Artificial Intelligence in Medicine (2009) **45**, 135—150

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Fully non-homogeneous hidden Markov model double net: A generative model for haplotype reconstruction and block discovery

Alessandro Perina [a,*], Marco Cristani [a], Luciano Xumerle [b], Vittorio Murino [a], Pier Franco Pignatti [b], Giovanni Malerba [b]

[a] *Department of Computer Science, University of Verona, Strada le Grazie 15, 37134 Verona, Italy*
[b] *Department of Mother and Child, Biology and Genetics, Section Biology and Genetics, University of Verona, Strada le Grazie 8, 37134 Verona, Italy*

Received 31 October 2007; received in revised form 21 August 2008; accepted 22 August 2008

KEYWORDS
Haplotype reconstruction;
Bayesian network;
Variational learning;
Block structure

Summary

Objective: In the last decade, haplotype reconstruction in unrelated individuals and haplotype block discovery have riveted the attention of computer scientists due to the involved strong computational aspects. Such tasks are usually addressed separately, but recently, statistical techniques have permitted them to be solved jointly. Following this trend we propose a generative model that permits researchers to solve the two problems jointly.
Method: The model inference is based on variational learning, which permits one to estimate quickly the model parameters while remaining robust even to local minima. The model parameters are then used to segment genotypes into blocks by thresholding a quantitative measure of boundary presence.
Results: Experiments on real data are presented, and state-of-the-art systems for haplotype reconstruction and strategies for block estimation are considered as comparison.
Conclusions: The proposed method can be used for a fast and reliable estimation of haplotype frequencies and the relative block structure. Moreover, the method can be easily used as part of a more complex system. The threshold used for block discovery can be related to the quality-of-fit reached in the model learning, resulting in an unsupervised strategy for block estimation.

* Corresponding author. Tel.: +39 045 8027803; fax: +39 045 8027068.
E-mail addresses: alessandro.perina@univr.it (A. Perina), marco.cristani@univr.it (M. Cristani), luciano.xumerle@medgen.univr.it (L. Xumerle), vittorio.murino@univr.it (V. Murino), pignatti@medgen.univr.it (P. Pignatti), giovanni.malerba@medgen.univr.it (G. Malerba).

0933-3657/$ — see front matter
doi:10.1016/j.artmed.2008.08.015

1. Introduction

Estimating haplotype[1] frequencies has become increasingly important in the mapping of complex disease genes, since large numbers of closely linked single nucleotide polymorphisms (SNPs) can be genotyped. SNPs are single base pair differences among individuals in a population. Association studies work on the premise that some SNP genotypes are correlated with a disease phenotype. Numerous studies have shown that the human genome contains regions of high *linkage disequilibrium* (LD) with low haplotype diversity [1]: these regions are called *haplotype blocks* or *LD blocks*, where LD is a non-random association of alleles between adjacent SNPs. It is worth noting that SNPs or haplotypes within LD blocks may serve as proxies for causative and still unknown alleles; therefore, an accurate study on the blocks diversity results in a key factor in genome-wide association studies [2] to identify LD blocks containing a susceptible genetic factor that was not yet genotyped. Unfortunately, the allele phase of multilocus genotypes in unrelated individuals is unknown and haplotypes need to be reconstructed [3] before the discovery of LD blocks [4].

Statistical strategies for haplotype reconstruction have been recently introduced [3,5—10]. However, all these strategies, except [6], either do not perform block discovery or do it after the haplotype reconstruction, so that block discovery might be affected by potential reconstruction errors.

In this paper, we propose a statistical generative model for haplotype reconstruction and block estimation, called *fully non-homogenous hidden Markov model double net* (FNH-HMM double net). The idea of a generative model is to describe the process that generated the observations, employing random variables connected by a conditional probability distribution. Dynamic Bayesian networks (DBNs), and in particular, hidden Markov models (HMMs) [11], are the most known examples of generative models employed for haplotype reconstruction. In [5—7], the idea is to estimate relevant hidden "ancestral" patterns from genotype data, i.e. biologically meaningful allele patterns that represent high frequency haplotype fragments, mimicking biological theories [3]. HMMs are employed to model the fact that alleles at nearby markers are likely to arise from the same ancestral pattern, thus resulting in a block-like structure, where each block begins and terminates in correspondence with recombination hot spots. Our approach closely mirrors the process of genotype generation, considering that different portions of the genotype have different probabilities of recombination and also that the LD is higher in some regions than in others (as in [5,6]). In our model, the phase information that determines haplotype reconstruction is explicitly modeled, which differs from all the other HMM-based approaches [5—7]. To this end, we employ binary variables with their own distribution estimated by the learning strategy. In this way, the haplotype reconstruction can be performed very easily once the model has completed the learning phase. The uncertainty at each site is therefore evaluated during the phasing process and cannot be computed in the other approaches where the haplotype inference is based on sampling strategies [7], or on a set of maximization procedures [5,6]. In our case, the explicit managements of the genotype generation process is done in terms of a complex (i.e. with several variables) model that has a modular structure; each module is a simple generative model (a FNH-HMM, here formally introduced). This, together with a novel inference strategy for learning based on *variational learning* [12], permits one to learn the model with a time complexity less than in [6] and comparable to that in [5,7]. More importantly, our strategy is not dependent on the parameter initialization, and less prone to local minima solutions.

The variational learning technique was introduced in this field by [13]. Here the authors introduced a similar generative model based on hybrid HMM for classification purposes: their goal was to distinguish genotypes that belong to patients affected by Crohn's disease from those belonging to healthy patients, starting with the exact knowledge of the block boundaries.

Together with the learning strategy, we equipped our model with an inference procedure strategy that permits it to estimate blocks. In practice, using the FNH-HMM double net, each reconstructed haplotype can be considered as the most probable path among estimated ancestral patterns (due to the recombination). Our inference identifies frequent splits and joins among the paths, which can be considered as block boundaries. Segmentation of block boundaries are determined by thresholding an econometric-based statistical measure, the *Gini index* [14], which represents the strength of a block. The threshold is easy to find, but, more importantly, it can be related to the degree of fit with which the learning step described the data (i.e. the data log-likelihood w.r.t. the model parameters). This means that (1) the uncertainty collected during the learning can explicitly flow down in the block estimation step and (2) the block estimation becomes an unsupervised operation.

[1] *Haplotypes* are combinations of DNA marker alleles in a single chromosome.

Notions of block boundary strength have appeared earlier in this field [15,16]. For example in [16] the authors described a dynamic programming algorithm for finding the optimal segmentation with respect to the minimum description length (MDL) principle [17]. However, the strength measure proposed in [16] is very different from the one proposed here, the time complexity is much higher, and the blocks are calculated using haplotypes.

The rest of the paper is organized as follows: Section 2 gives some background mathematical notions, fix the notation and introduces the FNH-HMM; Sections 3—6 explain our framework and Section 7 shows comparative experimental results on haplotype reconstruction and block estimation. Finally, Section 8 draws some conclusions and envisages future perspectives.

2. Preliminaries

2.1. Generative models and Bayesian networks

The goal of the generative modeling is to develop statistical models that can explain the input data (samples or visible variables, v), as tangible *effects* being generated by a combination of hidden variables (h), representing the *causes*, eventually interconnected with conditional interdependencies. A generative model jointly models the input and the causes via a joint probability distribution $P(v,h)$ or $P(\mathit{effects},\mathit{causes})$.

Graphical models are well-known instruments that represent effectively generative models; they use graphs to represent and manipulate joint probability distributions. The states of the graphs represent random (visible or hidden) variables and the edges codify conditional dependence relations among them. Several types of graphical models are present in the literature [18], and, among these, the most used is the Bayesian network.

A Bayesian network for random variables (RVs) $x_1,\ldots,x_R$ is a directed acyclic graph (see an example in Fig. 1 a).

The nodes of the graph represent the variables, while the directed arcs represent probabilistic dependencies among them. There are two kinds of nodes, the hidden nodes h modeling the hidden variables, and the observable nodes v representing the visible variables. In a Bayesian network a conditional probability function is specified for each RV given its parents, $P(x_i|x_{A_i})$, where A_i is the set of indices of x_i's parents and x_i represents either a hidden or visible variable.

Usually, each $P(x_i|x_{A_i})$ is governed by a set of (hidden) parameters θ_i specifying the particular (parametric) form assumed by the conditional probability function (e.g. Gaussian). The parameters $\theta = \bigcup_{i=1}^{R}\theta_i$ are treated like hidden variables and are thus included in h.

The joint distribution $P(x)$, $x=\{v,h\}$, is given by the product of all the conditional probability functions:

$$P(x)=\prod_{i=1}^{R}P(x_i|x_{A_i}) \tag{1}$$

For example, in the Bayesian network of Fig. 1 a we have:

$$P(a,b,c,y)=P(b)\cdot P(c)\cdot P(a|c)\cdot P(y|a,b,c) \tag{2}$$

After having fixed the topology of a Bayesian network (i.e. the nodes and their interdependencies, their conditional parametric functional form), it is necessary to learn the model. Learning consists in inferring the hidden quantities (hidden variables and parameters) using the observations, i.e. choosing a possible instance of values for h maximizing the *a posteriori* distribution $P(h|v)$ (*Maximum A Posteriori* learning), or alternatively, the likelihood $P(v|h)$ (*Maximum Likelihood* learning) [19].

