<j}\bigl(S_{i-1} + f(i,j)\bigr)\right\}.$$

For pre-computed values of f and ℓ, the naive implementation has complexity $\mathcal{O}(n^2)$, but $\mathcal{O}(ln)$ is achievable [42, 41], where l is the maximal length of the block.

Both alternative settings give nice convex dependence between the used markers and genome coverage. Adding new markers leads to saturation, where utility of markers decreases rapidly—all long blocks have been incorporated into the model. Therefore, an approximate graph can be very useful for selecting the number of tag markers.

Genotypes *versus* haplotypes For large scale studies asymmetric methods that measure haplotypes directly are too expensive. On the other hand, complete restoration of haplotypes from genotypes can and is computationally demanding. The closer look on Algorithm 1 reveals that it does not require entire haplotypes. It is sufficient to give haplotypes and their frequencies of valid blocks. Thus segmentation algorithms and haplotype inference can be merged—blocks are iteratively increased until they are valid.

6 Minimum description length principle

Another promising way to define haplotype blocks is through minimum description principle. Like combinatorial methods the corresponding algorithms search for a partition that minimises the cost function. But the cost function has different form, the minimal number of tag markers is not guaranteed. On the other hand, the optimal partition is in a certain sense the most probable explanation

of data. Therefore, minimum description principle is more suitable for biological studies.

All three proposed methodologies [31, 2, 19] follow the same general scheme. First a probabilistic model of haplotype measurements is specified. Then priors are assigned for various parameters and last optimisation algorithm is given. In the following we try to follow the underlined road map, but before we give a brief overview of minimum description length principle.

6.1 Minimum description length

The principle of minimum description length (MDL) is a powerful and well-established criterion for model selection. In a certain sense, the MDL principle is a bridge between frequentist and Bayesian approaches in statistics. The MDL principle can be viewed as a generalisation of maximum likelihood (ML) principle. The MDL used for models with different complexity—the ML approach leads to over-fitting when models have different structure. Therefore, complexity of models is penalised by additional term. On the other hand, the penalty can be interpreted as a subjective prior of model and the MDL will coincide with the Maximum A Posteriori estimate in Bayesian statistics.

Shortly put, the description length of the model is a summary length of data description and the model description

$$\ell(\mathcal{D}, \mathcal{M}) = \ell(\mathcal{D} \mid \mathcal{M}) + \ell(\mathcal{M}).$$

The two-stage description length is simplest and perhaps most natural concept (see [24] for more detailed discussion). The data is encoded in two steps: first the parameters of model and then the data. The model specifies probability distribution for the data and therefore the data can be encoded optimally. Although the simple models have smaller penalty, their distributions are usually more apart from the actual distribution and the resulting data description is longer. The MDL makes a reasonable compromise between complexity and precision.

The description length of the second stage is determined through the data likelihood $p(\mathcal{D} \mid \mathcal{M})$. The celebrated Shannon's theorem postulates that optimal code-length of discrete data $\mathcal{D}$ with respect to model $\mathcal{M}$ is

$$\ell(\mathcal{D} \mid \mathcal{M}) = \left\lceil \log \Pr\left[\, \mathcal{D} \mid \mathcal{M} \,\right] \right\rceil.$$

Discretization allows to generalise the result to continuous data, the resulting code length, omitting constant terms and rounding, is

$$\ell(\mathcal{D} \mid \mathcal{M}) = -\mathcal{L}\left[\mathcal{D} \mid \mathcal{M}\right] = -\log p(\mathcal{D} \mid \mathcal{M}).$$

The description length of the first stage is determined by prior information. The prior information allows to pick a parameter encoding scheme that yields minimal expectation of description length. Alternatively, the coding scheme can be used without any reasoning[4].

[4]This avoids ambiguouties accompanied with Bayesian priors.

If the parameter $\theta \in \{\theta_1, \ldots, \theta_n\}$ then Laplacian principle of indifference justifies uniform encoding and $\ell(\theta) = \log n$. Continuous parameters are discretisized and encoded like discrete values. Let m be the number of samples. Then theoretical results [35, 36, 24] indicate that precision $1/\sqrt{m}$ is sufficient and the corresponding length is $\ell(\theta) = 1/2 \cdot \log m$.

Often parameter encoding corresponds to hierarchical prior. Then the parameters are encoded according to the distribution determined by hyperparameters $\mathcal{H}$ and the description length is

$$\ell(\mathcal{M}) = \ell(\mathcal{M} \mid \mathcal{H}) + \ell(\mathcal{H}).$$

Well structured models give a rise to several layers of hyper parameters.

