Hidden Markov Modelling Techniques for Haplotype Analysis*

Mikko Koivisto, Teemu Kivioja, Heikki Mannila, Pasi Rastas, and Esko Ukkonen

Department of Computer Science and HIIT Basic Research Unit
P.O. Box 68 (Gustav Hällströminkatu 2b)
FIN-00014 University of Helsinki, Finland
Firstname.Lastname@cs.helsinki.fi

Abstract. A hidden Markov model is introduced for descriptive modelling the mosaic–like structures of haplotypes, due to iterated recombinations within a population. Methods using the minimum description length principle are given for fitting such models to training data. Possible applications of the models are delineated, and some preliminary analysis results on real sets of haplotypes are reported, demonstrating the potential of our methods.

1 Introduction

Hidden Markov models (HMMs) have become a standard tool in biological sequence analysis [8,2]. Typically they have been applied to modelling multiple sequence alignments of protein families and protein domains as well as to database searching for, say, predicting genes.

In this paper we introduce HMM techniques for modelling the structure of genetic variation between individuals of the same species. Such a variation is seen in so–called haplotypes that are sequences of allelic values of some DNA markers taken from the same DNA molecule. The single nucleotide polymorphisms (SNPs) are important such markers, each having two alternative values that may occur in haplotypes of different individuals. The SNPs cover for example the human genome fairly densely. The variation of haplotypes is due to point mutations and recombinations that take place during generations of evolution of a population. Studying the genetic variations and correlating them with variations in phenotype is the commonly followed strategy for locating disease causing genes and developing diagnostic tests to screen people having high risk for these diseases.

Several recent studies have uncovered some type of block structure in human haplotype data [1,10,4,18,11,6]. However, the recombination mechanism as such does not necessarily imply a global block structure. Rather, at least in old populations one expects to see a mosaic–like structure that reflects the recombination

* A research supported by the Academy of Finland under grant 201560

S. Ben-David, J. Case, A. Maruoka (Eds.): ALT 2004, LNAI 3244, pp. 37–52, 2004.

history of the population. Assume that the haplotypes of an observed population have developed in generations of recombinations from certain 'founder' haplotypes. Then the current haplotypes should consist of conserved sequence fragments (possibly corrupted by some rare later mutations) that are taken from the founder sequences. Our goal is to uncover such conserved fragments from a sample set of haplotypes.

Unlike most earlier approaches for analyzing haplotype structures, the model class introduced here has no bias towards a global block structure. Our model is an acyclic HMM, capable of emitting equal length sequences. Each state of the model can emit a sequence fragment that is to be put into some specific location of the entire sequence. As the location is independent of the locations of other states, the madel has no global block structure. The hidden part of each state represents a conserved fragment, and the transitions between the states model the cross–overs of recombinations. We distinguish a general variant whose transition probabilities depend on both states involved, and a 'simple' variant whose transition probabilities depend only on the state to be entered.

For selecting such a HMM for a given training data we suggest a method based on the minimum description length principle (MDL) by Rissanen [12,13] which is widely used in statistics, machine learning, and data mining [9,5]. An approximation algorithm will be given for the resulting optimization problem of finding a model with shortest description in the proposed encoding scheme. The algorithm consists of two parts that are iterated alternatingly. The greedy part reduces the set of the conserved fragments, and the optimizer part uses the usual expectation maximization algorithm for finding the transition probabilities for the current set of fragments.

Our idea of block–free modeling has its roots in [17] which gave a combinatorial algorithm for reconstructing founder sequences without assuming block structured fragmentation of the current haplotypes. In probabilistic modelling and using MDL for model selection we follow some ideas of [6]. Unfortunately in the present block–free case the optimization problem seems much harder. Recently, Schwartz [14] proposed a model basically similar to our simple model variant. However, his method for model selection is different from ours. A non–probabilistic variant of our model with an associated learning problem was introduced in [7].

The rest of the paper is organized as follows. In Section 2 we describe the HMMs for modelling haplotype fragmentations. Section 3 gives MDL methods for selecting such models. Section 4 discusses briefly the possible applications of our models. As an example we analyze two real sets of haplotypes, one from lactose tolerant and the other from lactose intolerant humans. Section 5 concludes the paper.

We assume basic familiarity with HMM techniques as described e.g. in [2].

2 Fragmentation Models

2.1 Haplotype Fragmentation

We want to model by a designated HMM the haplotypes in a population of a species of interest. The haplotypes are over a fixed interval of consecutive genetic markers, say m markers numbered $1, 2, \ldots, m$ from left to right. The markers may be of any type such as biallelic SNPs (single nucleotide polymorphisms) or multiallelic microsatellite markers. The alleles are the possible alternative 'values' a marker can get in the DNA of different individuals of the same species. A biallelic SNP has two possible (or most frequently occurring) values that actually refer to two alternative nucleotides that may occur in the location of the SNP in the DNA sequence. Hence each marker i has a corresponding set A_i of possible alleles, and each haplotype over the m markers is simply a sequence of length m in $A_1 \times \ldots \times A_m$.