In both cases, this choice cannot often be performed in closed form, and often is not even possible by exploring the space of the possible values assumed by h, since such space is exponential in the number of variables. Therefore, instead of considering the exact posterior distribution $P(h|v)$, it becomes advantageous to operate on approximations of $P(h|v)$, simpler than $P(h|v)$. Variational approximate learning [12] consists in inferring the

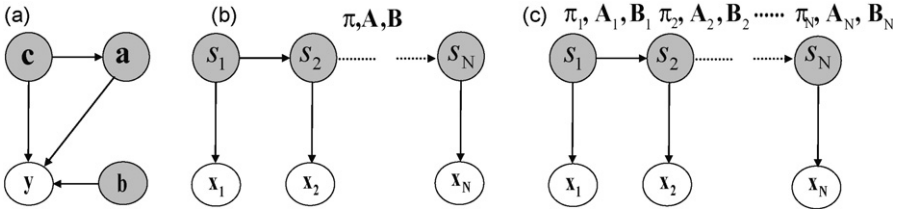

Figure 1 (a) An example of Bayesian network. a,b and c are the hidden variables, while y is the only observed variable; (b) HMM with the respective parameters; (c) FNH-HMM; note that transition and emission matrices are now time-dependent.

hidden quantities of a distribution $\mathcal{Q}(h)$ related to $P(h|v)$ while minimizing a quantity called *free energy*, which is defined as follows:

$$\mathcal{F}(P,\mathcal{Q}) = \int_h \mathcal{Q}(h)\log\mathcal{Q}(h) - \int_h \mathcal{Q}(h)\log P(h,v). \quad (3)$$

The free energy is a measure of the approximation accuracy of $P(h|v)$ by $\mathcal{Q}(h)$, since

$$\mathcal{F}(P,\mathcal{Q}) = \mathbb{KL}(P,\mathcal{Q}) - \log P(v) \quad (4)$$

where $\mathbb{KL}(P,\mathcal{Q})$ is the Kullback–Leibler divergence [14] between P and $\mathcal{Q}$.

2.2. Graphical models for sequential data: hidden Markov models

A special type of a Bayesian network is the dynamic Bayesian network (DBN) [20], aimed at modeling sequential data, which in turn are intended as realizations of a stochastic process. Roughly speaking, a DBN is a Bayesian network whose structure is replicated N times (for N *slices*), where N is the length of the sequence. Each slice can be connected with the other ones via additional conditional dependencies.

The best-known DBN is the discrete-time hidden Markov model λ, which can be viewed as a Markov model whose states are not directly observable (Fig. 1b); instead, each state is characterized by a probability distribution function, modeling the observations corresponding to that state. More formally, a HMM is defined by the following entities [11]:

(1) Q, $|Q| = L$, the finite set of (hidden) states;
(2) a transition matrix $\mathbf{A} = \{a^{mn}\}$, $1 \le m,n \le L$ representing the probability of moving from state m to state n:
$$a^{mn} = P(S_{k+1} = n | S_k = m), \quad 1 \le n,m \le L, \quad (5)$$
with $a^{mn} \ge 0$, $\sum_{n=1}^{L} a^{mn} = 1$, and where S_k denotes the state occupied by the model at index k. Depending on the context, the index k indicates a *time* index if the considered sequence is thought of as generated by a temporal stochastic process, or it is considered a *site* index if the sequence is atemporal, with its spatial structure regulated by a Markovian process;
(3) an emission matrix $\mathbf{B} = \{b^m(v)\}$, indicating the probability of emission of symbol $v \in V$ when the system state is m. In this problem context, $V = \{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{T}\}$ indicating Adenine (A), Cytosine (C), Guanine (G) and Thymine (T);
(4) the initial state probability distribution $\boldsymbol{\pi} = \{\pi^m\}$
$$\pi^m = P(S_1 = m), \quad 1 \le m \le L \quad (6)$$
with $\pi^m \ge 0$ and $\sum_{m=1}^{L} \pi^m = 1$.

For convenience, we represent an HMM by a triplet $\lambda = (\mathbf{A},\mathbf{B},\pi)$.

2.3. Fully non-homogeneous hidden Markov model

Suppose we have a set of J one-dimensional observation sequences each of length N, formed by symbols from the set V, i.e. $O^{(j)}$ with $j = 1 \ldots J$.

A fully non-homogeneous hidden Markov model (FNH-HMM) (shown in Fig. 1c) is a set $\lambda_{\mathrm{FNH}} = \{\mathbf{A}_k, \mathbf{B}_k, \pi\}_{k=1}^{N}$ composed by the following parameters:

(1) A *site-dependent* transition matrix $\mathbf{A}_k = \{a_k^{mn}\}$ where
$$a_k^{mn} = P(S_{k+1} = n | S_k = m), \quad 1 \le m,n \le L \quad \text{and} \quad k = 1,\ldots,N; \quad (7)$$
(2) A *site-dependent* emission matrix $\mathbf{B}_k = \{b_k^m(v)\}$ where
$$b_k^m(v) = P(v|S_k = m), v \in V, \quad 1 \le m \le L \quad \text{and} \quad k = 1,\ldots,N; \quad (8)$$
(3) An initial state distribution $\pi = \{\pi^m\}$, $1 \le m \le L$.

The learning of a FNH-HMM is devised as a modified version of the Baum–Welch (BW) algorithm.

In this phase, the E-step consists in first calculating the standard forward and backward variables α and β, paying attention that all the transition and emission probabilities involved are site dependent (i.e. dependent on k). From these variables, key quantities can be obtained, such as the conditional probability of two consecutive hidden states in an observation sequence at site k, i.e. $P(S_k = m, S_{k+1} = n | O^{(j)}) = \xi_k^{(j)}(m,n)$ and the conditional $P(S_k = m|O^{(j)}) = \sum_{n=1}^{L} \xi_k^{(j)}(m,n) = \gamma_k^{(j)}(m)$, where ξ is defined as

$$\xi_k^{(j)}(m,n) = \frac{\alpha_k^{(j)}(m)\, a_k^{mn}\, b_{k+1}^n(O_{k+1}^{(j)})\, \beta_{k+1}^{(j)}(n)}{P(O^{(j)}|\lambda)} \quad (9)$$

In the M-step the parameters are updated using these quantities. The transition $\mathbf{A}_k$ and the emission $\mathbf{B}_k$ matrices are updated as follows:

$$a_k^{mn} = \frac{\sum_{j=1}^{J} \xi_k^{(j)}(m,n)}{\sum_{j=1}^{J}\sum_{n=1}^{L} \xi_k^{(j)}(m,n)}$$
$$b_k^m(v) = \frac{\sum_{j=1}^{J} \gamma_k^{(j)}(m) \quad \text{s.t.} O_k = v}{\sum_{j=1}^{J}\sum_{n=1}^{L} \xi_k^{(j)}(m,n)}. \quad (10)$$

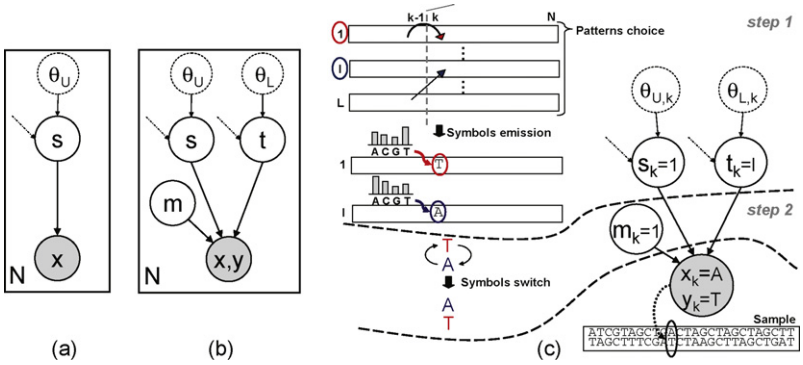

Figure 2 (a) FNH-HMM; (b) FNH-HMM *double net*: nodes in a solid box indicate that they are replicated the number of times indicated in the bottom left corner; point-dashed arrows mean 1st order Markov dependency. Filled (unfilled) circles mean observed (unobserved) random variables; dotted circles indicate the parameter set of the variables linked by the arrow; (c) SNPs' generative process: the picture is divided in two steps. Each step shows a portion of the process, drawn in an intuitive fashion (left) and in a formal graphical way (right).

The parameter vector π is calculated as was done for the HMM.

3. The proposed model: FNH-HMM double net

In our framework, O is formed by J observation *sequences* or *samples*. Each sample represents the genotype of the jth human subject, i.e. a sequence of N allele pairs; each k th SNP, $k = 1, \ldots, N$, is formed by unordered variable pairs $\{x_k, y_k\}$ taking values from $\{A, C, G, T\}$.