The Bayesian viewpoint gives another interpretation of MDL principle. According to Bayes' rule

$$p(\mathcal{M} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{M})p(\mathcal{M})}{p(\mathcal{D})} \propto p(\mathcal{D} \mid \mathcal{M})p(\mathcal{M})$$

and the logarithmic conversion gives

$$\mathcal{L}(\mathcal{M} \mid \mathcal{D}) = \mathcal{L}(\mathcal{D} \mid \mathcal{M}) + \mathcal{L}(\mathcal{M}) + const.$$

If the prior is chosen $p(\mathcal{M}) \propto \exp\bigl(-\ell(\mathcal{M})\bigr)$, then the model with minimum description length has maximal posterior probability. Moreover, the normalised value

$$\Pr\left[\,\mathcal{M} \mid \mathcal{D}\,\right] = \frac{\exp\bigl(-\ell(\mathcal{D}, \mathcal{M})\bigr)}{\sum\limits_{\mathcal{M}'} \exp\bigl(-\ell(\mathcal{D}, \mathcal{M}')\bigr)}$$

gives a posterior probability of $\mathcal{M}$, provided that the number of models is finite [5]. This link between description length and probabilities allows to estimate the significance of the model.

6.2 Independent block model

The independent block model [27, 31] consists of a segmentation and haplotype patterns. A haplotype pattern specifies a haplotype block along with the probabilistic measuring mechanism. The latter makes the method robust to genotyping errors. Although the original algorithm handles only di-allelic data, it can be generalised to multi-allelic case.

Observed haplotype block $\mathbf{h}^{(k)}$ decends from basic haplotype $\mathbf{a}_s^{(k)}$. But mutation and genotyping errors can alter $\mathbf{h}^{(k)}$. Therefore, a multivariate Bernoulli distribution is associated with $\mathbf{a}_s^{(k)}$. The parameter vector

$$\boldsymbol{\theta}_s^{(k)} = \left(\theta_{s,s_k}^{(k)}, \theta_{s,s_k+1}^{(k)}, \ldots, \theta_{s,e_k}\right)$$

[5] Since the number of possible codewords if finite the number of models is also finite. The inconsistency comes from discretization of the model space.

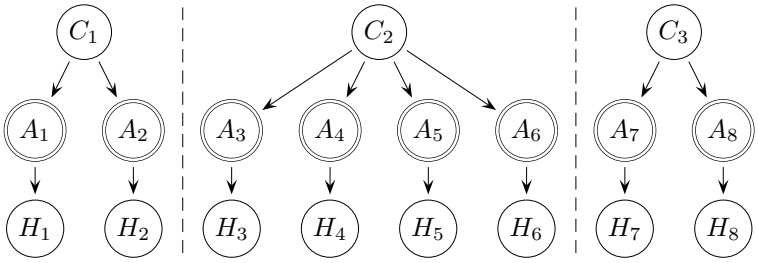

Figure 3: The internal structure of independent block model

contains probabilities of observing dominant allele in each block position. Thus the likelihood is

$$\Pr\left[\mathbf{h}^{(k)}\middle|\mathbf{a}_s^{(k)}\right] = \prod_{j=s_k}^{e_k} \left[\theta_{s,j}^{(k)}\right]^{h_j} \left[1-\theta_{s,j}^{(k)}\right]^{1-h_j}. \tag{1}$$

Let $m \times r$ matrix C consist of the class indices of the observed haplotypes. Then the conditional log-likelihood of the data is

$$\mathcal{L}\left[H\middle|\mathcal{M}\right] = \sum_{k=1}^{r}\sum_{i=1}^{m}\sum_{j=s_k}^{e_k}\left[h_{i,j}\log\theta_{c_{i,k},j}^{(k)} + (1-h_{i,j})\log(1-\theta_{c_{i,k},j}^{(k)})\right].$$

Parameter encoding The complete description of a segmentation $\mathcal{B}$ requires $r\log n$ bits. Block counts q_k require another $r\log m$ bits. Uniform prior of block numbers gives $\log q_k$ bits for each entry of C. A Bernoulli parameter requires $(\log m)/2$ bits and the overall description length is

$$\ell(\mathcal{M}) = r(\log m + \log n) + \sum_{k=1}^{r}\left[\frac{1}{2}q_k(e_k - s_k + 1)\log m + m\log q_k\right]. \tag{2}$$

Optimisation method The MDL cost function is also in additive form. Let us define block weight

$$\begin{aligned} f(s_k, e_k) = {} & \log m + \log n + m\log q_k + \frac{1}{2}q_k(e_k + s_k + 1)\log m \\ & - \sum_{i=1}^{m}\sum_{j=s_k}^{e_k}\left[h_{i,j}\log\theta_{c_{i,k},j}^{(k)} + (1-h_{i,j})\log(1-\theta_{c_{i,k},j}^{(k)})\right]. \end{aligned}$$

Then $\ell(H,\mathcal{M}) = f(s_1,e_1) + \cdots + f(s_r,e_r)$ and we can use the dynamic programming algorithm 1.

However, the function f has more complicated form than before. It is hard to derive the best model of block patters, because observations must be divided

into different clusters that minimise the the block description. The underlying problem is $\mathcal{NP}$-hard, but efficient classification algorithms provide approximate solutions. Current implementations use k-means algorithm, since the class centres of zero-one data coincide with the MLE estimate of $\boldsymbol{\theta}_s^{(k)}$ (see implementation details [31, 27]). Basically, the k-means routine is invoked several times with different cluster numbers and the best clustering is chosen. The maximum number of clusters is bounded by 10—the bigger number of haplotypes blocks have longer description and are extremely rare.

The multi-allelic data requires different clustering methodology, since Euclidean distance does not capture well similarities between categorical data. Nevertheless, a reasonable classification allows to compute the MLE estimate for parameters of multivariate multinomial distribution.

