# Probabilistic Logic Learning from Haplotype Data

Niels Landwehr[1] and Taneli Mielikäinen[2]

[1] Machine Learning Lab, Institute for Computer Science, University of Freiburg
Georges-Koehler Allee, Building 079, 79110 Freiburg, Germany
landwehr@informatik.uni-freiburg.de
[2] Helsinki Institute for Information Technology, University of Helsinki, Finland
Taneli.Mielikainen@cs.helsinki.fi

**Abstract.** The analysis of *haplotype* data of human populations has received much attention recently. For instance, problems such as *Haplotype Reconstruction* are important intermediate steps in gene association studies, which seek to uncover the genetic basis of complex diseases. In this chapter, we explore the application of probabilistic logic learning techniques to haplotype data. More specifically, a new haplotype reconstrcution technique based on Logical Hidden Markov Models is presented and experimentally compared against other state-of-the-art haplotyping systems. Furthermore, we explore approaches for combining haplotype reconstructions from different sources, which can increase accuracy and robustness of reconstruction estimates. Finally, techniques for discovering the structure in haplotype data at the level of haplotypes and population are discussed.

## 1 Introduction

In this chapter, we will look at applications of probabilistic logic learning and related approaches in the area of genetic data analysis. More specifically, we are concerned with analyzing *haplotype* data—a concise representation of the individual genetic make-up of an organism, that is encoded in a set of genetic *markers*. The analysis of haplotype data has become a central theme in modern bioinformatics, and is considered to be a promising approach to many important problems in human biology and medicine. Application areas range from the quest to identify genetic roots of complex diseases to analyzing the evolution history of populations or developing "personalized" medicine based on the individual genetic disposition of the patient.

The rest of the chapter is organized as follows. After starting with a brief introduction to the basic concepts of genetics, such as the genome, chromosomes, and haplotypes, three different haplotype data analysis problems will be discussed. The first problem concerns *haplotype reconstruction*: the problem of resolving the hidden phase information in genotype data obtained from laboratory measurements. For this problem a new statistical method based on Logical Hidden Markov Models is introduced. The second, related, problem is that of *combining*

L. De Raedt et al. (Eds.): Probabilistic ILP 2007, LNAI 4911, pp. 263–286, 2008.
© Springer-Verlag Berlin Heidelberg 2008

*haplotypings*, that is, the question how different haplotype reconstructions obtained from different algorithmic methods can be combined and jointly analyzed. The third problem is concerned with discovering the *structure* in haplotype data, at the level of haplotypes and populations of individuals.

### 1.1 Genomes, Chromosomes and Haplotypes

The *genome* is organized as a set of *chromosomes* [TJHBD97]. A chromosome is a *DNA molecule* consisting of *nucleotides*, small molecules that connect to form the long chain-like DNA molecule. Basically, four different nucleotides occur (Adenine, Cytosine, Guanine, Thymine), and the genetic information is encoded in the sequence of "letters" A,C,G and T. Thus, for our purposes, a DNA molecule is a sequence over the alphabet $\{A, C, G, T\}$, and a genome is then a collection of sequences in $\{A, C, G, T\}^*$.

Most of the genome is invariant between different human individuals. However, the genetic variations that do exist play a crucial role in determining our genetic individuality, they can e.g. contribute to risk factors of complex diseases or influence how an individual patient responds to a certain drug treatment. The analysis of genetic variation in human populations has therefore become a focus of attention in human biology recently [The05]. Most studied differences in the genome are single-nucleotide variations at particular positions in the genome, which are called *single nucleotide polymorphisms* (SNPs). The positions are also called *markers* and the different possible values *alleles*. A *haplotype* is a sequence of SNP alleles along a region of a chromosome, and concisely represents the (variable) genetic information in that region.

The genetic variation in SNPs is mostly due to two causes: *mutation* and *recombination*. A mutation changes a single nucleotide in the chromosome. Mutations are relatively rare, they occur with a frequency of about $10^{-8}$. While SNPs are themselves results of ancient mutations, mutations are usually ignored in statistical haplotype models due to their rarity. Recombination introduces variability by breaking up the chromosomes of the two parents and reconnecting the resulting segments to form a new and different chromosome for the offspring. Because the probability of a recombination event between two markers is lower if they are near to each other, there is a statistical correlation (so-called *linkage disequilibrium*) between markers which decreases with increasing marker distance. Statistical approaches to haplotype modeling are based on exploiting such patterns of correlation.

In diploid organisms such as humans there are two *homologous* (i.e., almost identical) copies of each chromosome. Determining haplotype information for an individual therefore means measuring a set of markers along a chromosome for both copies of the chromosome. Current practical laboratory measurement techniques produce a *genotype*—for $m$ markers, a sequence of $m$ unordered pairs of alleles. The genotype reveals which two alleles are present at each marker, but not their respective chromosomal origin. Genotypes, as sequences of unordered pairs, are an example of the way data is *structured* in haplotype analysis, posing challenges to standard propositional data analysis techniques. Using

propositional techniques, a genotype could be represented as a sequence of unordered pairs, where each unordered pair is considered as a letter in the alphabet. However, such a representation would not take into account the intrinsic structure in each letter as an unordered pair. These limitations can be overcome using a relational representation of the data, as will be shown in the next section.

A similarly challenging task is the representation of a haplotype pair in propositional form, as a haplotype pair consists of two haplotype sequences and there is no natural order for the sequences in the pair. In some cases it might be known which of the haplotypes is inherited from the maternal/paternal genome, but this does not yield a natural ordering: based on the current knowledge of genetics, it does not matter from which parent a particular copy of a chromosome is inherited. Such representational issues will also be discussed in the forthcoming sections. Furthermore, additional relational information could be taken into account. Individuals can be related (e.g., by family relations), and relations between different regions of the marker maps are sometimes known. For example, certain genes might be known to be correlated. Such information is typically probabilistic.

Because of the outlined difficulties with representing haplotype data in propositional form, this domain is an interesting challenge for statistical relational modeling techniques.

*Notational Convention.* For our purposes, a haplotype $h$ is a sequence of alleles $h[i]$ in markers $i = 1, \ldots, m$. In most cases, only two alternative alleles occur at an SNP marker, so we can assume that $h \in \{0, 1\}^m$. A genotype $g$ is a sequence of unordered pairs $g[i] = \{h_g^1[i], h_g^2[i]\}$ of alleles in markers $i = 1, \ldots, m$. Hence, $g \in \{\{0, 0\}, \{1, 1\}, \{0, 1\}\}^m$. A marker with alleles $\{0, 0\}$ or $\{1, 1\}$ is *homozygous* whereas a marker with alleles $\{0, 1\}$ is *heterozygous*. The number of heterozygous markers is denoted by $m'$ and the number of individuals in the population by $n$.

## 2 Haplotype Reconstruction

This section describes and formalizes the haplotype reconstruction (or *haplotyping*) problem, and presents a new method for statistical haplotype reconstruction based on Logical Hidden Markov Models (LOHMMs, see Chapter 3). We will start by defining the problem setting and present a basic LOHMM model for this domain. Two extensions to the basic model will be presented, and finally the method is compared against several state-of-the-art haplotyping techniques on real-world population data.

### 2.1 The Haplotype Reconstruction Problem

In order to obtain haplotype data for a set of human individuals, their genotypes are measured in the laboratory, and afterwards the haplotypes must be determined from this genotype data. There are two alternative approaches for this reconstruction: One is to use *family trios*, i.e., genotype two parents and the

corresponding child. If trios are available, most of the ambiguity in the phase (the order of the alleles in the genotype data) can be resolved analytically, and haplotypes be inferred. If no trios can be obtained, population-based computational methods have to be used to estimate the haplotype pair for each genotype. These approaches exploit statistical correlations between different markers to estimate a distribution over haplotypes for the population sample in question, and use this estimate to infer the most likely haplotype pair for each genotype in the sample. Because trios are more difficult to recruit and more expensive to genotype, population-based approaches are often the only cost-effective method for large-scale studies. Consequently, the study of such techniques has received much attention recently [SWS05, HBE+04].

*Problem 1 (haplotype reconstruction).* Given a multiset $\mathcal{G}$ of genotypes, find for each $g \in \mathcal{G}$ the most likely haplotypes $h_g^1$ and $h_g^2$ which are a *consistent* reconstruction of $g$, i.e., $g[i] = \{h_g^1[i], h_g^2[i]\}$ for each $i = 1, \ldots, m$.