In each sample, the values of every k th SNP are considered to be generated by two different *hidden* random variables, called also hidden states, here indicated with s_k and t_k. These variables can take index values $1, \ldots, L$. An ordered sequence of N hidden states $< s_1, \ldots, s_k, \ldots, s_N >$ $(< t_1, \ldots, t_N >)$, with the same index value builds an *ancestral pattern*. The hidden states s_k and t_k take the pattern index values by considering a first-order Markov property, i.e. considering the previous states, s_{k-1} and t_{k-1}, respectively (Fig. 2 c, step 1, *Pattern choice*).[2] The generation of a particular nucleotide symbol x_k and y_k by a given hidden state occurs by means of an *emission* distribution related to the state s_k and t_k, respectively (Fig. 2 c, step 1, *Symbols emission*).

Now, in order to simulate the allele phasing that produces the final SNP pair, we add a switch variable m_k that decides the order (the phase) of the alleles (Fig. 2 c, step 2, *Symbols switch*). This is a key aspect that distinguishes our approach from the others based on HMM for haplotype reconstruction [5—7]. The addition of the switch variable makes the generative process underlying our model easy and intuitive to understand. On the other hand, the addition of the switch variable makes the application of the classical EM algorithm for model learning very expensive. In our case, the problem is formally solved through the variational learning approach, that additionally overcomes the classical problem of the EM of being highly susceptible to the parameter initialization. Such problems afflict other HMM-based approaches, thus forcing repeated applications of the training algorithm. We call our model FNH-HMM *double net*, hereafter simply *double net*, which is depicted in a formal way in Fig. 2 b.

Another way to understand the generative behavior of our model can be observed in Fig. 3, where the generative process is shown for a portion of a genotype. In practice, the proposed model connects two different FNH-HMMs, where the connection holds at the observation level.

Formally, the double net has a joint distribution formed by the two sets of variables, $h = \{m_k, s_k, t_k\}$ and $v = \{x_k, y_k\}$, for each SNP k and for each sample j. The model parameters are the transition and emission matrices of the two chains and the initial state distribution $\theta = \{\mathbf{A}_{c,k}, \mathbf{B}_{c,k}, \pi_c\}$, for each $k = 1 \ldots N$. The pedix c discriminates between the parameters of the two chains, i.e. if $c = L$ it addresses the lower-chain parameters, if $c = U$ it addresses the upper-chain parameters (see Fig. 2c).

First of all, it is reasonable to consider the samples as i.i.d., so we consider only the joint distribu-

[2] Choosing the "right" L is an issue not faced here. Driven by biological considerations, we try $7 \leq L \leq 10$ obtaining results very similar in quality. Anyway, we set $L = 7$, with which we gather the best haplotype reconstruction quality measures.

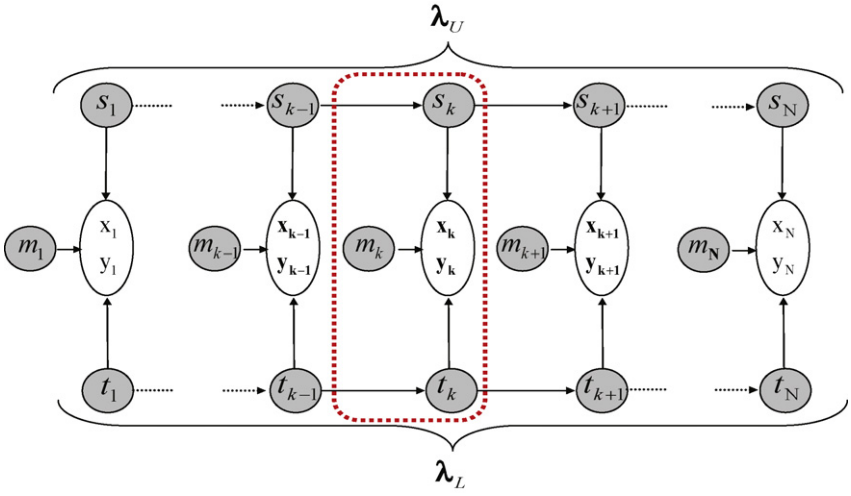

Figure 3 Alternative FNH-HMM double net representation: the two non-homogeneous hidden Markov chains are coupled at emission level. The dotted box refers to part of the model shown in Fig. 2 b.

tion over a single sample, thus dropping the apex (j). Therefore, we obtain

$$\begin{aligned} P(x,y,s,t,m) = {} & P(x_1,y_1|m_1,s_1,t_1)\,P(m_1)\,P \\ & \times (s_1)\,P(t_1)\prod_{k=2}^{N}[P(m_k)\,P \\ & \times (x_k,y_k|m_k,s_k,t_k)\,P \\ & \times (s_k|s_{k-1})\,P(t_k|t_{k-1})] \end{aligned} \tag{11}$$

In the equation above, the connection at the observation level between the FNH-HMMs is easily recognizable due to the term $P(x_k,y_k|m_k,s_k,t_k)$. The other terms indicate the switching variable probability, $P(m_k)$; the intra-chain transition probabilities, $P(s_k|s_{k-1})$ and $P(t_k|t_{k-1})$; and the initial state probabilities, $P(s_1)$ and $P(t_1)$.

The emission distribution can be further factorized, clarifying the meaning of the switching variable $m_k \in \{0,1\}$, which determines the phase of the chromosome pair $\{x_k,y_k\}$. If $m_k = 1$, the state s_k (t_k) generates the symbol x_k (y_k), otherwise, $m_k = 0$. This brings us to:

$$\begin{aligned} & P(x_k,y_k|m_k,s_k,t_k) \\ & = (P(x_k|s_k)P(y_k|t_k))^{m_k}\,(P(y_k|s_k)P(x_k|t_k))^{1-m_k} \end{aligned} \tag{12}$$

yielding to the following joint distribution

$$\begin{aligned} & P(x,y,s,t,m) \\ & \quad = (P(x_1|s_1)\,P(y_1|t_1))^{m_1}\,(P(y_1|s_1)\,P(x_1|t_1))^{1-m_1} \\ & \quad \times P(s_1)\,P(t_1)\,P(m_1)\prod_{k=2}^{N}[P(m_k)\,(P(x_k|s_k)\,P(y_k|t_k))^{m_k} \\ & \quad \times (P(y_k|s_k)\,P(x_k|t_k))^{1-m_k}\,P(s_k|s_{k-1})\,P(t_k|t_{k-1})]. \end{aligned}$$

4. Variational inference and learning

The free energy (see Section 2.1) of our model can be written by taking into account a generic form of $\mathcal{Q}(h)$

$$\mathcal{Q}(h) = \delta(\theta-\hat{\theta})\prod_{j=1}^{J} q(\{m_k^{(j)},s_k^{(j)},t_k^{(j)}\}_{k=1}^{N}|\theta) \tag{13}$$

where δ is the Dirac function. The equation above means that each sample is considered independent. Therefore, the free energy is

$$\begin{aligned} \mathcal{F} = \sum_{\text{samples}} \Bigg\{ & \sum_{m_k,s_k,t_k} q(\{m_k,s_k,t_k\}|\theta) \\ & \log\frac{q(\{m_k,s_k,t_k\}|\theta)}{P(\{x_k,y_k,m_k,s_k,t_k\}|\theta)} \Bigg\} \end{aligned} \tag{14}$$

To make the inference easier to handle and to make it tractable, we use the following constrained form of the function $\mathcal{Q}(h)$, where we employ a factorization of several multinomial distributions:

$$\begin{aligned} & q(\{s_k\}_{k=1}^{N},\{t_k\}_{k=1}^{N},\{m_k\}_{k=1}^{N}) \\ & \quad = q(\{s_k\}_{k=1}^{N})\,q(\{t_k\}_{k=1}^{N})\prod_{k=1}^{N} q(m_k) \end{aligned} \tag{15}$$

Using this form, known as *mean-field* [21], we can write down the free energy and easily solve the optimization problem (minimizing $\mathcal{F}$). The mean-field approximation is widely used because of its simplicity: it consists in assuming all the *component* hidden RVs are independent, given the data. This permits the estimation to be insensible to parameter initialization and less prone to local minima [22], unlike the Expectation–Maximization (EM) algorithm [23]. The first advantage holds because,

in the classical EM, variables are connected through conditional distributions. Setting initial values for the parameters of such distributions assumes *a priori* knowledge about how strongly the causes that generated the data are interconnected among themselves (in our case, it would assume knowledge about the identity of the ancestral pattern at each state). In such cases, the learning operates by *revising* such guesses when faced with the data. In the mean field approximation, such strong *a priori* knowledge does not exist. A random initialization of each (decoupled) parameter allows the data to *build* (and not to revise) the value of the conditional distribution in a more effective way. This, in turn, also helps to avoid local minima. For further details, see [22].

In order to minimize $\mathcal{F}$, we use the variational Expectation–Maximization algorithm [12] which alternates in minimizing the free energy w.r.t. the $\mathcal{Q}(h)$ distribution while keeping fixed the parameters (*E-Step*), and using the statistics over $\mathcal{Q}(h)$ just collected (*M-Step*) w.r.t. the parameters θ. When updating $\mathcal{Q}$, the only constraint is that $\int_{h_i} q(h_i^{(j)}) = 1$ for each hidden variable h_i and for each sample j. This constraint can be easily accounted for by using Lagrange multipliers. The updating rules are simply obtained by setting the derivatives of $\mathcal{F}$ equal to zero.