The fixed genome coverage problem is compelling also with the MDL cost function, because it allows to drop inconsistent markers. The setting and algorithm coincide with the combinatorial ones.

Computational complexity and locality The computation of $f(s_k, e_k)$ is dominated by complexity of clustering. As the complexity of k-means algorithm is $\mathcal{O}(ml)$, the overall complexity of the algorithm is $\mathcal{O}(lmn^2)$. The latter can be problem if the n is big enough.

As before, the divide-and-conquer technique decreases running-time. Moreover, we can bound the increase in description length. Unfortunately, the result does not directly quantify the difference between two models.

Lemma 2. *Let ℓ be the optimal description length and ℓ' the optimal description length with the border after the kth marker. Then the difference $\ell' - \ell \leq \log m + \log n + m \log q_{max}$, where q_{max} is the maximal number of haplotype blocks.*

Proof. The additional border after the kth marker increases only model complexity and thus the claim follows. □

Significance of the block boundaries The duality between description length and posterior probabilities allows to estimate a significance of the block boundaries. Let us consider only the segmentations with optimally tuned parameters. Let the set $\mathcal{S}_{j,j+1}$ consist of all segmentations with a boundary between the jth and $(j+1)$st marker and the set $\mathcal{S}$ of all segmentations. Then the probability

$$\Pr\left[\, \mathcal{S}_{j,j+1} \mid H, \mathcal{O}ptimal \,\right] = \frac{\sum\limits_{\mathcal{M} \in \mathcal{S}_{j,j+1}} \exp\bigl(-\ell(\mathcal{M})\bigr)}{\sum\limits_{\mathcal{M} \in \mathcal{S}} \exp\bigl(-\ell(\mathcal{M})\bigr)}.$$

The segmentations in the region $[i, j]$ can be divided into groups according the last block. Therefore the summary probability $Q(i, j)$ of all segmentations in the region $[i, j]$ decomposes

$$Q(i, j) = \sum_{i \leq k \leq j} Q(i, k-1) \exp(-f(k, j))$$

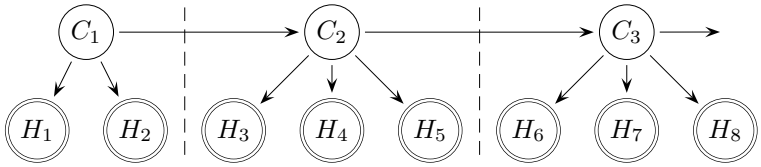

Figure 4: Internal structure of first order Markov model

and we can calculate the significance of block borders via dynamic programming

$$\Pr\left[\,\mathcal{S}_{j,j+1} \mid H, \mathcal{O}ptimal\,\right] = \frac{Q(1,j)Q(j+1,n)}{Q(1,n)}.$$

Robustness and missing data The measuring model makes the algorithm remarkably tolerant to noise. Experiments [31] show that $5-10\%$ of noise decreases only the significance of block boundaries but does not significantly change the segmentation.

The measuring mechanism allows to incorporate missing values into the data without imputation. In di-allelic case, the missing values can be encoded by numbers ranging $[0,1]$ depending on the certainty. Then the likelihood formula (1) must be rescaled, but rescaling yields a constant in the cost function which can be ignored. Also, the Euclidean distance remains appropriate dis-similarity measure and cluster centres and the Bernoulli parameters still coincide.

6.3 First order Markov model

The hidden Markov model was first used for haplotype block inference already by Daly et al. [14], where the sequence of block assignments of SNPs was considered a Markov chain. The methodology proposed by Anderson and Novembre [2] differs—they considered the sequence of blocks instead. The consecutive blocks are not completely independent and therefore first order Markov chain is more adequate than independence assumption. The second difference is missing error model—all haplotypes are assumed to be exact without genotyping errors. Third major difference is complex and somewhat cumbersome hierarchical prior.

The model consist of segmentation $\mathcal{B}$, induced haplotypes and description of the Markov chain. The haplotype blocks $\mathbf{a}_s^{(k)}$ are all observed marker sequences in the kth block. The Markov chain is determined by initial marginal probabilities

$$\theta_s^{(1)} = \Pr\left[\,\mathbf{a}_s^{(1)}\,\right], \qquad s = 1,\ldots,q_1$$

and transition probabilities

$$\psi_{s\to t}^{(k)} = \Pr\left[\,\mathbf{a}_t^{(k)} \mid \mathbf{a}_s^{(k-1)}\,\right], \qquad s = 1,\ldots,q_{k-1}, t = 1,\ldots,q_k.$$

Other marginal probabilities can be expressed in terms of initial probabilities and transition probabilities.