Our HMM will be based on the following scanario of the structure of the haplotypes as a result of the microevolutionary process that causes genetic variation within a species. We want to model the haplotypes of individuals in a population of some species such as humans. Let us think that the observed population was founded some generations ago by a group of 'founders'. According to the standard model of DNA microevolution, the DNA sequences of the current individuals are a result of iterated recombinations of the DNA of the founders, possibly corrupted by point mutations that, however, are considered rare. Simply stated, a recombination step produces from two DNA sequences a new sequence that consists of fragments taken alternatingly from the two parent sequences. The fragments are taken from the same locations of the parents as is their target location in the offspring sequence. Hence the nucleotides of the parent DNA shuffle in a novel way but retain their locations in the sequence.

The haplotypes reflect the same structure as they can be seen as subsequences obtained from the full DNA by restriction to the markers. So, if haplotype R is a recombination of haplotypes G and H, then all three can be written for some $c \geq 0$ as

$$\begin{aligned} R &= G_1 H_1 G_2 H_2 \cdots G_c H_c \\ G &= G_1 G'_1 G_2 G'_2 \cdots G_c G'_c \\ H &= H'_1 H_1 H'_2 H_2 \cdots H'_c H_c \end{aligned}$$

where $|G_i| = |H'_i| > 0$ for $1 \leq i \leq c$, and $|G'_i| = |H_i| > 0$ for $1 \leq i < c$ and $|G'_c| = |H_c| \geq 0$. Haplotype R has a cross-over between markers i and $i+1$ if the markers do not belong to the same fragment G_j or H_j for some j.

Assume that such recombination steps are applied repeatedly on an evolving set of sequences, starting from an initial set of founder haplotypes. Then a haplotype of a current individual is built from conserved sequence fragments that are taken from the haplotypes of the founders. In other words, each haplotype has a *parse*

$$f_1 f_2 \cdots f_h$$

where each f_i is a contiguous fragment taken from the same location of some founder haplotype, possibly with some rare changes due to point mutations. This parse is unknown. Our goal is to develop HMM modelling techniques that could help uncovering such parses as well as conserved haplotype segments.

To that end, we will introduce a family of HMMs whose states model the conserved sequence fragments and the transitions between the states model the cross-overs. The conserved fragments in the models will be taken from the sequences in $A_1 \times \cdots \times A_m$. The parameters of the HMM will be estimated from a training data that consists of some observed haplotypes in the current population. We will apply the minimum description length principle for the model selection to uncover the fragments that can be utilized in parsing several different haplotypes of the training data. Our model does not make any prior assumption on the distribution of the cross-over points that would prefer, say, a division of the haplotypes into global blocks between recombination 'hot spots'.

2.2 Model Architecture

A hidden Markov model $M = (F, \epsilon, W)$ for modeling haplotype fragmentation consists of a set F of the *states* of M, the *error parameter* $\epsilon \geq 0$ of M, and the *transition probabilities* W between the states of M.

Each state $f \in F$ is actually a haplotype fragment by which we mean contiguous segment of a possible haplotype between some *start marker* and some *end marker*, that is, f is an element of $A_s \times \cdots \times A_e$ where $1 \leq s \leq e \leq m$. We often call the states f the *fragments* of M. The start and end markers of f are denoted by $s(f)$ and $e(f)$, respectively.

The *emission probabilities* $P(d|f)$ of state (fragment) f give a probability distribution for fragments $d \in A_{s(f)} \times \cdots \times A_{e(f)}$ when the underlying founder fragment is f. This distribution will include a model for mutation and noise rates, specified by the noise parameter ϵ. It is also possible to incorporate a model for missing data which is useful if the model training data is incomplete. We adopt perhaps the simplest and most practical alternative, the missing–at–random model. Let us write $f = f_s \cdots f_e$ and $d = d_s \cdots d_e$. Assuming that the markers are independent of each other we have

$$P(d|f) = \prod_{i=s}^{e} P(d_i|f_i),$$

where we define

$$P(d_i|f_i) = \begin{cases} 1 & \text{if } d_i \text{ is missing} \\ 1-\epsilon & \text{if } d_i = f_i \\ \epsilon/(|A_i|-1) & \text{otherwise.} \end{cases}$$

We assume that a value for ϵ is a given constant and do not consider here its estimation from the data. Therefore we omit ϵ from the notation and denote a fragmentation model just as $M = (F, W)$.

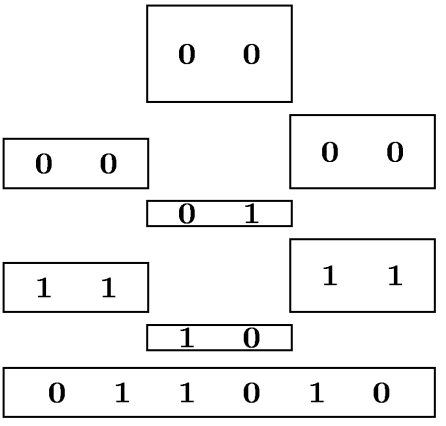

Fig. 1. A simple fragmentation model for haplotypes with 6 binary markers

Model M has a transition from state f to state g whenever f and g are adjacent, that is, if $e(f) + 1 = s(g)$. If this is the case, the transition probablity $W(f, g)$ is defined, otherwise not. The probabilities of the transitions from f must satisfy $\sum_{g \in F} W(f, g) = 1$.