In detail, the pseudo-code for the learning of our double net via the EM algorithm is shown below.

Initialization: Randomly choose values for the parameters θ, i.e. the emission and transition matrices of the two chains, and set $q(m_k^{(j)}) = 0.5$ for each sample j and for each position k.
E-Step: *Minimize* $\mathcal{F}$ *with respect to* $q(h_i^{(j)})$, *forbi each sample* j *and for each hidden variable* h_i, *keeping fixed the values* θ. This is done using the following updating rules:

$$\frac{\partial \mathcal{F}}{\partial q(s_k = i)} = 0 \quad \rightarrow \quad q(s_k = i) = p(s_k = i | \{x_k, y_k\}_{k=1}^N) = \gamma_{U,k}(i) \tag{16}$$

$$\frac{\partial \mathcal{F}}{\partial q(t_k = i)} = 0 \quad \rightarrow \quad q(t_k = i) = p(t_k = i | \{x_k, y_k\}_{k=1}^N) = \gamma_{L,k}(i) \tag{17}$$

where $\gamma_{U,k}(i)$ and $\gamma_{L,k}(i)$ are the probability of being in state i at time k given the observation O and the model λ ($P(s_k = i|O, \lambda)$) in the respective chain, see Section 2.3. The updating rules can be derived from the forward and backward variables α and β noting that, in the FNH-HMM, we have

$$\gamma_k(i) = \frac{\alpha_k(i)\,\beta_k(i)}{P(O|\lambda)} \tag{18}$$

where the forward and backward variables are calculated using the following weighted log-likelihood

$$\log p(x_k, y_k | s_k = i) = q(m_k = 1) \cdot b_{U,k}^i(x_k) + q(m_k = 0) \cdot b_{U,k}^i(y_k) \tag{19}$$

$$\log p(x_k, y_k | t_k = l) = q(m_k = 1) \cdot b_{L,k}^l(y_k) + q(m_k = 0) \cdot b_{L,k}^l(x_k) \tag{20}$$

Update the distributions over a mask variable as follows:

$$\frac{\partial \mathcal{F}}{\partial q(m_k = 1)} = 0 \rightarrow q(m_k = 1) \propto \exp\left(\sum_{s_k} q(s_k) \log b_{U,k}^{s_k}(x_k) + \sum_{t_k} q(t_k) \log b_{L,k}^{t_k}(y_k) \right) \tag{21}$$

$$\frac{\partial \mathcal{F}}{\partial q(m_k = 0)} = 0 \rightarrow q(m_k = 0) \propto \exp\left(\sum_{s_k} q(s_k) \log b_{U,k}^{s_k}(y_k) + \sum_{t_k} q(t_k) \log b_{L,k}^{t_k}(x_k) \right) \tag{22}$$

$q(m_k = 1)$ and $q(m_k = 0)$ are then normalized at every site k.

M-Step: *Minimize* $\mathcal{F}$ *with respect to the model parameters* θ *setting the derivative* $(\partial \mathcal{F}/\partial \theta) = 0$. Using the convex combination of the observation likelihoods (Eq. (19)), we can decouple the two FNH-HMMs and update independently the parameters of the two chains λ_L and λ_U using the standard Baum–Welch algorithm for FNH-HMMs.

At this point, it is worthy to note other differences of our approach w.r.t the most similar approaches in the literature: [5–7]. In [6], the haplotypes and the blocks are estimated simultaneously and the approach computes the maximum likelihood directly (i.e. it is not based on a Bayesian network), making the system really prone to local minima. In [7], the first-order Markov property that regulates the presence of a particular ancestral pattern is relaxed, bringing to a less accurate haplotype estimation as noted by the authors. In [5], within the context

of a similar generative framework, haplotypes are estimated after the Viterbi calculation of the ancestral pattern identities and, due to the absence of the variational trick, the computational load for the learning step is $O(\mathrm{JNL}^3)$. In our case, it is $O(\mathrm{JNL}^2)$, basing on FNH-HMM inference strategies.

5. Haplotype reconstruction

Once the model is learned, it is easy to reconstruct the haplotypes from the genotypes. For each genotype $O^{(j)} = \langle (x_1, y_1), \dots (x_k, y_k), \dots (x_N, y_N) \rangle$, the phase of each k th pair is given by the value of $q(m_k^{(j)} = 1)$: if it is larger than 0.5, then x_k belongs to the upper haplotype and y_k belongs to the lower, and vice versa if $q(m_k^{(j)} = 1) < 0.5$. Note that at each position we can easily understand the uncertainty with which the model estimated the haplotype. This could in principle lead to the creation of measures of confidence for our haplotype reconstruction. Basically, whereas the $q(m_k^{(j)} = 1)$ are pooled on 0s or 1s, strong certainty is associated to the related alleles phase.

6. Linkage disequilibrium block discovery

As mentioned in Section 3, the hidden patterns $1, \dots, L$ model ancestral haplotype sequences which have been fragmented and recombined throughout human history, producing all the observed haplotypes. As a first step toward the block discovery, we estimate the most probable pathway through these hidden patterns for each reconstructed haplotype sequence. This is done with a straightforward non-homogeneous version of the Viterbi algorithm [11], paying attention to use the log-likelihood previously introduced in Eq. (19). In this way, we account for all the uncertainty of the haplotype reconstruction in the block discovery task.

All the Viterbi paths (Fig. 4 a) are then placed on a lattice $L \times N$ (Fig. 4b). In this way, at each allele site k we can distinguish W_k distinct paths, each one indicated with $w_k(i), i = 1, \dots, W_k \leq L$; $\|w_k(i)\|$ indicates the number of haplotypes traversing $w_k(i)$ (see Fig. 4b).

We are now able to perform block discovery. The idea is that if two paths $w_k(i)$ and $w_k(i')$ do join at site $k+1$, they represent two sets of haplotypes with highly different haplotype fragments up to k, becoming similar after k. Therefore, a block boundary exists between k and $k+1$. Similar reasoning holds for a split site (see Fig. 4b). We translate this intuition with the *boundary presence strength* measure $\Omega(k, k+1) \in [0, 1)$, which models the existence of a block boundary between sites k and $k+1$, i.e.

$$\Omega(k, k+1) = \mathbf{1}_{\mathrm{Join}}(k)\, G(k) + \mathbf{1}_{\mathrm{Split}}(k+1)\, G(k+1) \tag{23}$$

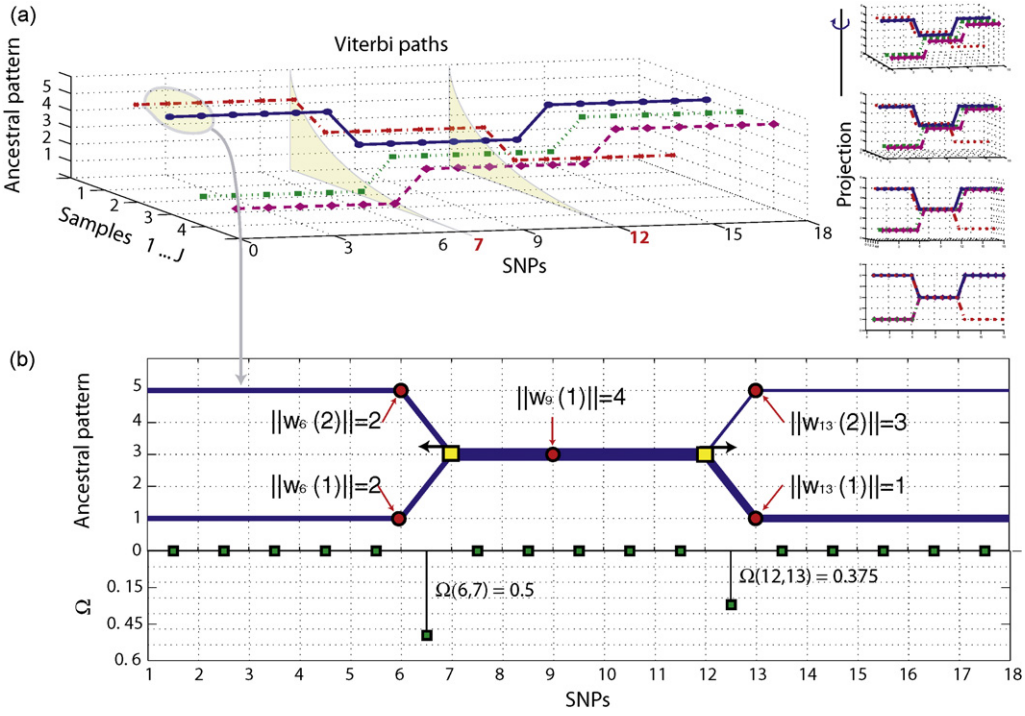

Figure 4 Toy example (J=4 haplotypes): (a) Viterbi paths; (b) paths over the lattice structure and the relative Ωplot; note that (b) is projection of (a) over the SNPs—"Ancestral Pattern" plane.

where $\mathbf{1}_{\text{Join}}(k)$ ($\mathbf{1}_{\text{Split}}(k+1)$) is equal to 1 when at least one join (split) is present at time k ($k+1$), and $G(\cdot)$ is the *Gini* index [14]:

$$G(k) = 1 - \sum_{i=1\ldots W_k} \left(\frac{\|w_k(i)\|}{W_k} \right)^2 \tag{24}$$

The Gini index can be used to describe whether a graph join or split is well balanced or not. For example, a split at site k is well balanced if the cardinalities $\{\|w_k\|\}$ of the child paths $\{w_k\}$ are similar. The idea is that the higher the $\Omega(k)$, the more likely the presence of a block boundary between site k and $k+1$. On the other hand, a low $\Omega(k)$ means that in the join (split) site k, a dominant path (i.e. with a high number of haplotypes associated) merges (splits) with one or more irrelevant paths (see Fig. 4b). Given a threshold τ_Ω, we can assign a block boundary to the site k when $\Omega(k) > \tau_\Omega$.