Let $H^{(k)}$ be the sub-matrix that corresponds to the kth block, $X_s^{(k)}$ be the count of $\mathbf{a}_s^{(k)}$ and $Z_{s\to t}^{(k)}$ the count of pairs $\mathbf{a}_s^{(k-1)}, \mathbf{a}_t^{(k)}$. Then the complete log-likelihood of the data is a sum of

$$\mathcal{L}\left[H^{(1)}\middle|\mathcal{M}\right] = \sum_{s=1}^{q_1} X_s^{(1)} \log \theta_s^{(1)},$$
$$\mathcal{L}\left[H^{(k)}\middle|H^{(k-1)},\mathcal{M}\right] = \sum_{s=1}^{q_{k-1}}\sum_{t=1}^{q_k} Z_{s\to t}^{(k)} \log \psi_{s\to t}^{(k)}.$$

Priors and parameter encoding The haplotype block are modelled by multivariate Bernoulli distribution with the parameter vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$ that consists of probabilities of dominant alleles. The prior avoids over-fitting since blocks with many haplotypes get high penalty. The summary log-likelihood of haplotype blocks is a sum of terms

$$\mathcal{L}\left[\mathbf{a}_1^{(k)}, \ldots, \mathbf{a}_{q_k}^{(k)}\middle|\boldsymbol{\phi}\right] = \sum_{s=1}^{q_k}\sum_{j=s_k}^{e_k}\left[a_{sj}^{(k)} \log \phi_j + (1 - a_{sj}^{(k)}) \log(1-\phi_j)\right].$$

The transition matrix $\psi^{(k)}$ gets additional penalty according to the number of different entries [6]. Let $\Delta^{(k)}$ be the indicator matrix of $\psi^{(k)}$, more precisely let zero represent dependence in the row

$$\delta_{s\to t}^{(k)} = \delta_{s\to u}^{(k)} = 0 \qquad \Leftrightarrow \qquad \phi_{s\to t}^{(k)} = \phi_{s\to u}^{(k)}$$

and one independence. To encode the sth row of $\Delta^{(k)}$, we first indicate how many ones are in the row. This requires 1 bit if all elements are zeros and $\log q_k$ otherwise. It is easy to see that a configuration of $z_s^{(k)}$ ones require $\log \binom{q_k}{z_s^{(k)}}$ bits [7]. The overall description length of $\Delta^{(k)}$ is

$$\ell\big(\Delta^{(k)}\big) = \sum_{s=1}^{q_{k-1}} \operatorname{sign}\big(z_s^{(k)}\big) \log q_{k-1} + \sum_{s=1}^{q_{k-1}}\sum_{i=0}^{z_s^{(k)}-1}\left[\log(z_s^{(k)} - i) - \log(i+1)\right].$$

The sth row of $\psi^{(k)}$ requires $q_k + 1 - z_s^{(k)}$ entries with the optimal coding length $(\log X_s^{(k-1)})/2$, this gives

$$\ell\big(\psi^{(k)}\big) = \frac{1}{2}\sum_{s=1}^{q_{k-1}} (q_k - z_s^{(k)} + 1) \log X_s^{(k-1)}$$

[6] I cannot understand why they do not use simple penalty function like summary square difference from the weighted row average.

[7] The original article [2] contains an error here—the number of possibilities is essentially binomial coefficient and not a falling factorial!

Additionally, the initial probabilities require[8] $\ell(\boldsymbol{\theta}) = q_1/2 \cdot \log m$ bits.

Let $Z^{(k)}_{s\to 0}$ be the summary count of all pairs that correspond to indicators $\delta^{(k)}_{s\to t} = 0$ and $w^{(k)}_s$ the count of $\delta^{(k)}_{s\to t} = 0$. Then the maximal likelihood estimate of transition matrix with respect to fixed configuration yields[9]

$$\psi^{(k)}_{s\to t} = \begin{cases} \frac{Z^{(k)}_{s\to t}}{X^{(k)}_s}, & \text{if } \delta^{(k)}_{s\to t} = 1, \\ \frac{Z^{(k)}_{s\to 0}}{w^{(k)}_s X^{(k)}_s}, & \text{otherwise.} \end{cases}$$

The result follows directly from ML estimate of multinomial distribution. Since there are exponential number of different assignments of $\Delta^{(k)}$, the minimising solution is obtained through hill-climbing algorithm. Clearly, the ML estimate for initial probabilities is $\theta^{(k)}_s = X^{(k)}_s/m$.

If we assume that haplotype blocks are drawn only once, we get the penalty of hyper-parameters

$$\ell(\boldsymbol{\phi}) = \sum_{k=1}^{r} (e_k - s_k + 1) \log q_k.$$

Finally, the exact description length of segmentation $\mathcal{B}$ is

$$\ell(\mathcal{B}) = \log n + \log \binom{m-1}{r}.$$

Optimisation method The cost function is not completely additive

$$\ell(H, \mathcal{M}) = \ell(\mathcal{B}) + f_2(e_1) + \sum_{k=2}^{r} f(s_{k-1}, e_{k-1}, e_k),$$

where

$$f_1(e_1) = \ell(\mathrm{H}^{(1)}) + \ell(\mathbf{a}^{(1)}_1, \ldots, \mathbf{a}^{(1)}_{q_1}) + \ell(\boldsymbol{\theta}^{(1)}, \boldsymbol{\phi}^{(1)}),$$
$$f_2(s_{k-1}, e_{k-1}, e_k) = \ell(\mathrm{H}^{(k)} \mid \mathrm{H}^{(k-1)}, \mathcal{M}) + \ell(\mathbf{a}^{(k)}_1, \ldots, \mathbf{a}^{(k)}_{q_k}) + \ell(\psi^{(k)}, \Delta^{(k)}, \boldsymbol{\phi}^{(k)}).$$

But if we specify the number of blocks r forehead, we can use dynamic programming. The optimal segmentation must have optimal prefix but the beginning of last block is also important. So the dependence between adjacent blocks forces two-dimensional cost matrix S and cubic time-complexity.