A transition function W defined this way has the drawback of fairly large number of probabilities to estimate. Therefore we consider the following simplified version. A fragmentation model is called *simple* if all probabilities $W(f, g)$ for a fixed g but varying f are equal. Hence g is entered with the same probability, independently of the previous state f. Then we can write $W(f, g) = W(g)$, for short. A simple fragmentation model is specified by giving the fragments F, the transition probability $W(g)$ for each $g \in F$, and the error parameter ϵ. From now on, in the rest of the paper, we only consider the simple fragmentation models.

An example of a simple fragmentation model over 6 (binary) markers is shown in Figure 1. The fragments for the states are shown inside each rectangle, the height of which encodes the corresponding transition probability. For example, each of the three states whose start marker is 1 have transition probability 1/3, and the three states with start marker 3 have transition probabilities 2/3, 1/6, and 1/6. When $\epsilon = 0$ this model generates haplotype 000000 with probability $1/3 \cdot 2/3 \cdot 1/2 = 1/9$ (the only path emitting this sequence with non-zero probability goes through the three states with fragment 00), and haplotype 111111 with probability 0.

2.3 Emission Probability Distribution

Let us recall how a hidden Markov model associates probabilities to the sequences it emits. A *path* through a simple fragmentation model $M = (F, W)$ over m markers is any sequence $(F_1, \ldots, F_h)$ of states of M such that $s(F_1) = 1$, $e(F_i) +$

$1 = s(F_{i+1})$ for $1 \leq i < h$, and $e(F_h) = m$. Let π be the set of all paths through M.

Consider then the probability of emitting a haplotype $H = H_1 \cdots H_m$. Some allele values may be missing in H. The probability that H is emitted from path $(F_1, \ldots, F_h)$ is

$$P(H, (F_1, \ldots, F_h)|M) = \prod_{i=1}^{h} W(F_i) P(H_{s(F_i)} \cdots H_{e(F_i)}|F_i)$$

which simply is the probability that the path is taken and each state along the path emits the corresponding fraction of H, the emission probabilities being as already defined. The probability that M emits H is then

$$P(H|M) = \sum_{(F_1,\ldots,F_h)\in\pi} P(H, (F_1, \ldots, F_h)|M). \tag{1}$$

3 MDL Method for Model Selection

3.1 Description Length

Let D be our training data consisting of n observed haplotypes $D_1, \ldots, D_n$ over m markers. The minimum description length (MDL) principle of Rissanen considers the description of the data D using two components: description of the model M and description of the data D given the model. Hence the total description length for the model and the data is

$$L(M, D) = L(M) + L(D|M)$$

where $L(M)$ is the length of the description of M and $L(D|M)$ is the length of the description of D when D is described using M.

The MDL principle states that the desired descriptions of the data are the ones having the minimum length $L(M, D)$ of the total description. For a survey of the connections between MDL, Bayesian statistics, and machine learning see [9,5].

To apply this principle we have to fix the encoding scheme that will give $L(M)$ and $L(D|M)$ for each particular M and D.

The model $M = (F, W)$ can be described by telling what are the fragments F_i in F and where they start. The transition probabilities $W(F_i)$ should also be given. The error parameter ϵ is assumed constant and hence it needs not to be encoded.

To encode fragment F_i we use

$$L(F_i) = \sum_{j=s(F_i)}^{e(F_i)} \log |A_j| + \log m$$

bits where $\log |A_j|$ bits are used for representing the corresponding allele of F_i, and $\log m$ bits are used for $s(F_i)$. The probabilities $W(F_i)$ are real numbers.

Theoretical arguments [5] indicate that an appropriate coding precision is obtained by choosing to use $\frac{1}{2}\log n$ bits for each independent real parameter of the model; recall that n is the size of training data D. In our case there are $|F| - t$ independent probability values where t denotes the number of different start markers of the fragments in F. One of the probabilities $W(F_i)$ for fragments F_i with the same $s(F_i)$ namely follows from the others as their sum has to equal 1. Thus the total model length becomes

$$\begin{aligned} L(M) &= \sum_{i=1}^{|F|} L(F_i) + \frac{|F| - t}{2} \log n \\ &= \sum_{i=1}^{|F|} (\sum_{j=s(F_i)}^{e(F_i)} \log |A_j| + \log m) + \frac{|F| - t}{2} \log n. \end{aligned} \tag{2}$$

Using the relation of coding lengths and probabilities [5], we use

$$L(D|M) = -\log \prod_{i=1}^{n} P(D_i|M) = -\sum_{i=1}^{n} \log P(D_i|M) \tag{3}$$

bits for describing the data D, given the fragmentation model M. Here probabilities $P(D_i|M)$ can be evaluated using (1).

We are left with designing an algorithm that solves the MDL minimization problem. One has to find an $M = (F, W)$ for a fixed error parameter ϵ that minimizes $L(M) + L(D|M)$ for a given data D.

3.2 Greedy Algorithms for MDL Optimization

As solving the MDL optimization exactly in this case seems difficult we give a greedy approximation algorithm. The algorithm has to solve two tasks that are to some degree independent. First, one has to find a good fragment set F for the model. Note that the model length $L(M)$ depends only on F (and on n). Second, for a fixed F one has to find a good W such that $L(D|M)$ is minimized.