It is valuable to note that all the block boundary measures in the literature [24] are strongly dependent on the accuracy with which the haplotypes are estimated, but there is no way to codify formally this dependency. In our approach, instead, we calculate for a given set of simulated genotypes (with the positions of blocks boundaries *a priori* estimated) of length N, the mean log-likelihood LL_m that results from the learning step. This represents a likelihood-based quality of fit criterion for the data, given the model. Then, we evaluate using the ground truth information about the "optimal" value of τ_Ω in order to minimize the error of block discovery for each genotype produced. We average the values obtaining Ω_m. By varying the length of the data we can build a look-up table of Ω_m, LL_m, and N. Therefore, when a novel dataset is available (length N), if the log-likelihood obtained during the training is higher than the one precalculated for that data length, we can inherit the correspondent τ_Ω. Otherwise, we have to be more conservative and must choose an higher value for τ_Ω. Note that all the data have been processed by fixing the number of ancestral haplotypes to $\tilde{L} = 7$. The value of $\tilde{L}$ has been chosen by maximizing on the haplotype reconstruction quality measures (see next section).

7. Experimental results

7.1. Haplotype reconstruction

To evaluate the proposed approach, our framework has been extensively tested on different data sets and compared to five other state-of-the-art systems.

Concerning the initialization issue, we again remark that no *a priori* knowledge has been used (random parameter initialization), and that all the results reported here derive from standard executions of the two inference strategies proposed here.

In the first two experiments, we compared the FNH-HMM double net to various haplotype reconstruction systems in terms of various quality measures. These tests show how our method successfully identifies the largest number of correct haplotypes, while keeping the number of incorrect haplotypes inferred and the correct haplotype frequencies in line with the other state-of-the-art systems. This observation may have practical value if our problem regards the functional genetics, where the issue is identifying the largest number of correct haplotypes, instead of minimizing the number of incorrect haplotypes inferred.

The first data set is taken from the hapmap project [25] from chromosome 7[3] from SNP marker *rs*323917 to SNP *rs*324375. The reconstructed haplotype frequencies are summarized in Table 1, where *Emp* stands for the empirical frequencies, *PhLD* stands for *Phase* with linkage disequilibrium [3], *fPh* stands for *fastPhase* [7], *SPHP* stands for *SNPHAP* [10], *Ger* stands for *Gerbil* [6], *Hit* stands for *Hit* [5] and *Double net* stands for our method. Table 3 summarizes the statistics in terms of various measures of quality. In particular, *haplotype frequency estimation* (I_F) and *haplotype identification index* (I_H), proposed in [26], are two appropriate quality measures for the haplotype reconstruction task.

The haplotype frequency estimation (I_F) is a measure defined as the proportion of haplotype frequencies in common between the estimated and the true haplotypes

$$I_F = 1 - \frac{1}{2} \sum_{\text{haplotypes}} |p_{e,k} - p_{t,k}| \tag{25}$$

where $p_{e,k}$ and $p_{t,k}$ are the estimated and the true haplotype frequency of the k th haplotype, respectively.

The index I_F varies between zero, when true haplotypes have estimated frequencies approaching zero, and one, when observed and estimated frequencies are identical. The index weights more heavily the high-frequency haplotypes.

A second commonly used index is the haplotype identification index I_H, defined as

$$I_H = \frac{2(m_{\text{true}} - m_{\text{missed}})}{m_{\text{true}} + m_{\text{estimated}}} \tag{26}$$

[3] Caucasoid *r* 21 phaseII.

Table 1 Haplotype frequencies obtained with a training set composed by 60 genotypes of 25 SNPs. Bold numbers indicate the best reconstruction

SNP																	Haplotype frequency						
Empirical haplotypes																	Algorithm						Double net
																	Emp	PhLD	fPhs	SPHP	Ger	Hit	
1	C	A	C	G	C	C	C	T	A	T	G	T	T	A	G	A	C	T	C	A	G	G	T	T	A	0.475000	0.4760	0.4833	0.4831	**0.4750**	**0.4750**	**0.475000**
2	C	G	T	A	T	G	C	T	A	T	G	T	C	G	G	A	A	C	T	C	T	A	A	C	A	0.183333	**0.1833**	**0.1833**	**0.1833**	**0.1833**	**0.1833**	**0.1833**
3	C	G	T	A	T	G	C	T	A	T	G	T	C	G	G	A	C	T	C	A	A	G	C	T	A	0.116667	**0.1167**	**0.1167**	**0.1167**	0.1000	0.1000	0.109999
4	G	A	C	G	T	C	C	T	A	T	G	T	C	A	G	A	C	T	C	A	A	G	C	A	A	0.066667	**0.0667**	**0.0667**	0.0663	0.0583	0.0583	**0.066667**
5	C	A	C	G	C	C	C	C	A	T	G	T	T	A	G	A	C	T	C	A	G	G	T	T	A	0.058333	**0.0583**	**0.0583**	0.0579	0.0500	0.0500	0.056754
6	C	A	C	G	T	C	C	T	A	T	A	T	C	G	G	A	A	C	T	C	A	A	G	C	T	0.041667	0.0407	0.0333	0.0340	**0.0417**	**0.0417**	0.039054
7	C	G	T	A	T	G	C	C	A	T	G	T	C	G	G	A	A	C	T	C	T	A	A	C	A	0.025000	**0.0250**	**0.0250**	**0.0250**	**0.0250**	**0.0250**	0.018000
8	C	A	C	G	C	C	C	C	T	A	T	G	T	T	A	G	G	A	C	T	T	A	G	T	T	0.016667	0.0157	0.0083	0.0090	**0.0167**	**0.0167**	0.014444
9	C	G	T	A	T	G	C	T	A	T	A	T	C	G	G	A	A	C	T	C	A	A	G	C	T	0.008333	**0.0083**	**0.0083**	**0.0083**	**0.0083**	**0.0083**	0.007566
10	C	A	C	G	C	C	C	T	A	T	G	T	T	A	G	A	C	C	C	C	A	G	G	T	T	0.008333	**0.0083**	**0.0083**	**0.0083**	**0.0083**	**0.0083**	**0.0083**

where m_{true} is the number of true haplotypes in the sample, $m_{estimated}$ is the number of estimated haplotypes, and m_{missed} is the number of true haplotypes not identified in the sample. The value of I_H can vary between one, when the identified haplotypes are exactly those present in the true sample, and zero, when none of the true haplotypes has been identified.

As a second test, we choose a dataset available on demand, taken from the pituitary growth hormone (GH1) [27]. The five genes of the human growth hormone locus reside within about 45 kilobases (kb) on chromosome 17, and the GH1 is by far the most thoroughly studied gene. It is unusually polymorphic, with 16 SNPs having been identified in a span of 535 base-pairs. The data is taken from the sequencing of 154 recruits of the British army. Using this data, Horan et al. [28], empirically determined 36 haplotypes. In our experiment, we consider this data as ground-truth. The empirical haplotype frequencies exhibit considerable dispersion (see Table 2, column Emp): two haplotypes are relatively common with frequencies of 33% and 16%, 31 have frequencies below 5% and 19 haplotypes have frequencies less than 1%. Subsequently, Adkins [27] compared five haplotype reconstruction algorithms on the same dataset.

The reconstructed haplotype frequencies are summarized in Table 2, where we reported the results obtained by Adkins [27]. We added four comparisons to the systems analyzed in the first experiment (see Table 1). Concerning the haplotype reconstruction systems in [27] in Table 2, *PhNLD* stands for *Phase* with no linkage disequilibrium taken into account [3], *PhLD* for *Phase* with linkage disequilibrium [3], *HPLT* stands for *H* aplotyper [8], *PLEM* stands for the partition–ligation expectation–maximization algorithm [9], and *SPHP* stands for *SNPHAP* [10].

The statistics in terms of I_F and I_H are summarized in Table 3, showing that all the methods have good performances in terms of frequencies in both tests. However our method always identifies the greatest number of haplotypes. This results brings about the best I_H index.