The latter can be tackled with heuristic reduction. Although the exact recursion formula is

$$S_{jk} = \min_{i<j} [S_{ij} + f(i, j, k)],$$

[8] Here, differently from original article, we do not encode marginal frequencies for all steps, since the penalty of $\psi^{(k)}$ is already covered. Also, the question about optimal prior for marginal frequencies disappears.

[9] Here again, the result is more evidently presented than in original article.

the optimal solution is likely to pass through column winners $S_{ij} = \min_{i<j} S_{ij}$, since the differences between $f(i,j,k)$ are small compared with differences among columns. The approximate algorithm looks only the column winners and the complexity becomes quadratic. For more details see the appendix B of [2].

The weight of $\ell(\mathcal{B})$ is small, but nevertheless there might be several iterations of dynamic algorithm until the solution is consistent to the pre-specified value of r. But when the description length of the segmentation is approximated $\ell(\mathcal{B}) = r \log n$, then the cost function becomes additive and one iteration provides an answer. Also, the significance of the block boundaries becomes computable, though the corresponding expression is more complex compared with the independent block model.

Robustness and missing data The method is clearly sensitive to errors, since there is no built-in flexibility. For example, few errors in large blocks will break them into smaller sub-blocks. Consider a block with two haplotypes $1 \ldots 1$ and $0 \ldots 0$, when a single genotyping error leads to three haplotypes $1 \ldots 1$, $0 \ldots 0$ and $0 \ldots 010 \ldots 0$. Then the cost of a single block parameters consists of

$$\begin{aligned}
&\ell(\mathcal{B}^{(k)}) \approx \log n, \qquad \ell(\boldsymbol{\phi}^{(k)}) = l \log 3 \\
&\ell(\mathbf{a}_1^{(k)}, \ldots, \mathbf{a}_3^{(k)}) = 3(l-1)\log 2 + 3\log 3 - 2\log 2
\end{aligned}$$

and cost of matrices $\psi^{(k)}$, $\Delta^{(k)}$. If the error is encapsulated into a sub-block, then the cost of parameters reduces

$$\begin{aligned}
&\ell(\mathcal{B}^{(k)}, \ldots, \mathcal{B}^{(k+2)}) \approx 3 \log n, \\
&\ell(\boldsymbol{\phi}^{(k)}, \ldots, \boldsymbol{\phi}^{(k+2)}) = l \log 2 \\
&\ell(\mathbf{a}_1^{(k)}, \ldots, \mathbf{a}_2^{(k+2)}) = 2l \log 2.
\end{aligned}$$

New transformation matrices add additional complexity, but the maximum cost is $4 \log m + 2$, if there is no packing and precision is $(\log m)/2$.

The difference between the probabilities of the Markov chains is negligible, because we introduced two rare rules $0 \ldots 0 \to 1$ and $1 \to 0 \ldots 0$. Consequently, the block is divided if inequality holds

$$2 \log n + 4 \log m + 2 < l \log 3 - 5 \log 2 + 3 \log 3.$$

The sensitivity can cause big variance in results, if missing data is handled by imputation. Compatibility classes of haplotypes allow to bypass the problem and a reasonable support threshold for haplotypes could make the method more robust.

Complexity and locality The cubic complexity makes the exact algorithm intractable for large sequences, but the quadratic approximation algorithm relives the problem. Unfortunately, there is no well-established similarity measure for segmentations. Therefore, we cannot quantify the cost of approximation. On

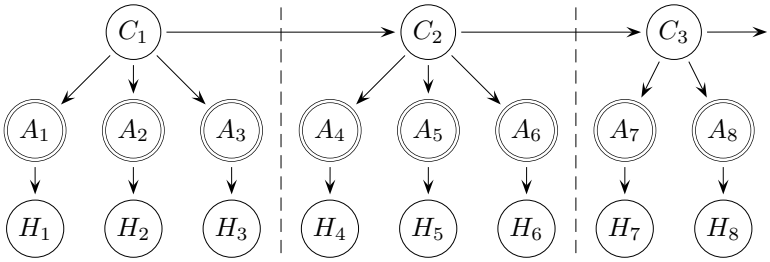

Figure 5: The internal structure of averaged first order Markov chain

the other hand, results [2] suggest that the difference in description lengths is small and the borders are similar. Still we need more formal argumentation augmented with practical results. The same lack of argumentation arises, if we compare the independent block model and the first order Markov chain.

Again with some effort one can find cost of dividing haplotypes into several regions. However, the result is not so neat as before.

6.4 Averaged first order Markov chain

Another and technically even more demanding approach [19] uses both Markov chain and stochastic mutation mechanism. The main difference compared with other two MDL methods is averaging over all possible block assignments. In other words, instead of two-stage coding we use mixture MDL principle to evaluate models (see [24] for additional discussion). The segmentation, haplotype blocks and various transition probabilities are treated differently from concrete block assignments. The former is the aim of the inference, whereas the latter corresponds to latent variables and is averaged out. A solution is the model that on average has greater support of observed data.