If $\mathcal{H}$ denotes a mapping $\mathcal{G} \to \{0,1\}^m \times \{0,1\}^m$, associating each genotype $g \in \mathcal{G}$ with a pair $\langle h_g^1, h_g^2 \rangle$ of haplotypes, the goal is to find the $\mathcal{H}$ that maximizes $\mathbb{P}(\mathcal{H} \mid \mathcal{G})$. It is usually assumed that the sample $\mathcal{G}$ is in Hardy-Weinberg equilibrium, i.e., that $\mathbb{P}(\langle h_g^1, h_g^2 \rangle) = \mathbb{P}(h_g^1)\,\mathbb{P}(h_g^2)$ for all $g \in \mathcal{G}$, and that genotypes are independently sampled from the same distribution. With such assumptions, the likelihood $\mathbb{P}(\mathcal{H} \mid \mathcal{G})$ of the reconstruction $\mathcal{H}$ given $\mathcal{G}$ is proportional to $\prod_{g \in \mathcal{G}} \mathbb{P}(h_g^1)\,\mathbb{P}(h_g^2)$ if the reconstruction is consistent for all $g \in \mathcal{G}$, and zero otherwise. In population-based haplotyping, a probabilistic model $\lambda$ for the distribution over haplotypes is estimated from the available genotype information $\mathcal{G}$. The distribution estimate $\mathbb{P}(h \mid \lambda)$ is then used to find the most likely reconstruction $\mathcal{H}$ for $\mathcal{G}$ under Hardy-Weinberg equilibrium.

### 2.2 A LOHMM Model for Haplotyping

Logical hidden Markov models (LOHMMs, see Chapter 3) upgrade traditional hidden Markov models to deal with sequences of structured symbols, rather than flat characters. The key idea underlying LOHMMs is to employ logical atoms as structured (output and state) symbols. More specifically, LOHMMs define *abstract* states such as $s(A, B)$ where $s$ is the state name and $A, B$ are logical variables. An abstract state represents a set of "ground" states, namely all variable-free logical specializations of the abstract state expression $s(A, B)$ (e.g., $s(1, 0)$). Abstract transitions such as $s(X, Y) \to s'(1, Y)$ describe how the model transitions between abstract states, and variable unification is used to share information between states, and between states and observations. Variants of the Expectation-Maximization and Viterbi algorithms used with standard HMMs can be derived for learning and inference in LOHMMs.

The basic motivation for using LOHMMs in haplotyping is that it is straightforward to encode genotypes (sequences of unordered pairs) as sequences of logical atoms. This can be done with a predicate $pair(X, Y)$, which can be grounded to $pair(0, 0)$ (homozygous 0), $pair(1, 1)$ (homozygous 1), and $pair(0, 1)$ (heterozygous). Using logical variables and unification, the two individual alleles in

the pair can be accessed. This allows to represent biological knowledge such as the assumption of Hardy-Weinberg equilibrium (the fact that a genotype is sampled by sampling two haplotypes independently and from the same distribution) in the LOHMM structure.

As underlying model for the distribution over haplotypes, we use a straightforward left-to-right Markov model $\lambda$ over the binary marker values at positions $t = 1, \ldots, m$:

$$\mathbb{P}(h) = \prod_{t=1}^{m} \mathbb{P}_t(h[t] \mid h[t-1], \lambda).$$

This is motivated by the observation that linkage disequilibrium is strongest for adjacent markers. Parameters of this model are of the form $\mathbb{P}_t(h[t] \mid h[t-1])$, the probability of sampling the new allele $h[t]$ at position $t$ after observing the allele $h[t-1]$ at position $t-1$. The Markov model on haplotypes can be extended to a LOHMM on genotypes as follows. The LOHMM is organized as a left-to-right model with layers $t = 1, \ldots, m$. At every layer $t$, one component of the model encodes the distribution $P(h[t+1] \mid h[t])$. This component is traversed twice for sampling the two new alleles $h^1[t+1], h^2[t+1]$ based on their respective histories $h^1[t], h^2[t]$. Afterwards, the unordered pair corresponding to the new allele pair is emitted.

Figure 1 shows a single layer (at marker $t$) of the LOHMM model. For sampling two new markers $h^1[t+1], h^2[t+1]$ at position $t+1$ based on the markers $h^1[t], h^2[t]$ at position $t$, we start at state $m_t(X, Y)$ with $h^1[t], h^2[t]$ bound to $X$ and $Y$. The model then transitions to the state $s_t(X, Y, x)$ to sample the first new marker $h^1[t+1]$. The multiple transitions from state $s_t$ to state $s'_t$ encode the distribution $P(h[t+1] \mid h[t])$. In $s'_t(A', B, x)$, the new marker $h^1[t+1]$ has been sampled and is bound to $A'$. Afterwards, the same path is traversed again to sample the second marker, with arguments in state $s_t$ swapped. This effectively samples the new marker $h^2[t+1]$ based on $h^2[t]$ *independently* and *from the same distribution*. Finally, the unordered pair corresponding to the two new markers is emitted in the transition from $s'_t$ to $m_{t+1}$. This is can be easily accomplished using the logical generality ordering on abstract states in LOHMMs: if the more specific abstract states for homozygous markers match the ground state a homozygous pair is emitted, otherwise, an (unordered) heterozygous pair. Note that this model only has 2 free parameters per layer, in contrast to a naive first-order HMM model on the the joint state of the two haplotypes, which would have 12 free parameters per layer.

This kind of model can be directly trained from genotype data using the EM algorithm for LOHMMs, and the most likely haplotype pair for a genotype can be read off the most likely state sequence for that observation returned by the Viterbi algorithm (see [KDR06]). However, initial experiments using the XANTHOS engine for LOHMMs showed that the computational overhead due to the general-purpose framework used in LOHMMs reduced the computational efficiency of the model. Fortunately, it is possible to compile the presented LOHMM model into an equivalent HMM model with parameter tying constraints. While

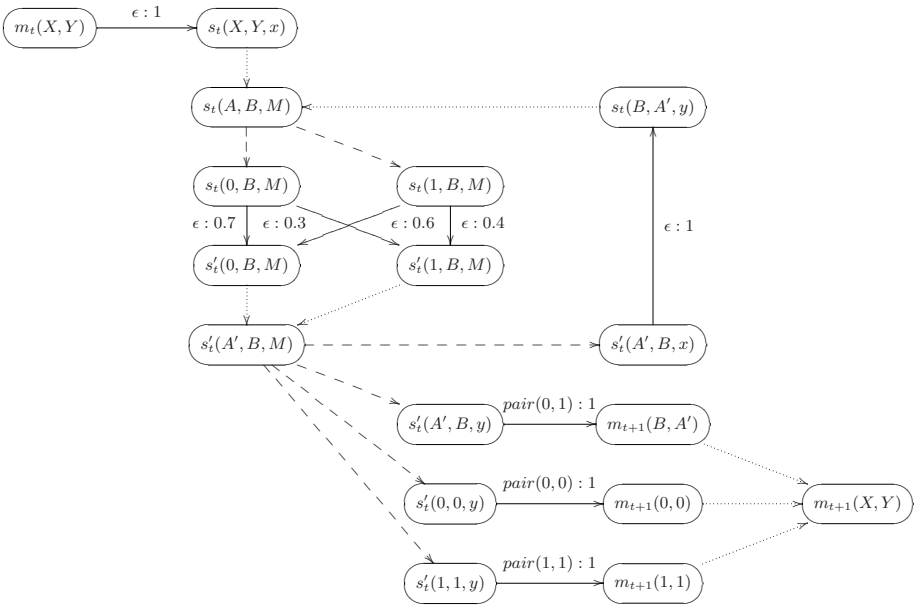


**Fig. 1.** LOHMM for haplotype reconstruction. One layer at marker position $t$ is shown. The standard syntax for visualizing LOHMMs is used: solid arrows represent abstract transitions, dashed arrows the "more general than" relation, and dotted arrows "must follow" links. For a more detailed description, see Chapter 3.

the details of this transformation are beyond the scope of this article, it generally follows the grounding mechanism for LOHMMs, as described in [KDR06].