Let us consider the second task first. Given F and D, we will use the well-known expectation maximization (EM) algorithm for finding W that gives (locally) maximum $P(D|M)$ from which we get minimum $L(D|M)$; see e.g. [2, pp 63–64]. The EM algorithm, delinated as Algorithm 1 below, starts with some initial W and then computes the expected number of times each state of the model with the current W is used when emitting D. These values are normalized to give updated W. The process is repeated with the new W until convergence (or a given number K of times). When converged, local maximum likelihood estimates for W have been obtained.

Algorithm 1. EM–algorithm for finding W for a model $M = (F, W)$.

1. Initialize W somehow.

2. (E–step) For each training data sequence $D_i \in D$ and fragment $f \in F$, compute $q_i(f)$ as the probability that M emits $D_{i,s(f)} \cdots D_{i,e(f)}$ from f. This can be done using the standard Forward and Backward algorithms for HMMs.
3. (M–step) For each fragment f, let $q(f) \leftarrow \sum_{i=1}^{n} q_i(f)$. Finally set

$$W(f) \leftarrow \frac{q(f)}{\sum_{f' \in F(s(f))} q(f')}$$

   where $F(s(f))$ denotes the subset of F that consists of all fragments having the same start marker as f.
4. Repeat steps 2 and 3 until convergence (or until a given number of iterations have taken).

We will denote by $EM(F)$ the W obtained by Algorithm 1 for a set F of fragments.

To analyze the running time of Algorithm 1, assume that we have precomputed the probabilities $P(D_{i,s(f)} \cdots D_{i,e(f)}|f)$ for each $f \in F$ and $D_i \in D$. Then the Forward and Backward algorithms in step 2 just spend a constant time per each fragment when they scan F from left to right and from right to left, respectively. This happens for each D_i. Hence step 2 needs time $O(n|F|)$. This obviously dominates the time requirement of step 3, too. So we obtain the following remark.

Proposition 1. *Algorithm 1 (EM algorithm) for $M = (F, W)$ takes time $O(Kn|F|)$ where K is the number of iterations taken.*

Let us then return to the first task, selecting fragment set F. According to our definition of fragmentation models, set F should be selected from the set of all possible fragments of the sequences in $A_1 \times \cdots \times A_m$. As this search space is of exponential sixe in m, we restrict the search in practice to the fragments that are present in D. Let us denote as $\Phi(D) = \{D_{i,j} \cdots D_{i,k} \mid D_i \in D, 1 \leq j \leq k \leq m\}$ the initial set of frgments obtained in this way. Then $|\Phi(D)| = O(nm^2)$.

We select F from $\Phi(D)$ using a simple greedy strategy: delete from $\Phi(D)$ the fragment whose elimination maximally improves the score $L(M) + L(D|M)$. If a fragment is deleted, then one has to remove also all other fragments that the deletion makes isolated. A fragment is isolated if no path through the model can contain it, which is easy to test. Repeat deletions until the score does not improve. The remaining set of fragments is F. This method is given in more detail as Algorithm 2.

Algorithm 2. Basic greedy MDL learning

Notation: $M(-f)$ = the model that is obtained from the current model $M = (F, W)$ by deleting fragment f and all fragments that become isolated as a side–effect of the removal of f from F and by updating W by Algorithm 1 such that $L(D|M(-F))$ is minimal.

1 Initialize $M = (F, W)$ as $F \leftarrow \Phi(D)$; $W \leftarrow EM(\Phi(D))$
2 Return Greedy(M) where procedure Greedy is as follows.

procedure Greedy(M), where $M = (F, W)$
  do
    $L \leftarrow L(M) + L(D|M)$
    $\Delta \leftarrow \max_{f \in F}(L - [L(M(-f)) + L(D|M(-f))])$
    $f_\Delta \leftarrow \arg\max_{f \in F}(L - [L(M(-f)) + L(D|M(-f))])$
    if $\Delta > 0$ **then** $M \leftarrow M(-f_\Delta)$
  until $\Delta \leq 0$
  return M
end Greedy

To implement Algorithm 2 efficiently one has to precompute the coding lengths of all the fragments in $\Phi(D)$. This can be done in time $O(|\Phi(D)|)$ as the length only depends on the location of the fragment but not on the content. Then $L(M)$ can be evaluated for any $M = (F, W)$ in time $O(|F|) = O(|\Phi(D)|)$. Training a new $M(-f)$ by Algorithm 1 can be done in time $O(Kn|F|) = O(Kn|\Phi(D)|)$ where K is the parameter limiting the number of iterations. To find Δ and f_Δ in procedure Greedy, a straightforward implementation just tries each $f \in F$, taking time $O(Kn|\Phi(D)|^2)$. This will be repeated until nothing can be deleted from the set of fragments, i.e., $O(|\Phi(D)|)$ times. The total time of Algorithm 2 hence becomes $O(Kn|\Phi(D)|^3) = O(Kn^4m^6)$.

As Algorithm 2 can be practical only for very small m and n, we next develop a faster incremental version of the greedy MDL training. This algorithm contructs intermediate models using the initial segments of the sequences in D, in increasing order of the length. A model, denoted M_{j+1} for the initial segments of length $j + 1$ will be constructed by expanding and retraining the model M_j obtained in the previous phase for the initial segments of length j.