The segmentation model $\mathcal{B}$ determines the size of blocks and the Markov chain $C_1, \ldots, C_r$ models block assignments. The value of C_k determines the haplotype block $\mathbf{a}_s^{(k)}$. But the mutation layer can change haplotype block $\mathbf{a}_s^{(k)}$ to a different observation $\mathbf{h}_s^{(k)}$. The model is captured in Figure 5 that represents corresponding Bayesian network. As before, $\psi_{s\to t}^{(k)}$ denotes probability that block $\mathbf{a}_s^{(k-1)}$ is followed by $\mathbf{a}_t^{(k)}$ and $\theta_s^{(1)}$ denotes the marginal probability of $\mathbf{a}_s^{(1)}$. The constrained mutation probabilities

$$\mu_{a\to h}^{(j)} = \Pr[\, H_j = h \mid A_j = a \,] \in [\mu_{min}, \mu_{max}], \qquad j = 1, \ldots, n.$$

make the model more robust. Original article suggest the interval $[10^{-3}, 10^{-6}]$, but due to the measurement errors $[10^{-3}, 0.03]$ is more appropriate.

A genotype data can be used directly without the haplotype inference phase, since by duplicating the structure we get model of genotype.

The evidence of concrete model is obtained through summing over all possible assignments of haplotype blocks

$$\Pr[\, H \mid \mathcal{M}\,] = \sum_{i=1}^{m} \sum_{c_1,\ldots,c_r} \Pr[\, c_1,\ldots,c_r\,] \prod_{k=1}^{r} \Pr\Big[\, \mathbf{a}_{c_k}^{(k)} \to \mathbf{h}_i^{(k)} \,\Big]. \qquad (3)$$

The efficient calculation is computationally demanding and is done by bucket elimination technique[15]. Nevertheless, the summation spoils linearity of log-likelihood and thus efficient dynamic programming methods in simple form are impossible.

Priors and parameter encoding Somewhat surprisingly, original article [19] assigns equal priors to all segmentations. Recall that the previous two models favoured segmentations with longer blocks. Also, difference in haplotype counts q_k are is ignored and the effective description length is determined by descriptions of $\boldsymbol{\theta}^{(1)}$,$\psi^{(1)},\ldots,\psi^{(r)}$ and haplotype blocks.

Initial probability vector require $\frac{q_1-1}{2}\log m$ bits and the transition matrix $\psi^{(k)}$ requires $\frac{q_k-1}{2}\sum_{s=1}^{q_{k-1}} \log m \cdot \theta_s^{(k-1)}$, where $\theta_s^{(k-1)}$ are marginal probabilities of the $(k-1)$st block. The haplotype blocks are assumed to be drawn form multivariate multinomial distribution, where all parameters are considered equiviprobable. Thus the ML estimate of hyper parameters leads to

$$\mathcal{L}\big[\mathbf{a}_s^{(k)}\big] = \sum_{j=s_k}^{e_k} \log f_j\big(a_{sj}^{(k)}\big), \qquad f_j(a) = \frac{1}{q_k}\#\big\{s:\ a_{sj}^{(k)} = a\big\}.$$

Optimisation method The averaging makes the optimisation harder, since we cannot convert sum (3) into product or sum of sub-functions that depend only on few block borders. Therefore, we cannot use dynamic programming at least in the present form. Currently employed optimisation technique is basically local greedy search. It consists of three different sub-steps: addition, nugging and deletion of block boundaries.

During an addition step a new boundary is added, if the resulting description is shorter than previous. The nugging step shifts "slightly" boundaries and if necessary corrects the estimate. The same applies to the deletion step. All substeps try to find the minimising points in the neighbourhood. To reduce the amount of computation, only the parameters of altered blocks are recalculated. First, haplotype blocks and mutation probabilities are optimised locally block by block. Then the parameters of the Markov chain inferred. Still the optimisation is computationally very demanding and additional shortcuts are used to speed up the process.

The averaging over all possible block assignments makes exact estimation of significance of haplotype boundaries intractable.

Robustness and missing data The model is essentially robust, if we allow large $0.03-0.1$ mutation probabilities. However, the heuristic greedy optimisa-

tion method itself can cause variance in results. Therefore, practical experiments are required to determine whether the algorithm behaves consistently.

The missing values are not a problem, because they can be integrated out from the likelihood formula.

7 Discussion and concluding remarks

Roughly, all haplotype inference methods can be divided into three classes: methods that use marker pairs, combinatorial methods and MDL methods. Marker pairs provide extremely local information and thus cannot provide globally optimal segmentations. Nevertheless, the corresponding statistics like LD measures and the FGT test are valuable for evaluating quality of blocks.

The main advantage of combinatorial methods is clear objective—minimal number of tag SNPs. The corresponding cost function coincides with the real cost of large scale association studies. Whether the block boundaries have biological reasoning is another issue. The main disadvantage is *ad hoc* criterion of valid block—the coverage and diversity are brutal and oversimplified measures from the biological and statistical viewpoint. Alternative stochastic criterion that uses probabilistic measuring model and constrained ML or MDL principle could provide well-grounded statistical approach, but still preserve the original objective.