### 2.3 Higher Order Models and Sparse Distributions

The main limitation of the model presented so far is that it only takes into account dependencies between adjacent markers. Expressivity can be increased by using a Markov model of order $k > 1$ for the underlying haplotype distribution [EGT04]:

$$\mathbb{P}(h) = \prod_{t=1}^{m} \mathbb{P}_t(h[t] \mid h[t-k, t-1], \lambda),$$

where $h[j, i]$ is a shorthand for $h[\max\{1, j\}] \dots h[i]$. Unfortunately, the number of parameters in such a model increases exponentially with the history length $k$. However, observations on real-world data (e.g., [DRS+01]) show that only few conserved haplotype fragments from the set of $2^k$ possible binary strings of length $k$ actually occur in a particular population. This can be exploited by modeling sparse distributions, where fragment probabilities which are estimated

**Algorithm 1.** The level-wise SpaMM learning algorithm

```
Initialize k := 1
λ_1 := INITIAL-MODEL()
λ_1 := EM-TRAINING(λ_1)
repeat
   k := k + 1
   λ_k := EXTEND-AND-REGULARIZE(λ_{k−1})
   λ_k := EM-TRAINING(λ_k)
until k = k_max
```

to be very low are set to zero. More precisely, let $p = \mathbb{P}_t(h[t] \mid h[t-k, t-1])$ and define for some small $\epsilon > 0$ a regularized distribution

$$\hat{\mathbb{P}}_t(h[t] \mid h[t-k, t-1]) = \begin{cases} 0 & \text{if } p \leq \epsilon; \\ 1 & \text{if } p > 1 - \epsilon; \\ p & \text{otherwise.} \end{cases}$$

If the underlying distribution is sufficiently sparse, $\hat{\mathbb{P}}$ can be represented using a relatively small number of parameters. The corresponding sparse hidden Markov model structure (in which transitions with probability 0 are removed) will reflect the pattern of conserved haplotype fragments present in the population. How such a sparse model structure can be learned without ever constructing the prohibitively complex distribution $\mathbb{P}$ will be discussed in the next section.

### 2.4 SpaMM: A Level-Wise Learning Algorithm

To construct the sparse order-$k$ hidden Markov model, we propose a learning algorithm—called **SpaMM** for **Spa**rse **M**arkov **M**odeling—that iteratively refines hidden Markov models of increasing order (Algorithm 1). More specifically, the idea of SpaMM is to identify conserved fragments using a level-wise search, i.e., by extending short fragments (in low-order models) to longer ones (in high-order models), and is inspired by the well-known Apriori data mining algorithm [AMS+96]. The algorithm starts with a first-order Markov model $\lambda_1$ on haplotypes where initial transition probabilities are set to $\dot{\mathbb{P}}_t(h[t] \mid h[t-1], \lambda_1) = 0.5$ for all $t \in \{1, \ldots, m\}, h[t], h[t-1] \in \{0, 1\}$. For this model, a corresponding LOHMM on genotypes can be constructed as outlined in Section 2.2, which can be compiled into a standard HMM with parameter tying constraints and trained on the available genotype data using EM.

The function EXTEND-AND-REGULARIZE($\lambda_{k-1}$) takes as input a model of order $k-1$ and returns a model $\lambda_k$ of order $k$. In $\lambda_k$, initial transition probabilities are set to

$$\dot{\mathbb{P}}_t(h[t] \mid h[t-k, t-1], \lambda_{k+1}) = \begin{cases} 0 & \text{if } \mathbb{P}_t(h[t] \mid h[t-k+1, t-1], \lambda_k) \leq \epsilon; \\ 1 & \text{if } \mathbb{P}_t(h[t] \mid h[t-k+1, t-1], \lambda_k) > 1 - \epsilon; \\ 0.5 & \text{otherwise,} \end{cases}$$

i.e., transitions are removed if the probability of the transition conditioned on a shorter history is smaller than $\epsilon$. This procedure of iteratively training, extending

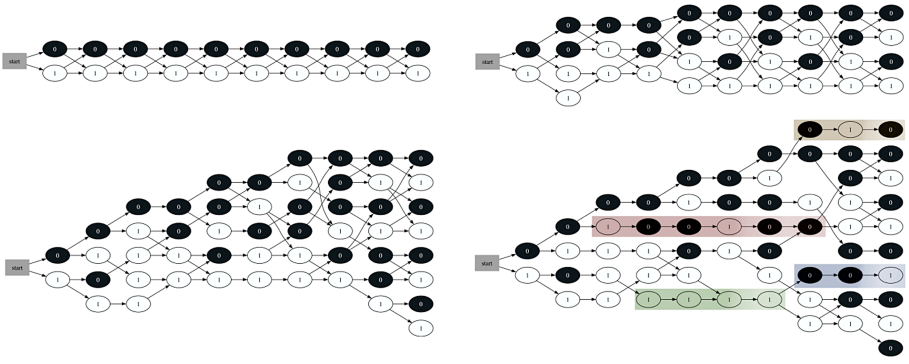

**Fig. 2. Visualization of the SpaMM Structure Learning Algorithm.** Sparse models $\lambda_1, \ldots, \lambda_4$ of increasing order learned on the Daly dataset are shown. Black/white nodes encode more frequent/less frequent allele in population. Conserved fragments identified in $\lambda_4$ are highlighted.

and regularizing Markov models of increasing order is repeated up to a maximum order $k_{max}$.

Figure 2 visualizes the underlying distribution over haplotypes learned in the first 4 iterations of the SpaMM algorithm on a real-world dataset. The set of paths through the lattice corresponds to the set of haplotypes which have non-zero probability according to the model. Note how some of the possible haplotypes are pruned and conserved fragments are isolated. Accordingly, the number of states and transitions in the final LOHMM/HMM model is significantly smaller than for a full model of that order.

### 2.5 Experimental Evaluation

The proposed method was implemented in the SpaMM haplotyping system[1]. We compared its accuracy and computational performance to several other state-of-the art haplotype reconstruction systems: PHASE version 2.1.1 [SS05], fastPHASE version 1.1 [SS06], GERBIL as included in GEVALT version 1.0 [KS05], HIT [RKMU05] and HaploRec (variable order Markov model) version 2.0 [EGT06]. All methods were run using their default parameters. The fastPHASE system, which also employs EM for learning a probabilistic model, uses a strategy of averaging results over several random restarts of EM from different initial parameter values. This reduces the variance component of the reconstruction error and alleviates the problem of local minima in EM search. As this is a general technique applicable also to our method, we list results for fastPHASE with averaging (fastPHASE) and without averaging (fastPHASE-NA).

The methods were compared using publicly available real-world datasets, and larger datasets simulated with the Hudson coalescence simulator [Hud02]. As

[1] The implementation is available at `http://www.informatik.uni-freiburg.de/~landwehr/haplotyping.html`

**Table 1. Reconstruction Accuracy on Yoruba and Daly Data.** Normalized switch error is shown for the Daly dataset, and average normalized switch error over the 100 datasets in the Yoruba-20, Yoruba-100 and Yoruba-500 dataset collections.

| Method | Yoruba-20 | Yoruba-100 | Yoruba-500 | **Daly** |
|---|---|---|---|---|
| PHASE | **0.027** | **0.025** | *n.a.* | 0.038 |
| fastPHASE | 0.033 | 0.031 | **0.034** | **0.027** |
| SpaMM | 0.034 | 0.037 | 0.040 | 0.033 |
| HaploRec | 0.036 | 0.038 | 0.046 | 0.034 |
| fastPHASE-NA | 0.041 | 0.060 | 0.069 | 0.045 |
| HIT | 0.042 | 0.050 | 0.055 | 0.031 |
| GERBIL | 0.044 | 0.051 | *n.a* | 0.034 |

real-world data, we used a collection of datasets from the Yoruba population in Ibadan, Nigeria [The05], and the well-known dataset of Daly et al [DRS+01], which contains data from a European-derived population. For these datasets, family trios are available, and thus true haplotypes can be inferred analytically.

For the Yoruba population, we sampled 100 sets of 500 markers each from distinct regions on chromosome 1 (**Yoruba-500**), and from these smaller datasets by taking only the first 20 (**Yoruba-20**) or 100 (**Yoruba-100**) markers for every individual. There are 60 individuals in the dataset after preprocessing, with an average fraction of missing values of 3.6%. For the **Daly** dataset, there is information on 103 markers and 174 individuals available after data preprocessing, and the average fraction of missing values is 8%. The number of genotyped individuals in these real-world datasets is rather small. For most disease association studies, sample sizes of at least several hundred individuals are needed [WBCT05], and we are ultimately interested in haplotyping such larger datasets. Unfortunately, we are not aware of any publicly available real-world datasets of this size, so we have to resort to simulated data. We used the well-known Hudson coalescence simulator [Hud02] to generate 50 artificial datasets, each containing 800 individuals (**Hudson** datasets). The simulator uses the standard Wright-Fisher neutral model of genetic variation with recombination. To come as close to the characteristics of real-world data as possible, some alleles were masked (marked as missing) after simulation.

The accuracy of the reconstructed haplotypes produced by the different methods was measured by normalized switch error. The switch error of a reconstruction is the minimum number of recombinations needed to transform the reconstructed haplotype pair into the true haplotype pair. (See Section 3 for more details.) To normalize, switch errors are summed over all individuals in the dataset and divided by the total number of switch errors that could have been made. For more details on the methodology of the experimental study, confer [LME+07].