Let $\Phi_j(D) = \{D_{i,k} \cdots D_{i,j} \mid D_i \in D, 1 \leq k \leq j\}$ be the set of fragments of D that end at marker j. To get M_{j+1}, fragments $\Phi_{j+1}(D)$ are added to M_j and then the useless fragments are eliminated as in Algorithm 2 by procedure Greedy to get M_{j+1}. Adding the fragments in such smaller portions to the optimization leads to a faster algorithm.

A detail needs additional care. Adding $\Phi_{j+1}(D)$ alone to the set of fragments may introduce isolated fragments that have no possibility to survive in the MDL optimization because they are never reached although they could be useful in encoding the data. To keep the new set of fragments connected we therefore also add fragments that bridge the gaps between the old fragments inherited from M_j and the new fragments in $\Phi_{j+1}(D)$. We use the following bridging strategy that by adding only the shortest bridges keeps the number of bridging fragments relatively small. We say that the set $\gamma(F)$ of the *gaps* of a fragment set F consists of all pairs (k, h) of integers such that $k - 1 = e(f)$ for some $f \in F$ but there is no $f \in F$ such that $k \leq e(f) \leq h$. Then define $\Phi_j(D, F) = \Phi_j(D) \cup \{D_{i,k} \cdots D_{i,h} \mid D_i \in D, (k, h) \in \gamma(F), h < j\}$. Now, we add for the new round the set $\Phi_{j+1}(D, F_j)$ where F_j is the fragment set of M_j.

Algorithm 3. Incremental greedy MDL learning

Notation: Procedure Greedy as in Algorithm 2.

1. Initialize $M = (F, W)$ as $F \leftarrow \emptyset$; $W \leftarrow \emptyset$
2. **for** $j \leftarrow 1, \ldots, m$ **do**

$$(F, W) \leftarrow \text{Greedy}(F \cup \Phi_j(D, F), EM(F \cup \Phi_j(D, F))$$

3. **return** $M = (F, W)$.

For a running time analysis of Algorithm 3 assume that the greedy MDL optimization is able to reduce the the size of the fragment set in each phase to $O(mn)$. This is a plausible assumption as the size of D is mn. Then the size of each $F \cup \Phi_j(D, E)$ given to procedure Greedy in step 2 stays $O(mn)$. Hence Greedy generates $O(m^2n^2)$ calls of Algorithm 1, each taking time $O(Kmn^2)$. This is repeated m times, giving altogether $O(m^3n^2)$ calls of Algorithm 1 and total time $O(Km^4n^4)$ for Algorithm 3.

Adding new fragments into the optimization in still smaller portions can in some cases give better running times (but possibly at the expense of weaker optimization results). One can for example divide $\Phi_j(D)$ into fractions $\Phi_{k,j}(D)$ consisting of equally long fragments which start at marker k and end at j. The greedy MDL optimization is performed after adding the next $\Phi_{k,j}(D)$ and the necessary bridging fragments.

4 Using the Fragmentation Models

Once we have trained a model M, the numerous possibilities of applying such an HMM become available. We delineate here some of them.

1. Parsing haplotypes. Given some haplotype H, the path through M that has the highest emission probability of H is called the *Viterbi path* of H. Such a path can be efficiently found by standard dynamic programming, e.g. [2]. The fragments on this path give a natural parsing of H in terms of the fragments of M. The parse is the most probable decomposition of H into conserved pieces as proposed by M. Such a parse can be visualized by associating a unique color with each fragment of M, and then showing the parse of H with the colors of the corresponding fragments. Using the same color for conserved pieces seems a natural idea, independently proposed at least in [15,17].

2. Cross–over and fragment usage probabilities. The probability that M assigns a cross–over between the markers i and $i + 1$ of haplotype H can be computed simply as the fraction between the probability of emitting H along paths having a cross–over in that point and the total probability $P(H|M)$ of emitting H along any path. Similarly, the probability that a certain fragment of M is used for emitting H is the fraction between the emission probability of H along paths containing this fragment and $P(H|M)$.

3. Comparisons between populations and case/control studies. Assume that we have available training data sets D^i from different populations. Then the corresponding trained models M^i can be used for example for classifying new haplotypes H on the basis of emission probabilities $P(H|M^i)$. The structure of the models M^i may also uncover interesting differences between the populations. For example, some strong fragments may be characteristic of a certain population but missing elsewhere. Also, the average length of the fragments should be related to the age of the population. An older population has experienced more recombinations, hence its fragments should be shorter on average as those of a younger population.

To demonstrate our methods we conclude by analyzing two real datasets related to the lactase nonpersistence (lactose intolerance) of humans [16]. Lactase nonpersistence limits the use of fresh milk among adults. The digestion of milk is catalyzed by an enzyme lactase (also called lactase–phlorizin hydrolase or LHP). Lactase activity is high during infancy but in most mammals declines after the weaning phase. In some healthy humans, however, lactase activity persists at high level throughout adult life, and this is known as lactase persistence. People with lactase nonpersistence have a much lower lactose digestion capacity than those with lactase persistence. A recent study [3] of a DNA region containing LCT, the gene encoding LHP, revealed that in certain Finnish populations a DNA variant (that is, an SNP), $\mathrm{C/T}_{-13910}$, almost 14 kb upstream from the LCT locus, completely associates with verified lactase nonpersistence.