7.2. Haplotype block partitioning

Concerning the block discovery, no "formal" ground-truth data is present in literature for the block estimation. Nevertheless, universally accepted measures and algorithms exist that have produced results that are considered as ground-truth [29,30]. Here, for comparison, we use one of the most well known methods to individuate blocks: the Gabriel method [1]. It is worth noting

that, as a drawback, the Gabriel method is *scarcely* robust to reconstruction errors.

The first dataset used is taken from chromosome 11 on the fads 1, 2, 3, genes (chr11q12.13) [31]. It is valuable to note that for this test we could not use the pituitary growth hormone (GH1) dataset because no relevant block structure is present (see [27]).

Pairwise LD table (LD plot) is a widely used data structure for block discovery. The pairwise measures D' [32](Fig. 5 a — left diagonal elements) and r^2 [33](Fig. 5 a — right diagonal elements) build the LD table. A high D' (r^2) value in position m, n (or n, m) indicates a block relation between the site m and n.

The pairwise LD plot for the dataset considered is shown in Fig. 5 a. Using the Gabriel method, we obtain three major blocks highlighted in the figure with yellow squares. The first block ranges from the first SNP to the fourth, the second consists of SNPs from six to eight, and the last consists of the last four SNPs. Since a LD block is a group of consecutive highly correlated SNPs, one can intuitively be convinced of the presence of the first block looking only at the D' measure and noticing that the first four SNPs present a high value (≈ 1) among one another.

With our approach, we first calculate the Viterbi path for each haplotype, that is the most likely sequence of hidden state values that led to that particular observation [11]. Subsequently, we project all the paths over the lattice structure determined by the *SNPs—Ancestral pattern* plane (Fig. 5b), calculating the number of paths that share the same path segment as described in Section 6.

At this point, we focus our attention on the *split* and *join* points, highlighted in Fig. 5 b with a square.

Using Eq. (23), we can easily calculate the block boundary strength $\Omega(k, k+1)$ for each couple of consecutive SNPs, which is higher when more balanced intersections are present. Fig. 5 c shows the resulting Ω-value: as expected, a high Ω value is present in correspondence with SNPs $(5, 6)$ due to the presence of many splits at SNP 5, and many joins at SNP 6. The same holds for SNPs number $(8, 9)$.

Considering our look-up table of τ_Ω values, we threshold the value of Ω with $\tau_\Omega = 0.3$, obtaining the block structure reported in Fig. 5 d. Other block discovery results are shown in Fig. 6. Here, the data is randomly taken from the Hapmap project. In Fig. 6 b, the Pairwise LD table is reported. The table, built using empirical haplotypes, confirms the block division presented in Fig. 6 a, obtained after solving the reconstruction task with FNH-HMM double net and calculating the block boundary strength Ω.

The fourth data set used consists of 11 SNPs taken from the interlukin-1 cluster on human chromosome

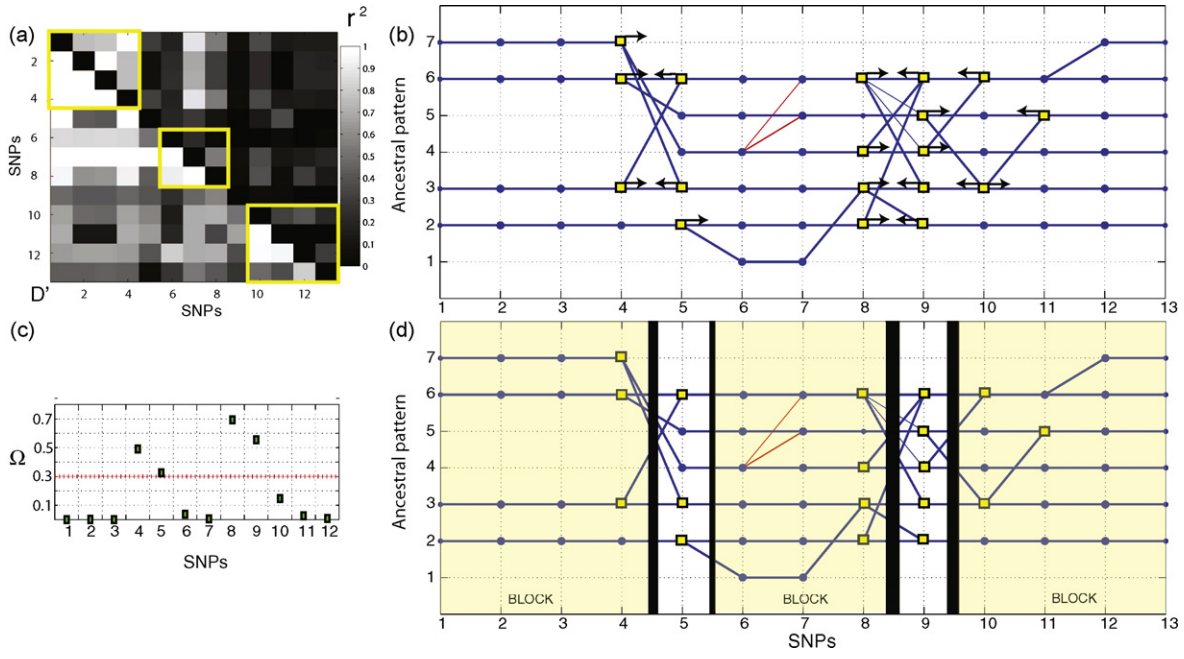

Figure 5 (a) Pairwise LD plot. The blocks are highlighted with a yellow square; (b) Viterbi paths over the ancestral pattern. Splits/joins are indicated with a square. The arrows indicate where the contributions of the split/joins point votes for a block boundary. In fact, as shown in Eq. (23), a join at position k increases the block presence strength between the $k-1$ th SNP and the k th SNP, while a split at k increases the block presence strength between k and $k+1$; (c) the Ω plot; (d) resulting blocks for the chr11q12.13 dataset with a threshold $\tau = 0.3$. The thickness of the boundaries is proportional to the value of the block presence strength Ω.