Table 1 shows the normalized switch error for all methods on the real-world datasets Yoruba and Daly. For the dataset collections Yoruba-20, Yoruba-100 and Yoruba-500 errors are averaged over the 100 datasets. PHASE and Gerbil

**Table 2. Average Error for Reconstructing Masked Genotypes on Yoruba-100.** From 10% to 40% of all genotypes were masked randomly. Results are averaged over 100 datasets.

| Method | 10% | 20% | 30% | 40% |
|---|---|---|---|---|
| fastPHASE | **0.045** | **0.052** | **0.062** | **0.075** |
| SpaMM | 0.058 | 0.066 | 0.078 | 0.096 |
| fastPHASE-NA | 0.067 | 0.075 | 0.089 | 0.126 |
| HIT | 0.070 | 0.079 | 0.087 | 0.098 |
| GERBIL | 0.073 | 0.091 | 0.110 | 0.136 |

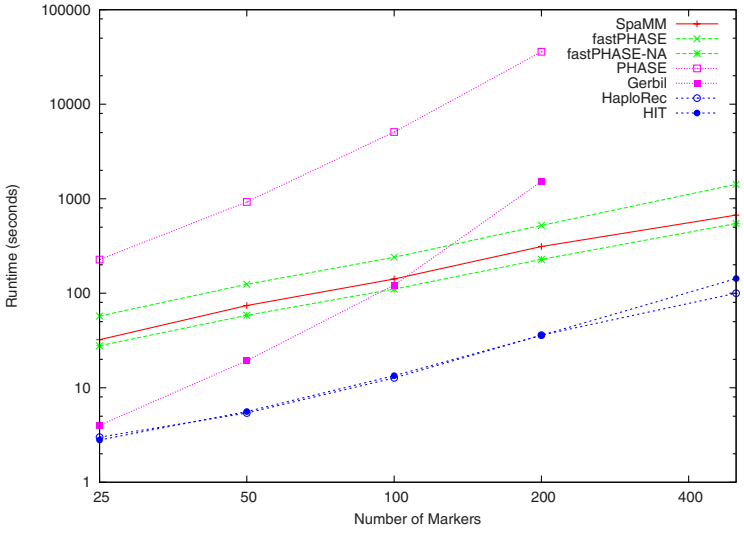


**Fig. 3. Runtime as a Function of the Number of Markers.** Average runtime per dataset on Yoruba datasets for marker maps of length 25 to 500 for SpaMM, fastPHASE, fastPHASE-NA, PHASE, Gerbil, HaploRec, and HIT are shown (logarithmic scale). Results are averaged over 10 out of the 100 datasets in the Yoruba collection.

did not complete on Yoruba-500 in two weeks[2]. Overall, the PHASE system achieves highest reconstruction accuracies. After PHASE, fastPHASE with averaging is most accurate, then SpaMM, and then HaploRec. Figure 3 shows the average runtime of the methods for marker maps of different lengths. The most accurate method PHASE is also clearly the slowest. fastPHASE and SpaMM are substantially faster, and HaploRec and HIT very fast. Gerbil is fast for small marker maps but slow for larger ones. For fastPHASE, fastPHASE-NA, HaploRec, SpaMM and HIT, computational costs scale linearly with the length of

[2] All experiments were run on standard PC hardware with a 3.2GHz processor and 2GB of main memory.

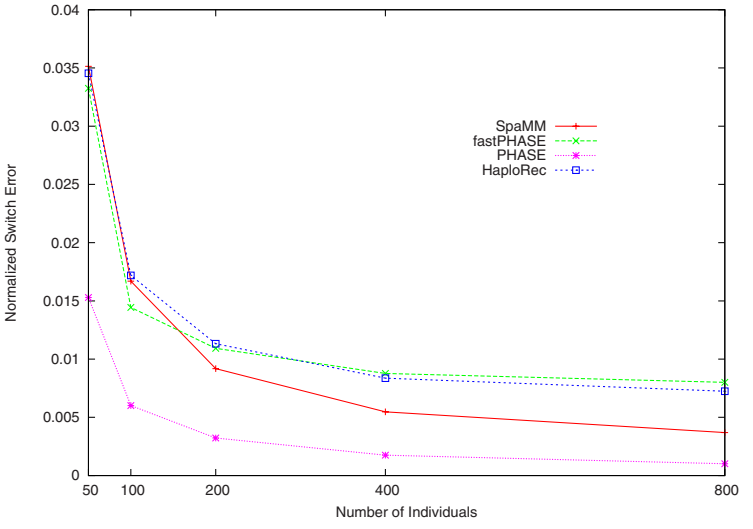


**Fig. 4. Reconstruction Accuracy as a Function of the Number of Samples Available.** Average normalized switch error on the Hudson datasets as a function of the number of individuals for SpaMM, fastPHASE, PHASE and HaploRec is shown. Results are averaged over 50 datasets.

the marker map, while the increase is superlinear for PHASE and Gerbil, so computational costs quickly become prohibitive for longer maps.

Performance of the systems on larger datasets with up to 800 individuals was evaluated on the 50 simulated Hudson datasets. As for the real-world data, the most accurate methods were PHASE, fastPHASE, SpaMM and HaploRec. Figure 4 shows the normalized switch error of these four methods as a function of the number of individuals (results of Gerbil, fastPHASE-NA, and HIT were significantly worse and are not shown). PHASE was the most accurate method also in this setting, but the relative accuracy of the other three systems depended on the number of individuals in the datasets. While for relatively small numbers of individuals (50–100) fastPHASE outperforms SpaMM and HaploRec, this is reversed for 200 or more individuals.

A problem closely related to haplotype reconstruction is that of genotype imputation. Here, the task is to infer the most likely genotype values (unordered allele pairs) at marker positions where genotype information is missing, based on the observed genotype information. With the exception of HaploRec, all haplotyping systems included in this study can also impute missing genotypes. To test imputation accuracy, between 10% and 40% of all markers were masked randomly, and then the marker values inferred by the systems were compared to the known true marker values. Table 2 shows the accuracy of inferred genotypes for different fractions of masked data on the Yoruba-100 datasets and Table 3 on the simulated Hudson datasets with 400 individuals per dataset. PHASE was

**Table 3. Average Error for Reconstructing Masked Genotypes on Hudson.** From 10% to 40% of all genotypes were masked randomly. Results are averaged over 50 datasets.

| Method | 10% | 20% | 30% | 40% |
|---|---|---|---|---|
| fastPHASE | 0.035 | 0.041 | 0.051 | 0.063 |
| SpaMM | **0.017** | **0.023** | **0.034** | **0.052** |
| fastPHASE-NA | 0.056 | 0.062 | 0.074 | 0.087 |
| HIT | 0.081 | 0.093 | 0.108 | 0.127 |
| GERBIL | 0.102 | 0.122 | 0.148 | 0.169 |

too slow to run in this task as its runtime increases significantly in the presence of many missing markers. Evidence from the literature [SS06] suggests that for this task, fastPHASE outperforms PHASE and is indeed the best method available. In our experiments, on Yoruba-100 fastPHASE is most accurate, SpaMM is slightly less accurate than fastPHASE, but more accurate than any other method (including fastPHASE-NA). On the larger Hudson datasets, SpaMM is significantly more accurate than any other method.

To summarize, our experimental results confirm PHASE as the most accurate but also computationally most expensive haplotype reconstruction system [SS06,SS05]. If more computational efficiency is required, fastPHASE yields the most accurate reconstructions on small datasets, and SpaMM is preferable for larger datasets. SpaMM also infers missing genotype values with high accuracy. For small datasets, it is second only to fastPHASE; for large datasets, it is substantially more accurate than any other method in our experiments.

## 3 Comparing Haplotypings

For haplotype pairs, as structured objects, there is no obvious way of measuring similarity—if two pairs are not identical, their distance could be measured in several ways. At the same time, comparing haplotypings is important for many problems in haplotype analysis, and therefore a *distance* or *similarity* measure on haplotype pairs is needed. The ability to compare haplotypings is useful, for example, for evaluating the quality of haplotype reconstructions, if (at least for part of the data) the correct haplotypings are known. An alternative approach to evaluation would be to have an accurate generative model of haplotype data for the population in question, which could assign probability scores to haplotype reconstructions. However, such a model seems even harder to obtain than known correct haplotype reconstructions (which can be derived from family trios).

Moreover, a distance measure between haplotypes allows to compute *consensus* haplotype reconstructions, which average between different, conflicting reconstructions—for example, by minimizing the sum of distances. This opens up possibilities for the application of ensemble methods in haplotype analysis, which can increase accuracy and robustness of solutions. Finally, comparison

operators can be used to study the structure of populations (Section 4.1) or structure of haplotypes (Section 4.2).