The datasets we analyse consist of haplotypes over 23 SNP markers in the vicinity of this particular SNP. The first dataset D_+, the persistent haplotypes, consists of 38 haplotypes of lactase persistent individuals, and the second dataset D_-, the nonpersistent haplotypes, consists of 21 haplotypes of lactase nonpersistent individuals. We trained models for datasets D_-, D_+, and $D_+ \cup D_-$ using Algorithm 3 with $\epsilon = 0.001$ and $K = 3$ iterations of the EM algorithm. Figures 2, 3, and 4 show the result. The fragments are shown as colored rectangles. Their height encodes the transition probability somewhat differently from the encoding used in Fig. 1. Here the height of fragment f gives the value $\tilde{W}(f) = q(f)/n$ where $q(f)$ is as in Algorithm 1 after the last iteration taken, and n is the number of haplotypes in the training data. Hence the height represents the avarage usage of f when the model emits the training data. The measure $\tilde{W}(f)$ describes the importance of each fragment for the training data better than the plain transition probability $W(f)$.

We observe that the model for persistent haplotypes has on average longer fragments than the model for nonpersistent haplotypes. This suggests, consistently with common belief, that lactase persistence is the younger of the two traits. When analyzing the Viterbi paths in the model for $D_+ \cup D_-$, we observed that all such paths for persistent haplotypes go through the large fragment in the middle of the model (number 4 from the top), while none of the Viterbi paths for the nonpersistent haplotypes includes it.

We also sampled two thirds of D_+ and D_- and trained models for the two samples. Let M_+ and M_- be the two models obtained. We then com-

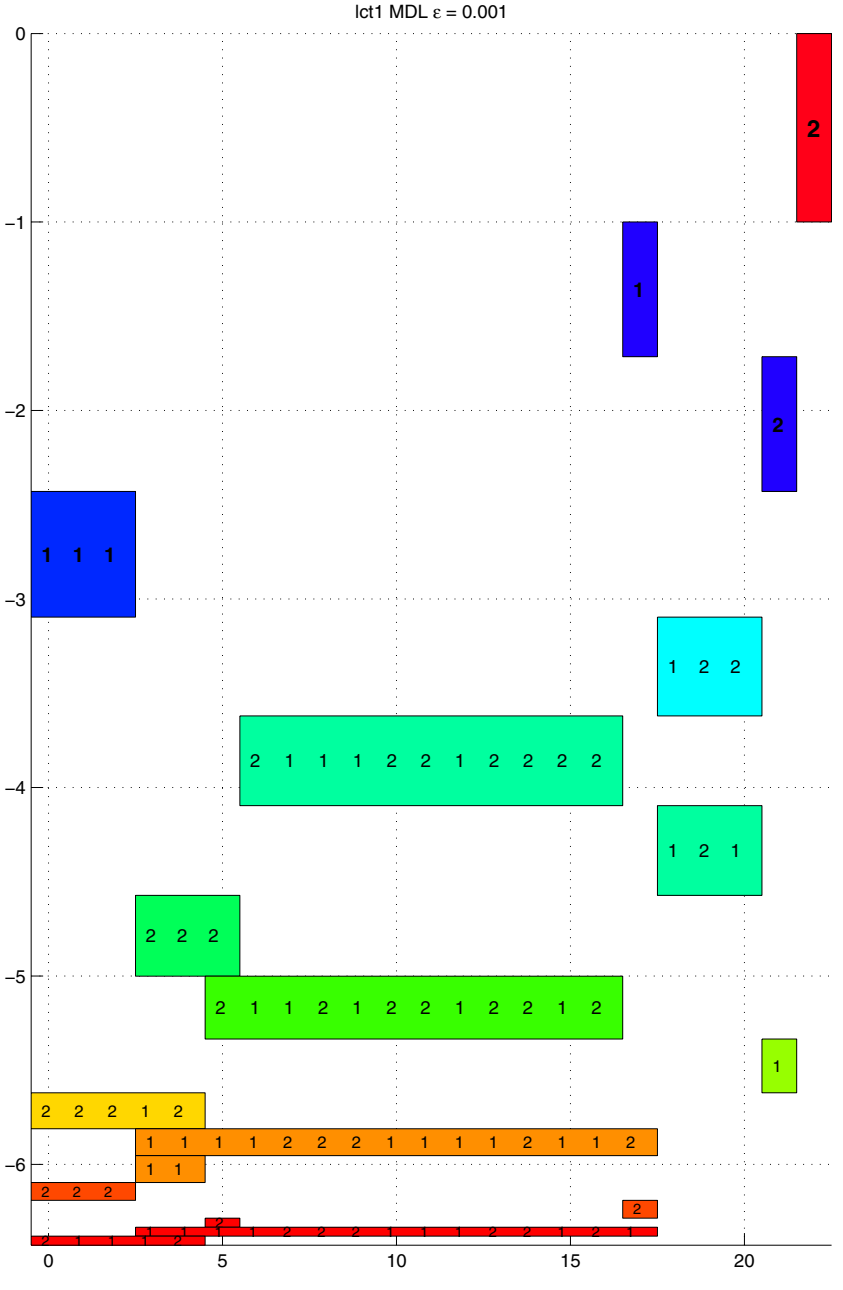

Fig. 2. A fragmentation model for nonpersistent haplotypes.