Still the combinatorial approach is reasonably robust and flexible, besides handles missing data without complex imputation routines. The thresholds allow to control both robustness and accuracy. The search of test markers can occasionally slow down the algorithm, but this is common for all segmentation algorithms that provide tag SNPs.

The problem of optimal marker set is subject of independent interest. The problem can be restated as a set-cover problem that has been studied for years. Therefore, results of linear integer programming and specific set-cover solvers can reduce complexity of tagging phase and simplify the haplotype inference.

The MDL methods have solid statistical grounding and provide also good results in practice. However, the optimal solution does not guarantee the minimal number of tag SNPs and therefore we have a discrepancy between obtained and desired objectives. Another trade-off here is between computational complexity and statistical accuracy. Though more sophisticated methods based on the Markov chains can be statistically more precise, they also have higher computational complexity. Unfortunately, there is no well-established similarity measure between segmentations and thus we cannot evaluate the cost of trade-offs. But the independent block model seems to be good compromise between accuracy and complexity.

Depending on objectives MDL methods can be preferred to combinatorial ones and vice versa. But the concordance between economical utility and cost function makes the combinatorial methods more appealing for large scale studies. However, this does not diminish value of other methods, since economy in genotyping is not the only objective.

References

[1] Joshua M. Akey, KunZhang, Momiao Xiong, Peter Doris, and Li Jin. The effect that genotyping errors have on the robustness of common linkage-disequilibrium measures. *The American Journal of Human Genetics*, 68(6):1447–1456, Jun 2001.

[2] Eric C. Anderson and John Novembre. Finding haplotype block boundaries by using the minimum-description-length principle. *The American Journal of Human Genetics*, 73(2):336–354, Aug 2003.

[3] Lidia A. Zaozerskaya Anton V. Eremeev, Alexander A. Kolokolov. Hybrid algorithm for set covering problem. Available online, 200?

[4] H. I. Avi-Itzhak, X. Su, and F. M. De La Vega. Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. In *Pacific Symposium on Biocomputing 2003*, pages 446–477, 2003.

[5] Patrizia Beraldi and Andrzej Ruszczynski. The probabilistic set-covering problem. *Operations Research*, 50(6):956–967, Nov 2002.

[6] Koen M. J. De Bontridder, B. Haldersson, M. Haldersson, C. A. J. Hurkens, Jan K. Lenstra, R. RAVI, and Leen Stougie. Approximation algorithms for the test cover problem. Technical Report 10, Technische Universiteit Eindhoven, 2002. SPOR report.

[7] Koen M. J. De Bontridder, B. J. Lageweg, Jan K. Lenstra, James B. Orlin, and Leen Stougie. Branch-and-bound algorithms for the test cover problem. In *ESA 2002*, volume 2461 of *Lecture Notes in Computer Science*, pages 223–?? Springer, 2002.

[8] Alberto Caprara. Algorithms based on lp relaxations for combinatorial optimization problems. Technical Report OR-97-11, DEIS-Operations Research Group, 1997. Summary of the Ph.D Thesis.

[9] Alberto Caprara, Matteo Fischetti, and Paolo Toth. A heuristic method for the set covering problem. In *IPCO 1996*, volume 1084 of *Lecture Notes in Computer Science*, pages 72–84. Springer, 1995.

[10] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Algorithms for the set covering problem. Technical Report OR-98-3, DEIS-Operations Research Group, 1998.

[11] David Claiton. Choosing set of haplotype tagging snps from a larger set of diallelic loci, 2001. Available online `http://www.nature.com/ng/journal/v29/n2/extref/ng1001-233-S10.pdf`.

[12] A. G. Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7:111–122, 1990.

[13] Francis S. Collins, Michael Morgan, and Aristides Patrinos. The human genome project: lessons from large-scale biology. *Science*, 300(5617):286–290, Apr 2003.

[14] Mark J. Daly, John D. Rioux, Stephen F. Schaffner, Thomas J. Hudson, and Eric S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, Oct 2001.

[15] Rina Dechter. Bucket elimination: A unifying framework for probabilistic inference. In E. Horvitz and F. Jensen, editors, *12th Conference on Uncertainty in Artificial Intelligence*, pages 211–219, Aug 1996.

[16] Laurent Excoffier and Montgomery Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–927, 1995.

[17] Daniele Fallin1 and Nicholas J. Schork. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *The American Journal of Human Genetics*, 67:947–959, 2000.

[18] Stacey B. Gabriel, Stephen F. Schaffner, Huy Nguyen, Jamie M. Moore, Jessica Roy, Brendan Blumenstiel, John Higgins, Matthew DeFelice, Amy Lochner, Maura Faggart, Shau Neen Liu-Cordero, Charles Rotimi, Adebowale Adeyemo, Richard Cooper, Ryk Ward, Eric S. Lander, Mark J. Daly, and David Altshuler. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225–2229, Jun 2002.