Although we could simply represent the haplotype data in a relational form and use standard relational distance measures, distance measures customized to this particular problem will take our knowledge about the domain better into account, and thus yield better results. In the rest of this section we will discuss different approaches to define distances between haplotype pairs and analyze their properties. Afterwards, we discuss algorithms to compute consensus haplotypes based on these distances, and present some computational complexity results.

### 3.1 Distance Computations

The genetic distance between two haplotype pairs is a complex function, which depends on the information the chromosomes of the two individuals contain (and, in principle, even other chemical properties of the DNA sequences). However, modeling distance functions at this level is rather tedious. Instead, simpler distance functions aiming to capture some aspects of the relevant properties of the genetic similarity have to be used.

In this section we consider distance functions based on markers, i.e., distances between haplotype pairs. These can be grouped into two categories: distances induced by distances between individual haplotypes, and distance functions that work with the pair directly. Pair-wise Hamming distance is the most well-known example for the first category, and switch distance for the second. We will also give a unified view to both of the distance functions by proposing a $k$-Hamming distance which interpolates between pair-wise Hamming distance and switch distance.

**Hamming distance and other distances induced by distances on sequences.** The most common distance measure between sequences $s, t \in \Sigma^m$ is the Hamming distance that counts the number of disagreements between $s$ and $t$, i.e.,

$$d_H(s,t) = |\{i \in \{1,\ldots,m\} : s[i] \neq t[i]\}| \,. \tag{1}$$

The Hamming distance is not directly applicable for comparing the genetic information of two individuals, as this information consists of a pair of haplotypes. To generalize the Hamming distance to pairs of haplotypes, let us consider haplotype pairs $\{h_1^1, h_1^2\}$ and $\{h_2^1, h_2^2\}$. The distance between the pairs should be zero if the sets $\{h_1^1, h_1^2\}$ and $\{h_2^1, h_2^2\}$ are the same. Hence, we should try to pair the haplotypes both ways and take the one with the smaller distance, i.e., $d_H(\{h_1^1, h_1^2\}, \{h_2^1, h_2^2\}) = \min\left\{d_H(h_1^1, h_2^1) + d_H(h_1^2, h_2^2), d_H(h_1^1, h_2^2) + d_H(h_1^2, h_2^1)\right\}$.

Note that a similar construction can be used to map any distance function between haplotype sequences to a distance function between pairs of haplotypings. Furthermore, if the distance function between the sequences satisfies the triangle inequality, so does the corresponding distance function for haplotype reconstructions.

**Proposition 1.** *Let $d\colon \Sigma^m \times \Sigma^m \to \mathbb{R}_{\geq 0}$ be a distance function between sequences of length $m$ and*

$$d(\{h_1^1, h_1^2\}, \{h_2^1, h_2^2\}) = \min\{d(h_1^1, h_2^1) + d(h_1^2, h_2^2), d(h_1^1, h_2^2) + d(h_1^2, h_2^1)\}$$

*for all $h_1^1, h_1^2, h_2^1, h_2^2 \in \Sigma^m$. If $d$ satisfies the triangle inequality for comparing sequences, i.e.,*

$$d(s,t) \leq d(s,u) + d(t,u)$$

*for all $s, t, u \in \Sigma^m$, then $d$ satisfies the triangle inequality for comparing unordered pairs of sequences, i.e.,*

$$d(h_1, h_2) \leq d(h_1, h_3) + d(h_2, h_3)$$

*for all $h_1^1, h_1^2, h_2^1, h_2^2, h_3^1, h_3^2 \in \Sigma^m$.*

*Proof.* Choose arbitrary sequences $h_1^1, h_1^2, h_2^1, h_2^2, h_3^1, h_3^2 \in \Sigma^m$. We show that the claim holds for them and hence for all sequences of length $m$ over the alphabet $\Sigma$. Assume, without loss of generality, that $d(\{h_1^1, h_1^2\}, \{h_2^1, h_2^2\}) = d(h_1^1, h_2^1) + d(h_1^2, h_2^2)$ and $d(\{h_1^1, h_1^2\}, \{h_3^1, h_3^2\}) = d(h_1^1, h_3^1) + d(h_1^2, h_3^2)$. For $d(\{h_2^1, h_2^2\}, \{h_3^1, h_3^2\})$ there are two cases as it is the minimum of $d(h_2^1, h_3^1) + d(h_2^2, h_3^2)$ and $d(h_2^2, h_3^1) + d(h_2^1, h_3^2)$.

If $d(\{h_2^1, h_2^2\}, \{h_3^1, h_3^2\}) = d(h_2^1, h_3^1) + d(h_2^2, h_3^2)$, then

$$\begin{aligned} &d(\{h_1^1, h_1^2\}, \{h_3^1, h_3^2\}) + d(\{h_2^1, h_2^2\}, \{h_3^1, h_3^2\}) = \\ &d(h_1^1, h_3^1) + d(h_1^2, h_3^2) + d(h_2^1, h_3^1) + d(h_2^2, h_3^2) = \\ &\big[d(h_1^1, h_3^1) + d(h_2^1, h_3^1)\big] + \big[d(h_1^2, h_3^2) + d(h_2^2, h_3^2)\big] \geq d(h_1^1, h_2^1) + d(h_1^2, h_2^2). \end{aligned}$$

If $d(\{h_2^1, h_2^2\}, \{h_3^1, h_3^2\}) = d(h_2^2, h_3^1) + d(h_2^1, h_3^2)$, then

$$\begin{aligned} &d(\{h_1^1, h_1^2\}, \{h_3^1, h_3^2\}) + d(\{h_2^1, h_2^2\}, \{h_3^1, h_3^2\}) = \\ &d(h_1^1, h_3^1) + d(h_1^2, h_3^2) + d(h_2^2, h_3^1) + d(h_2^1, h_3^2) = \\ &\big[d(h_1^1, h_3^1) + d(h_2^2, h_3^1)\big] + \big[d(h_1^2, h_3^2) + d(h_2^1, h_3^2)\big] \geq \\ &\qquad d(h_1^1, h_2^2) + d(h_1^2, h_2^1) \geq d(h_1^1, h_2^1) + d(h_1^2, h_2^2). \end{aligned}$$

Thus, the claim holds. □

The approach of defining distance functions between haplotype pairs based on distance functions between haplotypes has some limitations, regardless of the distance function used. This is because much of the variance in haplotypes originates from genetic *cross-over*, which breaks up the chromosomes of the parents and reconnects the resulting segments to form a new chromosome for the offspring. A pair $\{\hat{h}^1, \hat{h}^2\}$ of haplotypes which is the result of a cross-over between two haplotypes $h^1, h^2$ should be considered similar to the original pair $\{h^1, h^2\}$,

even though the resulting sequences can be radically different. This kind of similarity cannot be captured by distance functions on individual haplotypes.

**Switch distance.** An alternative distance measure for haplotype pairs is to compute the number of *switches* that are needed to transform a haplotype pair to another haplotype pair that corresponds to the same genotype. A switch between markers $i$ and $i+1$ for a haplotype pair $\{h^1, h^2\}$ transforms the pair $\{h^1, h^2\} = \{h^1[1,i]h^1[i+1,m], h^2[1,i]h^2[i+1,m]\}$ into the pair $\{h^1[1,i]h^2[i+1,m], h^2[1,i]h^1[i+1,m]\}$. It is easy to see that for any pair of haplotype reconstructions corresponding to the same genotype, there is a sequence of switches transforming one into the other. Thus, this *switch distance* is well defined for the cases we are interested in.

The switch distance, by definition, assigns high similarity to haplotype pairs if one pair can be transformed into the other by a small number of recombination events. It also has the advantage over the Hamming distance that the order of the haplotypes in the haplotype pair does not matter in the distance computation: the haplotype pair can be encoded uniquely as a bit sequence consisting of just the switches between the consecutive heterozygous markers, i.e., as a *switch sequence*:

**Definition 1 (Switch sequence).** *Let $h^1, h^2 \in \{0,1\}^m$ and let $i_1 < \ldots < i_{m'}$ be the heterozygous markers in $\{h^1, h^2\}$. The switch sequence of a haplotype pair $\{h^1, h^2\}$ is a sequence $s(h^1, h^2) = s(h^2, h^1) = s \in \{0,1\}^{m'-1}$ such that*

$$s[j] = \begin{cases} 0 & \text{if } h^1[i_j] = h^1[i_{j+1}] \text{ and } h^2[i_j] = h^2[i_{j+1}] \\ 1 & \text{if } h^1[i_j] \neq h^1[i_{j+1}] \text{ and } h^2[i_j] \neq h^2[i_{j+1}] \end{cases} \tag{2}$$

The switch distance between haplotype reconstructions can be defined in terms of the Hamming distance between switch sequences as follows.