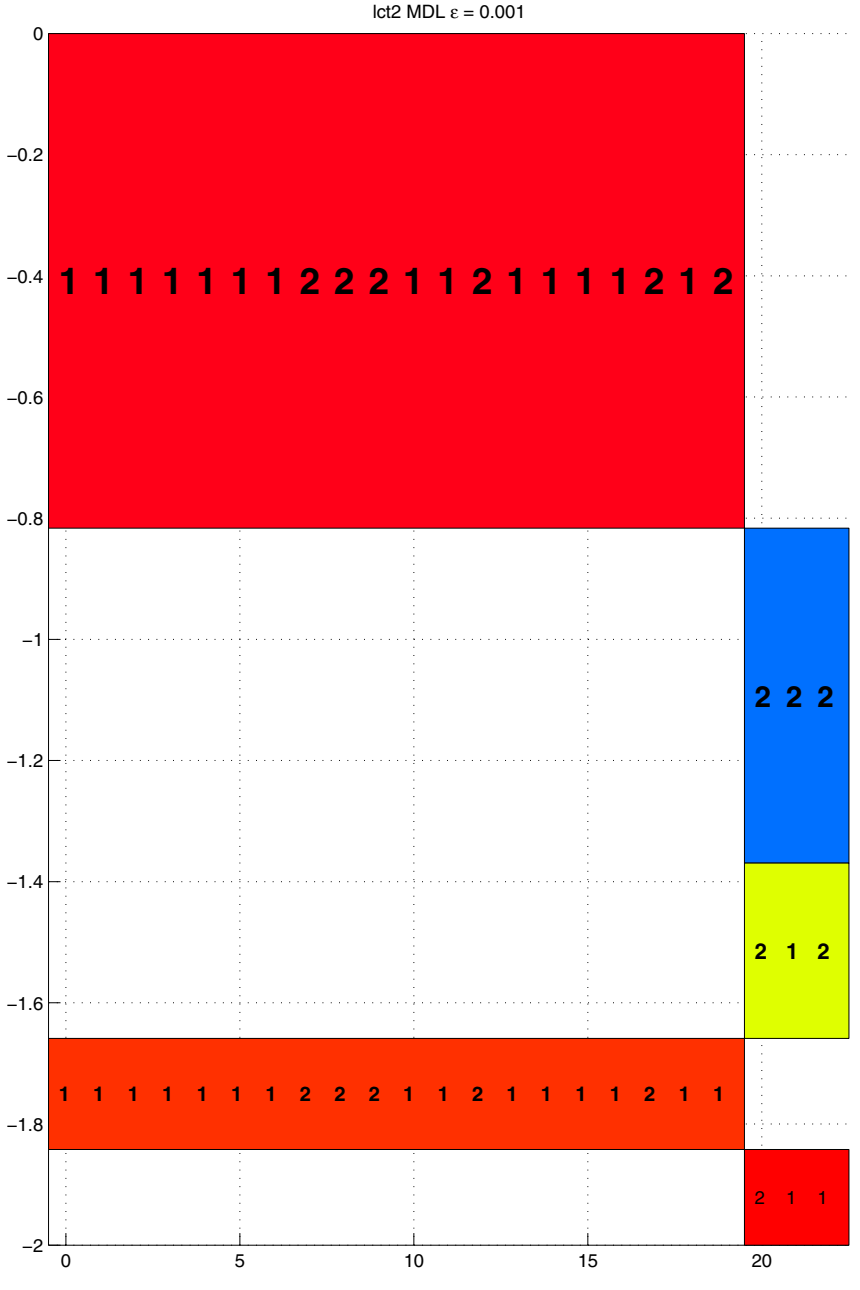

Fig. 3. A fragmentation model for persistent haplotypes.