Table 2 Haplotype frequencies. Bold numbers indicate the best reconstruction

	SNP															Haplotype frequency									
	Empirical haplotypes															Algorithm									Double net
																Emp	PhNLD	PhLD	HPLT	PLEM	SPHP	fPh	Ger	Hit	
1	G	G	G	G	G	T	A	T	G	A	A	G	A	A	T	0.334	0.312	0.321	0.325	**0.333**	0.332	0.320	0.326	0.331	0.305
2	G	G	G	G	T	T	A	G	G	G	A	G	A	A	T	0.162	0.166	**0.162**	0.166	0.181	0.171	0.168	0.1536	0.166	**0.162**
3	G	G	T	T	G	T	A	G	G	A	A	G	A	A	T	0.091	0.097	0.097	0.101	0.098	0.102	0.092	**0.088**	0.098	0.123
4	G	G	T	T	G	T	A	G	–	A	A	G	A	A	T	0.052	0.055	0.055	0.049	0.047	**0.050**	0.059	0.059	0.055	0.040
5	G	G	G	G	T	T	G	G	G	G	A	G	A	A	T	0.042	0.052	0.052	0.052	**0.049**	0.050	0.056	0.069	0.052	0.057
6	G	G	T	T	G	T	A	G	–	A	A	G	A	A	G	0.029	0.032	0.032	0.032	**0.030**	**0.030**	0.033	0.033	0.033	0.016
7	G	G	G	G	T	T	A	G	G	G	T	G	A	A	T	0.026	0.032	0.032	0.032	**0.028**	0.029	0.033	0.036	0.029	0.039
8	G	G	T	T	G	T	A	G	G	G	A	G	A	A	T	0.019	0.016	0.016	0.013	0.016	0.018	**0.019**	0.023	0.016	**0.019**
9	G	G	G	G	T	T	A	T	G	G	A	G	A	A	T	0.019	0.013	0.013	0.013	0.011	0.011	0.013	0.013	0.013	**0.016**
10	G	G	T	T	G	T	A	G	–	G	A	G	A	A	T	0.019	0.023	0.026	0.023	0.025	0.025	0.023	**0.019**	0.026	0.010
11	G	G	G	G	T	T	G	G	G	G	A	G	G	C	T	0.016	**0.016**	**0.016**	**0.016**	0.014	0.014	0.013	**0.016**	0.013	0.010
12	G	G	G	G	T	T	A	G	G	A	A	G	A	A	T	0.016	**0.010**	0.006	0.006	0.008	0.008	0.006	**0.010**	0.006	**0.010**
13	G	–	G	G	T	T	G	G	G	G	A	G	A	A	T	0.016	**0.016**	**0.016**	0.013	0.010	0.013	**0.016**	**0.016**	**0.016**	**0.016**
14	G	G	G	G	T	C	A	G	G	G	T	G	A	A	T	0.016	**0.016**	**0.016**	**0.016**	**0.016**	**0.016**	**0.016**	**0.016**	**0.016**	**0.016**
15	G	G	T	T	G	T	A	G	G	G	T	G	A	A	T	0.013	**0.010**	**0.010**	0.010	0.006	0.009	0.006	0.006	**0.010**	0.006
16	G	G	G	G	T	T	G	G	G	A	A	G	A	A	T	0.013	**0.013**	**0.013**	0.016	0.008	0.008	**0.013**	0.010	0.010	0.010
17	G	–	G	G	T	T	A	G	G	G	A	G	A	A	T	0.013	**0.013**	**0.013**	**0.013**	0.011	0.011	0.010	**0.013**	0.010	**0.013**
18	G	G	G	G	T	T	A	G	–	G	A	G	A	A	T	0.010	–	0.006	**0.010**	0.007	0.008	0.006	0.006	0.006	**0.010**
19	A	G	G	G	T	T	A	G	G	G	A	G	A	A	T	0.010	0.013	0.013	**0.010**	0.005	**0.010**	**0.010**	0.013	**0.010**	**0.010**
20	G	G	G	G	G	T	A	G	–	A	A	G	A	A	T	0.010	–	0.003	**0.010**	0.006	0.005	0.003	0.003	0.003	0.003
21	G	G	G	G	T	T	G	G	G	G	A	G	A	A	G	0.010	**0.010**	**0.010**	**0.010**	0.011	0.011	0.006	**0.010**	**0.010**	0.016
22	G	G	T	T	G	T	A	T	G	A	A	G	A	A	T	0.010	0.013	**0.010**	0.013	0.007	0.007	**0.010**	0.013	0.006	0.012
23	G	G	G	G	G	T	A	G	G	A	A	G	A	A	T	0.006	0.016	0.013	**0.006**	**0.006**	0.008	0.010	0.010	0.010	0.010
24	G	G	T	T	G	T	G	G	–	A	A	G	A	A	T	0.006	–	–	–	–	–	–	–	–	–
25	G	G	T	T	G	T	A	G	G	A	A	G	A	A	G	0.003	–	–	–	0.004	0.004	0.006	0.006	**0.003**	0.010
26	G	G	G	G	T	T	G	G	G	G	T	G	A	A	T	0.003	**0.006**	**0.006**	**0.006**	0.007	0.007	**0.006**	–	**0.006**	**0.006**
27	G	G	G	G	T	T	A	T	G	A	A	G	A	A	T	0.003	**0.003**	**0.003**	–	–	–	–	**0.003**	–	–
28	G	G	G	G	T	T	A	G	–	A	A	G	A	A	T	0.003	–	–	–	–	–	**0.003**	**0.003**	–	**0.003**
29	A	G	G	G	T	T	A	G	G	A	A	G	A	A	T	0.003	–	–	–	–	–	–	–	–	–
30	G	–	G	G	T	T	A	G	G	A	A	G	A	A	T	0.003	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
31	G	G	G	G	T	T	G	G	–	G	A	G	A	A	T	0.003	0.010	–	–	–	–	**0.003**	0.006	**0.003**	**0.003**
32	G	G	T	T	G	T	G	G	G	G	A	G	A	A	G	0.003	–	–	–	0.002	–	–	–	–	**0.003**
33	G	G	G	G	T	T	A	G	G	G	A	G	G	C	T	0.003	**0.003**	**0.003**	**0.003**	0.004	0.004	**0.003**	**0.003**	**0.003**	0.010
34	G	–	G	G	T	C	A	G	G	G	T	G	A	A	T	0.003	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
35	G	G	G	G	G	T	A	G	G	A	C	C	A	A	T	0.003	–	–	**0.003**	**0.003**	**0.003**	–	–	–	**0.003**
36	G	G	G	G	T	T	A	G	G	G	T	G	A	A	G	0.003	–	–	–	**0.003**	**0.003**	**0.003**	–	**0.003**	**0.003**

Table 2 (*Continued*)

	SNP															Haplotype frequency									
	Empirical haplotypes															Algorithm									Double net
																Emp	PhNLD	PhLD	HPLT	PLEM	SPHP	fPh	Ger	Hit	
	Incorrect haplotypes																								
1	G	–	G	G	T	T	G	G	G	G	A	G	G	C	T	–	–	–	–	0.002	0.002	0.003	–	0.003	–
2	G	G	G	G	T	T	G	G	–	A	A	G	A	A	T	–	–	0.006	–	0.009	0.008	–	–	0.003	–
3	G	G	G	G	T	T	G	T	–	G	A	G	A	A	T	–	–	–	–	–	0.002	–	–	–	–
4	G	–	G	G	T	T	G	T	G	A	A	G	A	A	T	–	–	–	0.003	–	–	–	–	–	–
5	G	–	G	G	T	T	G	T	G	G	A	G	A	A	T	–	–	–	–	0.003	0.002	–	–	–	–
6	G	G	G	G	T	T	G	T	G	G	A	G	A	A	T	–	–	0.003	–	–	–	–	–	–	–
7	G	G	G	G	T	T	G	T	G	A	A	G	A	A	T	–	–	–	–	–	0.004	–	–	0.006	–
8	A	G	G	G	T	T	A	T	G	A	A	G	A	A	T	–	–	–	0.003	0.003	0.003	0.003	–	0.003	0.003
9	G	G	G	G	T	T	A	G	G	G	C	C	A	A	T	–	0.003	0.003	–	–	–	–	–	–	–
10	A	G	T	T	G	T	A	G	–	A	A	G	A	A	T	–	–	–	–	0.002	–	–	–	–	0.003
11	A	G	T	T	G	T	A	G	G	G	T	G	A	A	T	–	–	–	–	0.003	–	–	–	–	0.003
12	G	G	T	T	T	T	G	G	–	G	A	G	A	A	T	–	–	–	0.006	0.004	–	–	–	–	–
13	G	G	T	T	T	T	G	G	G	A	A	G	A	A	T	–	–	–	–	0.003	–	–	–	–	–
14	G	G	G	G	G	T	A	T	–	A	A	G	A	A	T	–	0.010	–	–	–	–	0.006	–	–	–
15	G	G	G	G	G	T	A	T	G	G	A	G	A	A	T	–	0.006	0.006	0.006	0.008	0.008	–	0.003	0.006	0.006
16	G	G	G	G	G	T	A	T	G	A	A	G	A	A	G	–	0.006	0.006	0.006	–	–	–	0.010	–	0.010
17	G	G	T	T	G	T	A	T	–	A	A	G	A	A	T	–	–	–	–	–	–	–	–	–	0.006
	Minor allele frequency																								
	0.01		0.25		0.40		0.11		0.13		0.06		0.02		0.05										
		0.03		0.25		0.02		0.37		0.41		0.003		0.02											

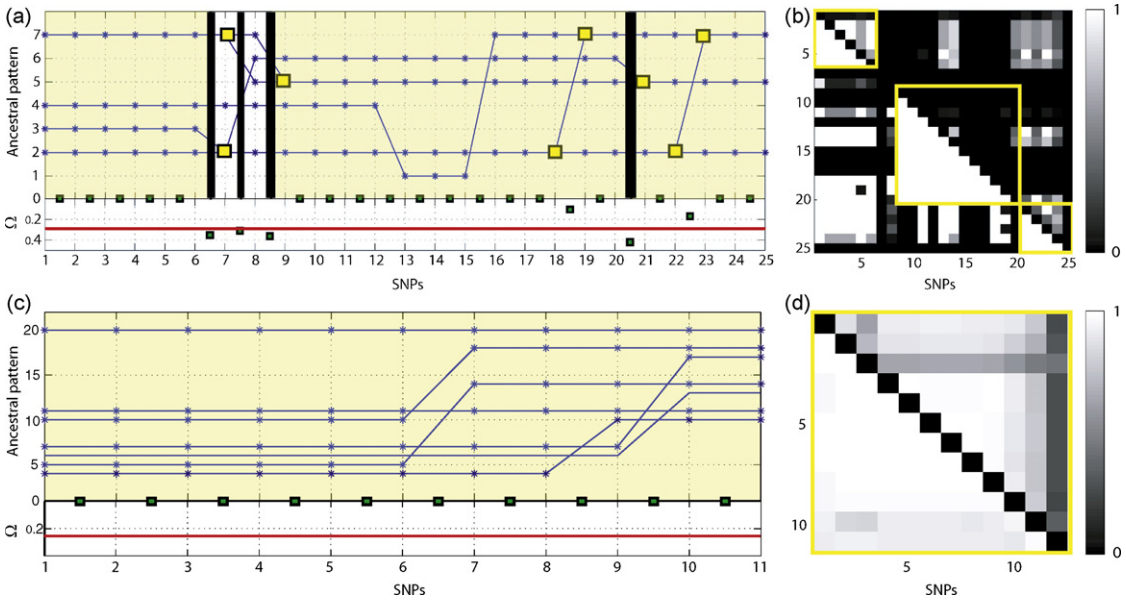

Figure 6 (a) and (c) Viterbi paths over the lattice structure (top) and correspondent Ω plot (bottom). Splits/joins are indicated with yellow rectangles; block boundaries are shown with a bar whose thickness is proportional to the Ω value; (b) and (d) pairwise LD table: D' (left diagonal elements) and r^2 (right diagonal elements) values confirm the block boundaries found with our method.