[19] Gideon Greenspan and Dan Geiger. Model-based inference of haplotype block variation. In *Proceedings of the seventh annual international conference on Computational Molecular Biology*, pages 131–137. ACM, 2003.

[20] Tal Grossman and Avishai Wool. Computational experience with approximation algorithms for the set covering problem. *The European Journal of Operational Research?*, 101(1):81–92, 1997.

[21] International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–933, Feb 2001.

[22] Dan Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *Journal of Computational Biology*, 8(3):305–323, 2001.

[23] Dan Gusfield, K. Balasubramanian, and Dalit Naor. Parametric optimization of sequence alignment. *Algorithmica*, 12(4/5):312–326, 1994.

[24] Mark H. Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[25] Developing a haplotype map of the human genome for finding genes related to health and disease. Available online `http://www.genome.gov/10001665`, Jul 2001. See also `http://hapmap.cshl.org` and `http://www.hapmap.org`.

[26] 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(1):147–164, Sep 1985.

[27] M. Koivisto, M. Perola, T. 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 *Pacific Symposium on Biocomputing 2003*, pages 502–513, 2003.

[28] Leonid Kruglyak and Deborah A. NickersonLeonid. Variation is the spice of life. *Nature Genetics*, 27(3):234–236, Mar 2001.

[29] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1993.

[30] Stan Liao and Srinivas Devadas. Solving covering problems using lpr-based lower bounds. In *34th Annual Conference on Design Automation*, pages 117–120. ACM, 1997.

[31] H. Mannila, M. Koivisto, M. Perola, T. Varilo, W. Hennah, J. Ekelund, M. Lukk, L. Peltonen, and E. Ukkonen. Minimum description length block finder, a method to identify haplotype blocks and to compare the strength of block boundaries. *The American Journal of Human Genetics*, 73(1):86–94, Jul 2003.

[32] Carlo Mannino and Antonio Sassano. Solving hard set covering problems. *Operations Research Letters*, 18:1–5, 1995.

[33] Tianhua Niu, Zhaohui S. Qin, Xiping Xu, and Jun S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *The American Journal of Human Genetics*, 70:157–169, 2002.

[34] Nila Patil, Anthony J. Berno, David A. Hinds, Wade A. Barrett, Jigna M. Doshi, Coleen R. Hacker, Curtis R. Kautzer, Danny H. Lee, Claire Marjoribanks, David p. McDonough, Bich T. Nguyen, Michael C. Norris, John B. Sheehan, Naiping Shen, David Stern, Renee P. Stokowski, Daryl J. Thomas, Mark O. Trulson, Kanan R. Vyas, Kelly A. Frazer, Stephen P. Fodor, and David R. Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(5547):1719–1723, Nov 2001.

[35] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[36] Jorma Rissanen. Stochastic complexity in learning. In Paul M. B. Vitányi, editor, *EuroCOLT*, volume 904 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 1995.

[37] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.

[38] Matthew Stephens, Nicholas J. Smith, and Peter Donnelly. A new statistical method for haplotype reconstruction from population data. *The American Journal of Human Genetics*, 68:978–989, 2001.

[39] Ning Wang, Joshua M. Akey, Kun Zhang, Ranajit Chakraborty, and Li Jin. Distribution of recombination crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *The American Journal of Human Genetics*, 71:1227–1234, 2002.

[40] Gregory R. Warnes. *R packages: The genetics package*, 2003.

[41] Michael S. Waterman. *Introduction to Computational Biology: Maps, sequences and genomes*. CRC Press, 1995.

[42] Michael S. Waterman, M. Eggert, and E. Lander. Parametric sequence comparisons. In *Proc. Natl. Acad. Sci. USA*, volume 89, Jul 1992.

[43] Laurence A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.

[44] Mutsunori Yagiura, Masahiro Kishida, and Toshihide Ibaraki. A 3-flip neighborhood local search for the set covering problem. Available online `http://citeseer.nj.nec.com/559607.html`, 2000. Presented at ISMP2000.

[45] Kui Zhang, Peter Calabrese, Magnus Nordborg, and Fengzhu Sun. Haplotype block structure and its applications to association studies: Power and study designs. *The American Journal of Human Genetics*, 71:1386–1394, 2002.

[46] Kui Zhang, Ting Chen, Michael S. Waterman, and Fengzhu Sun. A set of dynamic algorithms for haplotype block partition and tag snp selection via haplotype data and genotype data. In *Proceedings of DIMACS workshop on SNP*, 2003.

[47] Kui Zhang, Minghua Deng, Ting Chen, Michael S. Waterman, and Fengzhu Sun. A dynamic programming algorithm for haplotype block partitioning. In *Proc. Natl. Acad. Sci. USA*, volume 99 of *Applied mathematics*, pages 7335–7339, May 2002.

[48] Kui Zhang, Fengzhu Sun, Michael S. Waterman, and Ting Chen. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. *The American Journal of Human Genetics*, 73(1):63–73, Jul 2003.

[49] Kui Zhang, Michael S. Waterman, and Ting Chen. Dynamic programming algorithms for haplotype block partitioning: applications to human chromsome 21 haplotype data. In *Proceedings of the Seventh Annual International Confertence on Research in Computational Molecular Biology (RECOMB2003)*, pages 332–340. ACM, 2003.