**Definition 2 (Switch distance).** *Let $\{h_1^1, h_1^2\}$ and $\{h_2^1, h_2^2\}$ be haplotype pairs corresponding to the same genotype. The switch distance between the pairs is*

$$d_s(h_1, h_2) = d_s(\{h_1^1, h_1^2\}, \{h_2^1, h_2^2\}) = d_H(s(h_1^1, h_1^2), s(h_2^1, h_2^2))$$

As switch distance is the Hamming distance between the switch sequences, the following proposition is immediate:

**Proposition 2.** *The switch distance satisfies the triangle inequality.*

***k*-Hamming distance.** Switch distance considers only a very small neighborhood of each marker, namely only the previous and the next heterozygous marker in the haplotype. On the other extreme, the Hamming distance uses the complete neighborhood (via the min operation), i.e., the whole haplotypes for each marker. The intermediate cases are covered by the following $k$-Hamming distance in which all windows of a chosen length $k \in \{2, \ldots, m\}$ are considered. The intuition behind the definition is that each window of length $k$ is a potential location for a gene, and we want to measure how close the haplotype reconstruction $\{h^1, h^2\}$ gets to the true haplotype $\{h_2^1, h_2^2\}$ in predicting each of these potential genes.

**Definition 3 ($k$-Hamming distance).** *Let $\{h_1^1, h_1^2\}$ and $\{h_2^1, h_2^2\}$ be pairs of haplotype sequences corresponding to the same genotype with $m'$ heterozygous markers in positions $i_1, \ldots, i_m$. The $k$-Hamming distance $d_{k-H}$ between $\{h_1^1, h_1^2\}$ and $\{h_2^1, h_2^2\}$ is defined by*

$$d_{k-H}(h_1, h_2) = \sum_{j=1}^{m'-k+1} d_H(h_1[i_j, \ldots, i_{j+k-1}], h_2[i_j, \ldots, i_{j+k-1}])$$

*unless $m' < k$, in which case $d_{k-H}(h_1, h_2) = d_H(h_1, h_2)$.*

It is easy to see that $d_{2-H} = 2d_S$, and that for haplotyping pairs with $m'$ heterozygous markers, we have $d_{m'-H} = d_{m-H} = d_H$. Thus, the switch distance and the Hamming distance are the two extreme cases between which $d_{k-H}$ interpolates for $k = 2, \ldots, m' - 1$.

### 3.2 Consensus Haplotypings

Given a distance function $d$ on haplotype pairs, the problem of finding the *consensus haplotype pair* for a given set of haplotype pairs can be stated as follows:

*Problem 2 (Consensus Haplotype).* Given haplotype reconstructions $\{h_1^1, h_1^2\}$, $\ldots, \{h_l^1, h_l^2\} \subseteq \{0,1\}^m$, and a distance function $d : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}_{\geq 0}$, find:

$$\{h^1, h^2\} = \operatorname*{argmin}_{h^1, h^2 \in \{0,1\}^m} \sum_{i=1}^{l} d(\{h_i^1, h_i^2\}, \{h^1, h^2\}).$$

Consensus haplotypings are useful for many purposes. They can be used in ensemble methods to combine haplotype reconstructions from different sources in order to decrease reconstruction errors. They are also applicable when a representative haplotyping is needed, for example for a cluster of haplotypes which has been identified in a haplotype collection.

The complexity of finding the consensus haplotyping depends on the distance function $d$ used. As we will show next, for $d = d_S$ a simple voting scheme gives the solution. The rest of the distances considered in Section 3.1 are more challenging. If $d = d_{k-H}$ and $k$ is small, the solution can be found by dynamic programming. For $d = d_{k-H}$ with large $k$ and $d = d_H$, we are aware of no efficient general solutions. However, we will outline methods that can solve most of the problem instances that one may encounter in practice. For more details, confer [KLLM07].

**Switch distance: $d = d_S$.** For the switch distance, the consensus haplotyping can be found by the following voting scheme:

(1) Transform the haplotype reconstructions $\{h_i^1, h_i^2\} \subseteq \{0,1\}^m$, $i = 1, \ldots, l$ into switch sequences $s_1, \ldots, s_l \in \{0,1\}^{m'-1}$.
(2) Return the haplotype pair $\{h^1, h^2\}$ that shares the homozygous markers with the reconstructions $\{h_i^1, h_i^2\}$ and whose switch sequence $s \in \{0,1\}^{m'-1}$ is defined by $s[j] = \operatorname{argmax}_{b \in \{0,1\}} |\{j \in \{1, \ldots, m'-1\} : s_i[j] = b\}|$.

The time complexity of this method is $O(lm)$.

**$k$-Hamming distance: $d = d_{k-H}$.** The optimal consensus haplotyping is

$$h_* = \{h_*^1, h_*^2\} = \operatorname*{argmin}_{\{h^1,h^2\}\subseteq\{0,1\}^m} \sum_{i=1}^{l} d_{k-H}(h_i, h).$$

The number of potentially optimal solutions is $2^{m'}$, but the solution can be constructed incrementally based on the following observation:

$$\begin{aligned} h_* &= \operatorname*{argmin}_{\{h^1,h^2\}\subseteq\{0,1\}^m} \sum_{i=1}^{l} d_{k-H}(h_i, h) \\ &= \operatorname*{argmin}_{\{h^1,h^2\}\subseteq\{0,1\}^m} \sum_{i=1}^{l} \sum_{j=1}^{m'-k+1} d_H(h_i[i_j, \ldots, i_{j+k-1}], h[i_j, \ldots, i_{j+k-1}]) \end{aligned}$$

Hence, the cost of any solution is a sum of terms

$$D_j(\{x, \bar{x}\}) = \sum_{i=1}^{l} d_H(h_i[i_j, \ldots, i_{j+k-1}], \{x, \bar{x}\}), \quad j = 1, \ldots, m'-k+1, x \in \{0,1\}^k,$$

where $\bar{x}$ denotes the complement of $x$. There are $(m' - k + 1)2^{k-1}$ such terms. Furthermore, the cost of the optimal solution can be computed by dynamic programming using the recurrence relation

$$T_j(\{x, \bar{x}\} = \begin{cases} 0 & \text{if } j = 0 \\ D_j(\{x, \bar{x}) + \min_{b\in\{0,1\}} T_{j-1}(\{bx, \overline{bx}\}) & \text{if } j > 0 \end{cases}$$

Namely, the cost of the optimal solution is $\min_{x\in\{0,1\}^k} T_{m'}(\{x, \bar{x}\})$ and the optimal solution itself can be reconstructed by backtracking the path that leads to this position. The total time complexity for finding the optimal solution using dynamic programming is $\mathcal{O}(lm + 2^k kl(m' - k))$: the heterozygous markers can be detected and the data can be projected onto them in time $\mathcal{O}(lm)$, and the optimal haplotype reconstruction for the projected data can be computed in time $\mathcal{O}(2^k kl(m' - k))$. So the problem is fixed-parameter tractable[3] in $k$.

**Hamming distance: $d = d_H$.** An ordering $(h^1, h^2)$ of an optimal consensus haplotyping $\{h^1, h^2\}$ with Hamming distance determines an ordering of the unordered input haplotype pairs $\{h_1^1, h_1^2\}, \ldots, \{h_l^1, h_l^2\}$. This ordering can be represented by a binary vector $o = (o_1, \ldots, o_l) \in \{0,1\}^l$ that states for each $i = 1, \ldots, l$ that the ordering of $\{h_i^1, h_i^2\}$ is $(h_i^{1+o_i}, h_i^{2-o_i})$. Thus, $o_i = \operatorname{argmin}_{b\in\{0,1\}} d_H(h^1, h_i^{1+b})$, where ties are broken arbitrarily.

[3] A problem is called fixed-parameter tractable in a parameter $k$, if the running time of the algorithm is $f(k)\,\mathcal{O}(n^c)$ where $k$ is some parameter of the input and $c$ is a constant (and hence not depending on $k$.) For a good introduction to fixed-parameter tractability and parameterized complexity, see [FG06].