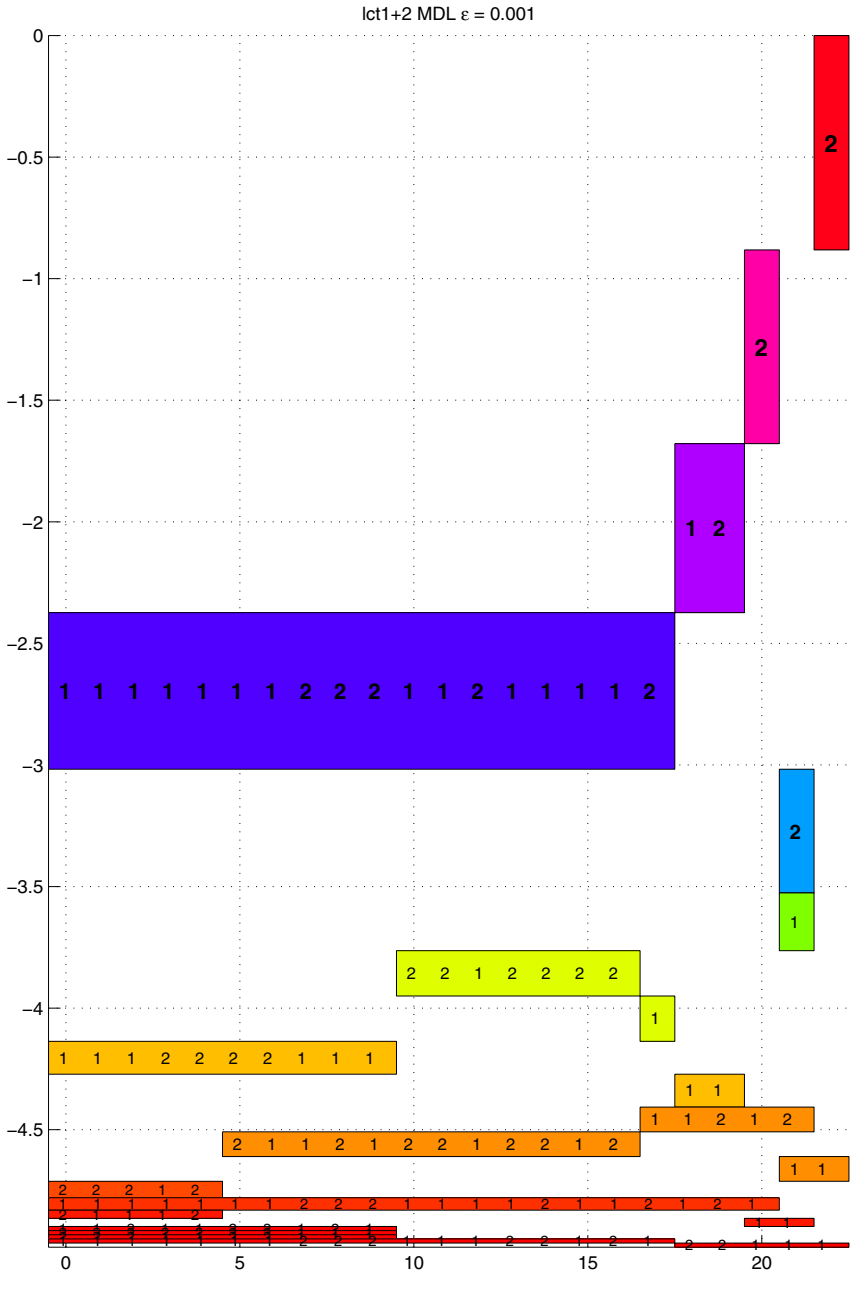

Fig. 4. A fragmentation model for the union of persistent and nonpersistent haplotypes

puted for all test haplotypes H outside the training samples the quantity $Q(H) = \log_{10} P(H|M_+)/P(H|M_-)$. One obviously expects that $Q(H) > 0$ when H is a persistent haplotype and $Q(H) < 0$ otherwise. Satisfyingly we found out in this experiment, that $Q(H)$ varied from 6.0 to 11.8 for persistent test haplotypes and from -2.6 to -44.7 for nonpersistent test haplotypes.

5 Conclusion

While our experimental evaluation of the methods suggests that useful results can be obtained in this way, many aspects still need further work. More experimental evaluation on generated and real data is necessary. We only used the simple models but also the general case should be considered. The approximation algorithm for the MDL learning seems to work quite robustly but theoretical analysis of its performance is missing. Also faster variants of the learning algorithm may be necessary for larger data. It may also be useful to relax the model such that the fragments of the model are not required to cover the haplotypes entirely but small gaps between them are allowed.

References

[1] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander. High–resolution haplotype structure in the human genome. *Nature Genetics* 29: 229–232, 2001.

[2] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press 1998.

[3] N. S. Enattah, T. Sahi, E. Savilahti, J. D. Terwilliger, L. Peltonen, and I. Järvelä. Identification of a variant associated with adult–type hypolactasia. *Nature Genetics* 30: 233-237, 2002.

[4] S. B. Gabriel, S. F. Schaffner, H. Ngyen, J. M. Moore *et al.* The structure of haplotype blocks in the human genome. *Science* 296: 2225–2229, 2002.

[5] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association* 96: 746–774, 2001.

[6] 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: *Proc. Pacific Symposium on Biocomputing*, 502–513, World Scientific 2003.

[7] M. Koivisto, P. Rastas, and E. Ukkonen. Recombination systems. In: *Theory is Forever – Essays dedicated to Arto Salomaa*, *LNCS* 3113: 159–169, 2004.

[8] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.* 235: 1501-1531, 1994.

[9] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer–Verlag 1997.

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

[11] M. S. Phillips, R. Lawrence, R. Sachidanandam, A. P. Morris, D. J. Balding *et al.* Chromosome–wide distribution of haplotype blocks and the role of recombination hot spots. *Nature Genetics* 33: 382–387, 2003.

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

[13] J. Rissanen. Stochastic complexity. *J. Royal Statistical Society* B 49: 223–239, 1987.

[14] R. Schwartz. Haplotype motifs: An algorithmic approach to locating evolutionarily conserved patterns in haploid sequences. *Proc. Computational Systems Bioinformatics (CSB'03)*, 306–314, IEEE Computer Society 2003.

[15] R. Schwartz, A. G. Clark, and S. Istrail. Methods for inferring block–wise ancestral history from haploid sequences: The haplotype coloring problem. *WABI 2002*, *LNCS* 2452, 44–59.

[16] D. M. Swallow. Genetics of lactase persistence and lactose intolerance. *Annu. Rev. Genet.* 37: 197–219, 2003.

[17] E. Ukkonen. Finding founder sequences from a set of recombinants. *WABI 2002*, *LNCS* 2452, 277–286.

[18] K. Zhang, M. Deng, T. Chen, M. S. Waterman and F. Sun. A dynamic programming algorithm for haplotype block partition. *PNAS* 99: 7335–7339, 2002.