Table 3 Accuracy of inference of haplotype structure on the GHI gene promoter

Algorithm	ch17-GH1 [27]			Hapmap sample	
	I_H	I_F	# correct / # wrong	I_H	I_F
Phase v2 no LD	0.81	0.91	27/4	0.9524	0.9878
Phase v2 with LD	0.81	0.93	28/5	—	—
Haplotyper 1.0	0.81	0.93	28/5	—	—
PL-EM 1.0	0.82	0.92	31/9	—	—
SNPHAP 1.0	0.82	0.93	30/7	0.9091	0.9878
fastPhase	0.87	0.93	31/3	0.9524	0.9874
Gerbil	0.88	0.93	30/2	0.9091	0.9832
HIT	0.84	0.94	30/5	0.9091	0.9832
FNH-HMM double net	0.88	0.90	33/6	0.9761	0.9853

correct and # wrong stand for respectively the number of corrected and wrong haplotypes inferred.

2q12-2q14 presented in [34]. In Fig. 6 c, the paths over the ancestral patterns inferred after the model training are depicted. No splits or joins are present, and thus only a haplotype block is present here, as confirmed by [34] and by the LD plot shown in Fig. 6 d.

8. Conclusions

Haplotype analysis is actually used in medical genetics to localize the genetic region containing susceptibility genes for genetic diseases. Therefore, haplotype frequency estimation and dissection in the LD structure of chromosomal regions are important tasks, since LD structures vary across the genome and among populations. For that reason, it is important that available computational tools are able to resolve the haplotype phase from unrelated individual genotypes and are able to identify suitable patterns of LD structures in the regions to be studied. In this paper, we proposed a generative framework based on hidden first-order Markov processes able to perform haplotype reconstruction and block discovery at the same time, using two model inferences. The model is based on a connection between two FNH-HMM. The model learning has been carried out under a variational context, and it relies essentially on two independent computations of the Baum–Welch algorithm. In this way, a fast inference procedure is obtained that is linear in the length of genotypes, insensible to model initialization, and less prone to local minima w.r.t those

approaches employing exact versions of the EM algorithm for model learning. The time complexities of the other efficient approaches given here for comparison are $O(\mathrm{JNL}^3)$ for *HIT*, $O(\mathrm{JNL}^2)$ for *fastPHASE* and for *Gerbil* complexity was not expressed analytically by the authors. The complexity of the other methods ranges between $O(N^2)$ and $O(N^3)$ making it prohibitively expensive to apply them on long haplotypes [6].

To validate the approach (1) we exhaustively compared our method with five other state-of-the-art systems for haplotype reconstruction in terms of various accuracy measures and (2) we tested the block discovery capability with a well-known block discovery method. The proposed method showed performances that are similar to, and for some measures even better than, the best known reconstruction methods. It is worth noting that our block discovery inference takes into account uncertainty in the haplotype reconstruction and imputation of haplotypes is based on a small number of core haplotypes. These features will be further investigated to develop a unified approach for fine linkage disequilibrium mapping of genes involved in complex diseases.

Future efforts will also be devoted to investigating an extension of our model that considers hidden Markov processes of order higher than one, evaluating the trade-off between the quality of the obtained results and the involved computational complexity.

References

[1] Gabriel B, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, et al. The structure of haplotype blocks in the human genome. Science 2002;296(5576):2225—9.

[2] Zhang K, Calabrese P, Nordborg M, Sun F. Haplotype block structure and its application to association studies: power and study designs. American Journal of Human Genetics 2002;71(6):1386—94.

[3] Stephens M, Donnelly P. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. American Journal of Human Genetics 2003;73(5): 1162—9.

[4] Chen Y, Lin CH, Sabatti C. Volume measures for linkage disequilibrium. BMC Genetics 2006;7(1):54—61.

[5] Rastas P, Koivisto M, Mannila H, Ukkonen E. A hidden markov technique for haplotype reconstruction. In: Casadio R, Myers G, editors. Algorithms in Bioinformatics, vol. 3692 of Lecture Notes in Computer Science. New York, NY, USA: Springer; 2005. p. 140—51.

[6] Kimmel G, Shamir R. Gerbil: Genotype resolution and block identification using likelihood. Proceedings of the National Academy of Science of the United States of America (PNAS) 2005;102:158—62.

[7] Scheet P, Stephens M. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. American Journal of Human Genetics 2006;78(4):629—44.

[8] Niu T, Qin ZS, Xu X, Liu JS. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. American Journal of Human Genetics 2002;78(1):157—69.

[9] Zhaohui TN, Qin ZS, Liu JS. Partition—ligation—expectation—maximization algorithm for haplotype inference with single-nucleotide polymorphisms. American Journal of Human Genetics 2002;71(5):1242—7.

[10] D. Clayton, Snphap: A program for estimating frequencies of large haplotypes of snps (version 1.0), http://www-gene.cimr.cam.ac.uk/clayton/software/snphap.txt, Accessed: 11 December 2007 (2001).

[11] Rabiner L. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of IEEE 1989;77(2):257—86.

[12] Jordan M, Ghahramani Z, Jaakkola T, Saul L. An introduction to variational methods for graphical models. Machine Learning 1999;37(2):183—233.

[13] Jojic V, Jojic N, Heckerman D. Joint discovery of haplotype blocks and complex trait associations from snp sequences. In: Chickering DM, Halpern JY, editors. Proceedings of the Uncertainty in artificial intelligence 2004, UAI'04. Arlington, Virginia: UAI Press; 2004. p. 286—92.

[14] Duda RO, Hart PE, Stork DG. Pattern Classification, 2nd Edition, Hoboken, NJ: Wiley—Interscience; 2000.

[15] Li W. Dna segmentation as a model selection process. In: Lengauer T, Sankoff D, Istrail S, Peuvzner P, Waterman M, editors. RECOMB'01: Proceedings of the fifth annual international conference on computational biology. New York, NY, USA: ACM; 2001. p. 204—10.

[16] Koivisto M, Perda M, Varilo T, Hennah W, Ekelund J, Lukk M, et al. An mdl method for finding haplotype blocks and for estimating the strength of haplotype block boundaries. In: Altman RB, Dunker AK, Hunter L, Jung TA, Klein TE, editors. Proceedings of the Pacific Symposium on Biocomputing (PSB), vol. 8. Singapore: World Scientific; 2003. p. 502—13.

[17] Rissanen J. Modelling by shortest data description. Automatica 1978;14:465—71.

[18] Ghahramani Z, Beal M. Graphical models and variational methods. In: Opper M, Saad D, editors. Advanced mean field methods: theory and practice. Cambridge, MA, USA: MIT Press; 2001. p. 161—78. Ch. 11.

[19] D. Heckerman, A tutorial on learning with Bayesian networks, Tech. Re MSR-TR-95—06, Microsoft Research, Redmond, WA, USA, revised November, 1996 (1995).

[20] Ghahramani Z. Learning dynamic Bayesian networks. In: Giles CL, Gori M, editors. Adaptive Processing of Sequences and Data Structures. Berlin: Springer-Verlag; 1998. p. 168—97.

[21] Jaakkola T. Tutorial on variational approximation methods. In: Opper M, Saad D, editors. Advanced mean field methods: theory and practice. Cambridge, MA, USA: MIT Press; 2001. p. 129—60. Ch. 10.

[22] Jojic N, Winn J, Zitnick L. Escaping local minima through hierarchical model selection: automatic object discovery, segmentation, and tracking in video. In: CVPR'06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington, DC, USA: IEEE Computer Society; 2006 . p. 117—24.

[23] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 1977;39:1—38.

[24] Mueller JC. Linkage disequilibrium for different scales and applications. Briefings in Bioinformatics 2004;5(4): 355—64.

[25] International HapMap Consortium. The international hapmap project. Nature 2003;426(6968):789—96.
[26] Excoffier L, Slatkin M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. Molecular Biology and Evolution 1995;12(5): 921—7.
[27] Adkins RM. Comparison of the accuracy of methods of computational haplotype inference using a large empirical dataset. BMC Genetics 2004;5:22—8.
[28] Horan M, Millar DS, Hedderich J, Lewis G, Newsway V, Mo N, et al. Human growth hormone 1 (gh1) gene expression: complex haplotype-dependent influence of polymorphic variation in the proximal promoter and locus control region. Human Mutation 2003;21(4):408—23.
[29] Schwartz R, Halldorsson BV, Bafna V, Clark AG, Istrail S. Robustness of inference of haplotype block structure. Journal of Computational Biology 2003;10(1): 13—9.
[30] Indap A, Marth G, Struble C, Tonellato P, Olivier M. Analysis of concordance of different haplotype block partitioning algorithms. BMC Bioinformatics 2005;6(1):303—16.
[31] Malerba G, Schaeffer L, Xumerle L. Snps of the fads gene cluster are associated with polyunsaturated fatty acids in a cohort of patients with cardiovascular disease. Lipids 2008;5(4):355—64.
[32] Lewontin RC. The interaction of selection of linkage. Genetics 1964;49(1):49—67.
[33] Carlson C, Eberle M, Rieder M, Yi Q, Kruglyak L, Nickerson D. Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. American Journal of Human Genetics 2004;74(1): 106—20.
[34] Gohlke H, Illig T, Bahnweg M, Klopp N, Andre E, Altmuller J, et al. Association of the interleukin-1 receptor antagonist gene with asthma. American Journal Respiratory and Critical Care Medicine 2004;169(11):1217—23.