**Table 4.** The total switch error between true haplotypes and the haplotype reconstructions over all individuals for the baseline methods. For Yoruba and HaploDB, the reported numbers are the averages over the 100 datasets.

| Method | Daly | Yoruba | HaploDB |
|---|---|---|---|
| PHASE | 145 | 37.61 | 108.36 |
| fastPHASE | 105 | 45.87 | 110.45 |
| SpaMM | 127 | 54.69 | 120.29 |
| HaploRec | 131 | 56.62 | 130.28 |
| HIT | 121 | 73.23 | 123.95 |
| Gerbil | 132 | 75.05 | 134.22 |
| Ensemble | 104 | 39.86 | 103.06 |
| Ensemble w/o PHASE | 107 | 43.18 | 105.68 |

If the ordering $o$ is known and $l$ is odd, the optimal haplotype reconstruction can be determined in time $\mathcal{O}(lm)$ using the formulae

$$h^1[i] = \underset{b\in\{0,1\}}{\text{argmax}} = \left|\left\{j \in \{1,\ldots,l\} : h_j^{1+o_j}[i] = b\right\}\right| \tag{3}$$

and

$$h^2[i] = \underset{b\in\{0,1\}}{\text{argmax}} = \left|\left\{j \in \{1,\ldots,l\} : h_j^{2-o_j}[i] = b\right\}\right|. \tag{4}$$

Hence, finding the consensus haplotyping is polynomial-time equivalent to the task of determining the ordering vector $o$ corresponding to the best haplotype reconstruction $\{h^1, h^2\}$.

The straightforward way to find the optimal ordering is to evaluate the quality of each of the $2^{l-1}$ non-equivalent orderings. The quality of a single ordering can be evaluated in time $\mathcal{O}(lm)$. Hence, the consensus haplotyping can be found in total time $\mathcal{O}(lm+2^l lm')$. The runtime can be reduced to $\mathcal{O}(lm+2^l m')$ by using Gray codes [Sav97] to enumerate all bit vectors $o$ in such order that consecutive bit vectors differ only by one bit. Hence, the problem is fixed-parameter tractable in $l$ (i.e., the number of methods).

### 3.3 Experiments with Ensemble Methods

Consensus haplotypings can be used to combine haplotypings produced by different systems along the lines of ensemble methods in statistics. In practice, genetics researchers often face the problem that different haplotype reconstruction methods give different results and there is no straightforward way to decide which method to choose. Due to the varying characteristics of haplotyping datasets, it is unlikely that one haplotyping method is generally superior. Instead, different methods have different relative strengths and weaknesses, and will fail in different parts of the reconstruction. The promise of ensemble methods lies in "averaging out" those errors, as far as they are specific to a small subset of methods (rather

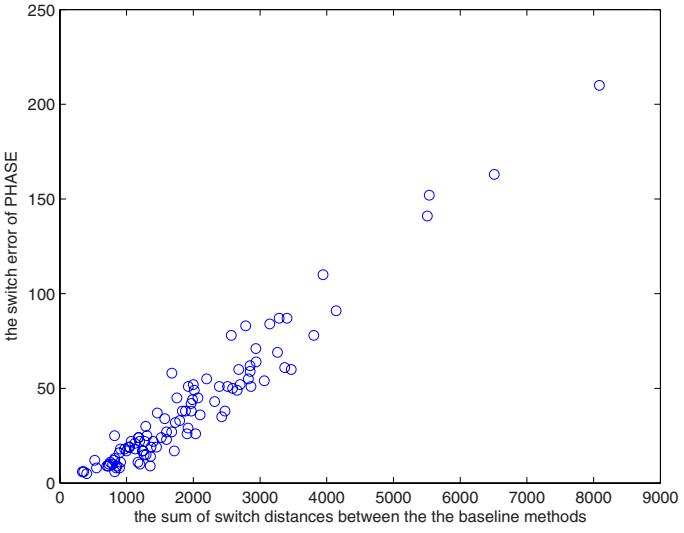


**Fig. 5.** The switch error of PHASE vs. the sum of the switch distances between the baseline methods for the Yoruba datasets. Each point corresponds to one of the Yoruba datasets, x-coordinate being the sum of distances between the reconstructions obtained by the baseline methods, and y-coordinate corresponding to the switch errors of the reconstructions by PHASE.

than a systematic error affecting all methods). This intuition can be made precise by making probabilistic assumptions about how the reconstruction methods err: If the errors in the reconstructions were small random perturbations of the true haplotype pair, taking a majority vote (in an appropriate sense depending on the type of perturbations) of sufficiently many reconstructions would with high probability correct all the errors.

Table 4 lists the reconstruction results for the haplotyping methods introduced in Section 2 on the Daly, Yoruba and HaploDB [HMK+07] datasets, and results for an ensemble method based on all individual methods (Ensemble) and all individual methods except the slow PHASE system (Ensemble w/o PHASE). The ensemble methods simply return the consensus haplotype pair based on switch distance. For the HaploDB dataset, we sampled 100 distinct marker sets of 100 markers each from chromosome one. The 74 available haplotypes in the data set were paired to form 37 individuals.

It can be observed that the ensemble method generally tracks with the best individual method, which varies for different datasets. Furthermore, if PHASE is left out of the ensemble to reduce computational complexity, results are still close to that of the best method including PHASE (Daly,Yoruba) or even better (HaploDB).

Distance functions on haplotypings can also be used to compute estimates of confidence for the haplotype reconstructions for a particular population. Figure 5 shows that there is a strong correlation between the sum of distances

between the individuals methods (their "disagreement") and the actual, normally unknown reconstruction error of the PHASE method (which was chosen as reference method as it was the most accurate method overall in our experiments). This means that the agreement of the different haplotyping methods on a given population is a strong indicator of confidence for the reconstructions obtained for that population.

## 4 Structure Discovery

The main reason for determining haplotype data for (human) individuals is to relate the genetic information contained in the haplotypes to phenotypic traits of the individual, such as susceptibility to certain diseases. Furthermore, haplotype data yields insight into the organization of the human genome: how individual markers are inherited together, the distribution of variation in the genome, or regions which have been evolutionary conserved (indicating locations of important genes). At the data analysis level, we are therefore interested in analyzing the structure in populations—to determine, for example, the difference in the genetic make-up of a case and a control population—and the structure in haplotypes, e.g. for finding evolutionary conserved regions. In the rest of this section, we will briefly outline approaches to these structure discovery tasks, and in particular discuss representational challenges with haplotype and population data.

### 4.1 Structure in Populations

The use of haplotype pairs to infer structure in populations is relevant for relating the genetic information to phenotypical properties, and to predict the phenotypical properties based on the genetic information. The main approaches for determining structure in populations are classification and clustering.

As mentioned in the introduction, the main problem with haplotype data is that the data for each individual contains two binary sequences, where each position has a different interpretation. Hence, haplotype data can be considered to consist of unordered pairs of binary feature vectors, with sequential dependencies between nearby positions in the vector (the markers that are close to each other can, for example, be located on the same gene).

A simple way to propositionalize the data is to neglect the latter, i.e., the sequential dependence in the vectors. In that case the unordered pair of binary vectors is transformed into a ternary vector with symbols $\{0,0\}$, $\{0,1\}$, and $\{1,1\}$. However, the dependences between the markers are relevant. Hence, a considerable fraction of the information represented by the haplotypes is then neglected, resulting in less accurate data analysis results.

Another option is to fix the order of the vectors in each pair. The problem in that case is that the haplotype vectors are high-dimensional and hence fixing a total order between them is tedious if not impossible. Alternatively, both ordered pairs could be added to the dataset. However, then the data analysis technique has to take into account that each data vector is in fact a pair of unordered data vectors, which is again non-trivial.

The representational problems can be circumvented considering only the distances/similarities between the haplotype pairs, employing distance functions such as those we defined in the previous section. For example, nearest-neighbor classification can be conducted solely using the class labels and the inter-point distances. Distance information also suffices for hierarchical clustering. Furthermore, K-means clustering is also possible when we are able to compute the consensus haplotype pair for a collection of haplotype pairs. However, the distance functions are unlikely to grasp the fine details of the data, and in genetic data the class label of the haplotype pair (e.g., case/control population in gene mapping) can depend only on a few alleles. Such structure would be learnable e.g. by a rule learner, if the data could be represented accordingly.

Yet another approach is to transform the haplotype data into tabular from by feature extraction. However, that requires some data-specific tailoring and finding a reasonable set of features is a highly non-trivial task, regardless of whether the features are extracted explicitly or implicitly using kernels.

The haplotype data can, however, be represented in a straightforward way using relations. A haplotype pair $\{h^1, h^2\}$ is represented simply by a ternary predicate $m(i, j, h^i[j]), i = 1, 2, j = 1, \ldots, m$. This avoids the problem of fixing an order between the haplotypes, and retains the original representation of the data. Given this representation, probabilistic logical learning techniques could be used for classification and clustering of haplotype data. Some preliminary experiments have indicated that using such a representation probabilistic logic learning methods can in principle be applied to haplotype data, and this seems to be an interesting direction for future work.

### 4.2 Structure in Haplotypes

There are two main dimensions of structure in haplotype data: horizontal and vertical. The vertical dimension, i.e., structure in populations, has been briefly discussed in the previous section. The horizontal dimension corresponds to linear structure in haplotypes, such as segmentations. In this section, we will briefly discuss approaches for discovering this kind of structure.

Finding segmentation or block structure in haplotypes is considered one of the most important tasks in the search for structure in genomic sequences [DRS$^+$01, GSN$^+$02]. The idea for discovering the underlying block structure in haplotype data is to segment the markers into consecutive blocks in such a way that most of the recombination events occur at the segment boundaries. As a first approximation, one can group the markers into segments with simple (independent) descriptions. Such block structure detection takes the chemical structure of the DNA explicitly into account, assuming certain bonds to be stronger than others, whereas the genetic marker information is handled only implicitly. On the other hand, the genetic information in the haplotype markers could be used in conjunction with the similarity measures on haplotypes described in Section 3 to find haplotype segments, and consensus haplotype fragments for a given segment.

The haplotype block structure hypothesis has been criticized for being overly restrictive. As a refinement of the block model, mosaic models have been

suggested. In mosaics there can be different block structures in different parts of the population, which can be modeled as a clustered segmentation [GMT04] where haplotypes are clustered and then a block model is found for each cluster. Furthermore, the model can be further refined by taking into account the sequential dependencies between the consecutive blocks in each block model and the shared blocks in different clusters of haplotypes. This can be modeled conveniently using a Hidden Markov Model [KKM+04]. Finally, the HMM can be extended also to take into account haplotype pairs instead of individual haplotypes.

A global description of the sequential structure is not always necessary, as the relevant sequential structure can concern only a small group of markers. Hence, finding frequent patterns in haplotype data, i.e., finding projections of the haplotype pairs on small sets of markers such that the projections of at least a $\sigma$-fraction of the input haplotype pairs agree with the projection for given $\sigma > 0$ is of interest. Such patterns can be discovered by a straightforward modification of the standard level-wise search such as described in [MT97]. For more details on these approaches, please refer to the cited literature.

## 5 Conclusions

A haplotype can be considered a projection of (a part of) a chromosome to those positions for which there is variation in a population. Haplotypes provide cost-efficient means for studying various questions, ranging from the quest to identify genetic roots of complex diseases to analyzing the evolution history of populations or developing "personalized" medicine based on the individual genetic disposition of the patient. Haplotype data for an individual consists of an unordered pair of haplotypes, as cells carry two copies of each chromosome (maternal and paternal information). This intrinsic structure in haplotype data makes it difficult to apply standard propositional data analysis techniques to this problem. In this chapter, we have studied how (probabilistic) relational/structured data analysis techniques can overcome this representational difficulty, including Logical Hidden Markov Models (Section 2) and methods based on distances between pairs of vectors (Section 3).

In particular, we have proposed the SpaMM system, a new statistical haplotyping method based on Logical Hidden Markov Models, and shown that it yields competitive reconstruction accuracy. Compared to the other haplotyping systems used in the study, the SpaMM system is relatively basic. It is based on a simple Markov model over haplotypes, and uses the logical machinery available in Logical Hidden Markov Models to handle the mapping from propositional haplotype data to intrinsically structured genotype data. A level-wise learning algorithm inspired by the Apriori data mining algorithm is used to construct sparse models which can overcome model complexity and data sparseness problems encountered with high-order Markov chains. We furthermore note that using an embedded implementation LOHMMs can also be competitive with other special-purpose haplotyping systems in terms of computational efficiency.

Finally, we have discussed approaches to discovering structure in haplotype data, and how probabilistic relational learning techniques could be employed in this field.

**Acknowledgments.** We wish to thank Luc De Raedt, Kristian Kersting, Matti Kääriäinen and Heikki Mannila for helpful discussions and comments, and Sampsa Lappalainen for help with the experimental study.

## References

[AMS+96] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press (1996)

[DRS+01] Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., Lander, E.S.: High-Resolution Haplotype Structure in the Human Genome. Nature Genetics 29, 229–232 (2001)

[EGT04] Eronen, L., Geerts, F., Toivonen, H.: A Markov Chain Approach to Reconstruction of Long Haplotypes. In: Altman, R.B., Dunker, A.K., Hunter, L., Jung, T.A., Klein, T.E. (eds.) Biocomputing 2004, Proceedings of the Pacific Symposium, Hawaii, USA, 6-10 January 2004, pp. 104–115. World Scientific, Singapore (2004)

[EGT06] Eronen, L., Geerts, F., Toivonen, H.: HaploRec: efficient and accurate large-scale reconstruction of haplotypes. BMC Bioinformatics 7, 542 (2006)

[FG06] Flum, J., Grohe, M.: Parameterized Complexity Theory. In: EATCS Texts in Theoretical Computer Science, Springer, Heidelberg (2006)

[GMT04] Gionis, A., Mannila, H., Terzi, E.: Clustered segmentations. In: 3rd Workshop on Mining Temporal and Sequential Data (TDM) (2004)

[GSN+02] Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E.S., Daly, M.J., Altshuler, D.: The structure of haplotype blocks in the human genome. Science 296(5576), 2225–2229 (2002)

[HBE+04] Halldórsson, B.V., Bafna, V., Edwards, N., Lippert, R., Yooseph, S., Istrail, S.: A survey of computational methods for determining haplotypes. In: Istrail, S., Waterman, M.S., Clark, A. (eds.) DIMACS/RECOMB Satellite Workshop 2002. LNCS (LNBI), vol. 2983, pp. 26–47. Springer, Heidelberg (2004)

[HMK+07] Higasa, K., Miyatake, K., Kukita, Y., Tahira, T., Hayashi, K.: D-HaploDB: A database of definitive haplotypes determined by genotyping complete hydatidiform mole samples. Nucleic Acids Research 35, D685–D689 (2007)

[Hud02] Hudson, R.R.: Generating samples under a wright-fisher neutral model of genetic variation. Bioinformatics 18, 337–338 (2002)

[KDR06] Kersting, K., De Raedt, L., Raiko, T.: Logical hidden markov models. Journal for Artificial Intelligence Research 25, 425–456 (2006)

[KKM+04] Koivisto, M., Kivioja, T., Mannila, H., Rastas, P., Ukkonen, E.: Hidden markov modelling techniques for haplotype analysis. In: Ben-David, S., Case, J., Maruoka, A. (eds.) ALT 2004. LNCS (LNAI), vol. 3244, pp. 37–52. Springer, Heidelberg (2004)

[KLLM07] Kääriäinen, M., Landwehr, N.: Sampsa Lappalainen, and Taneli Mielikäinen. Combining haplotypers. Technical Report C-2007-57, Department of Computer Science, University of Helsinki (2007)

[KS05] Kimmel, G., Shamir, R.: A Block-Free Hidden Markov Model for Genotypes and Its Applications to Disease Association. Journal of Computational Biology 12(10), 1243–1259 (2005)

[LME+07] Landwehr, N., Mielikäinen, T., Eronen, L., Toivonen, H., Mannila, H.: Constrained hidden markov models for population-based haplotyping. BMC Bioinformatics (to appear, 2007)

[MT97] Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)

[RKMU05] Rastas, P., Koivisto, M., Mannila, H., Ukkonen, E.: A hidden markov technique for haplotype reconstruction. In: Casadio, R., Myers, G. (eds.) WABI 2005. LNCS (LNBI), vol. 3692, pp. 140–151. Springer, Heidelberg (2005)

[Sav97] Savage, C.: A survey of combinatorial gray codes. SIAM Review 39(4), 605–629 (1997)

[SS05] Stephens, M., Scheet, P.: Accounting for Decay of Linkage Disequilibrium in Haplotype Inference and Missing-Data Imputation. The American Journal of Human Genetics 76, 449–462 (2005)

[SS06] Scheet, P., Stephens, M.: A Fast and Flexible Statistical Model for Large-Scale Population Genotype Data: Applications to Inferring Missing Genotypes and Haplotypic Phase. The American Journal of Human Genetics 78, 629–644 (2006)

[SWS05] Salem, R., Wessel, J., Schork, N.: A comprehensive literature review of haplotyping software and methods for use with unrelated individuals. Human Genomics 2, 39–66 (2005)

[The05] The International HapMap Consortium. A Haplotype Map of the Human Genome. Nature, 437, 1299–1320 (2005)

[TJHBD97] Thompson Jr., J.N., Hellack, J.J., Braver, G., Durica, D.S.: Primer of Genetic Analysis: A Problems Approach, 2nd edn. Cambridge University Press, Cambridge (1997)

[WBCT05] Wang, W.Y.S., Barratt, B.J., Clayton, D.G., Todd, J.A.: Genome-wide association studies: Theoretical and practical concerns. Nature Reviews Genetics 6, 109–118 (2005)