Efficient Haplotype Inference with Pseudo-Boolean Optimization [1]

Ana S. Graça[1], João Marques-Silva[2], Inês Lynce[1] and Arlindo L. Oliveira[1]

[1] IST/INESC-ID, Technical University of Lisbon, Portugal

[2] School of Electronics and Computer Science, University of Southampton, UK

Abstract

Haplotype inference from genotype data is a key computational problem in bioinformatics, since retrieving directly haplotype information from DNA samples is not feasible using existing technology. One of the methods for solving this problem uses the pure parsimony criterion, an approach known as Haplotype Inference by Pure Parsimony (HIPP). Initial work in this area was based on a number of different Integer Linear Programming (ILP) models and branch and bound algorithms. Recent work has shown that the utilization of a Boolean Satisfiability (SAT) formulation and state of the art SAT solvers represents the most efficient approach for solving the HIPP problem.
Motivated by the promising results obtained using SAT techniques, this work investigates the utilization of modern Pseudo-Boolean Optimization (PBO) algorithms for solving the HIPP problem. Applying PBO to existing ILP models, the results are promising, and motivate the development of a new PBO model (RPoly) for the HIPP problem, which has a compact representation and eliminates key symmetries. Experimental results indicate that RPoly outperforms the SAT-based approach on most problem instances, being, in general, significantly more efficient.

Haplotype Inference by Pure Parsimony (HIPP)

The human genome is constituted by pairs of chromosomes, with one element inherited from each parent. The conflated data of both chromosomes on a pair is the genotype, while the genetic information of a single chromosome is the haplotype. A genotype is not always equal to the respective haplotypes due to SNPs (Single Nucleotide Polymorphisms). The value of a particular SNP may be X, Y or X/Y, depending on whether the organism is homozygous with allele X, homozygous with allele Y or heterozygous. To understand the genetic contribution to diseases and their origins, it is often more informative to have haplotype information rather than genotype data. However, using currently available techniques, it is not feasible to examine separately copies of chromosomes. The challenge is to infer haplotype data from genotype data.

	genotype	T	A/G	C	C/T	G	A	C/T
individual 1	haplotype 1	T	G	C	T	G	A	C
	haplotype 2	T	A	C	C	G	A	T
individual 2	haplotype 1	T	A	C	T	G	A	T
	haplotype 2	T	G	C	C	G	A	C

We represent each haplotype by a string over the alphabet {0,1}, where 0 represents the wild type allele, while 1 represents the mutant. Each genotype is represented by a string over {0,1,2}. Homozygous sites are represented by the values 0 or 1, depending on whether both haplotypes have value 0 or 1 at that site, respectively. Heterozygous sites are represented by value 2. For example,

genotype 0 2 1 2 0 1 2

can be explained by

haplotype 1 0 1 1 0 0 1 0

and haplotype 2 0 0 1 1 0 1 1

Different methods have been proposed for the problem of haplotype inference. The Haplotype Inference by Pure-Parsimony (HIPP) approach aims at finding a solution to the problem that minimizes the total number of distinct haplotypes required. This problem is APX-hard.
Given a set $\mathcal{G}$ of n genotypes, each of length m, the HIPP problem consists in finding a minimum-size set $\mathcal{H}$ of haplotypes that explain all genotypes in $\mathcal{G}$.
For example, explain genotypes 2120, 2102 and 1221
A possible solution (using 6 haplotypes):

$$2120 = 0100 \oplus 1110$$
$$2102 = 1100 \oplus 0101$$
$$1221 = 1011 \oplus 1101$$

A pure parsimony solution (using 4 haplotypes):

$$2120 = 0100 \oplus 1110$$
$$2102 = 0100 \oplus 1101$$
$$1221 = 1011 \oplus 1101$$

Haplotype Inference by Pure Parsimony Models

The first ILP model proposed for the HIPP problem was *RTIP*. Although being efficient for small size instances, RTIP is exponential on the population size n, and therefore this model is inadequate for larger problem instances due to its complexity. RTIP inspired a branch-and-bound algorithm to the problem, known as *Hapar*.
A more recent ILP model, *PolyIP*, is polynomial on n and m. Anyway, PolyIP, used with CPLEX, is severely limited on the size of the problems it could handle.
Recently, a SAT based approach for this problem, *SHIPs*, has shown that the use of effective constraint satisfaction methods leads to an efficient solution of this problem.
This work explores the utilization of modern Pseudo-Boolean Optimization (PBO) algorithms for solving the HIPP problem, originating two promising approaches *PolyPB* and *RPoly*.

Pseudo-Boolean Models

From an ILP point of view, PBO, also known as 0-1 integer programming, can be seen as a specialization to ILP where all variables are Boolean and all coefficients are integer.

$$\begin{array}{ll} \textit{minimize} & \sum_j c_j x_j; \\ \textit{subject to} & \sum_j a_{ij} x_j \geq b_i; \\ & x_j \in \{0,1\}; \\ & a_{ij}, b_i, c_j \in \mathbb{Z}; \end{array}$$

Given that the HIPP ILP models are also PBO models, PBO solvers can be considered. PolyPB is the result of applying MiniSAT+ to the PolyIP model.

As the results are promising, PolyPB motivates the development of a new PBO model more efficient and with a more compact representation. Reduced Poly model (RPoly) associates two haplotypes with each genotype, and conditions are defined which capture when a different haplotype is used for explaining a given genotype. Since the values of the haplotypes explaining the homozygous sites are identified beforehand, RPoly only associates variables with heterozygous sites.
In practice, the model associates two haplotypes, h_i^a and h_i^b, with each genotype g_i, and these haplotypes are required to explain g_i. Moreover, the model associates a variable $t_{i\,j}$ with each heterozygous site (i, j). Hence, $t_{i\,j} = 1$ indicates that $h^a_{i\,j} = 1$ and $h^b_{i\,j} = 0$, whereas $t_{i\,j} = 0$ indicates that $h^a_{i\,j} = 0$ and $h^b_{i\,j} = 1$. The value of h_i^a and h_i^b at homozygous sites j is implicitly assumed.
Variable $x^{p\,q}_{i_1\,i_2}$, with $p, q \in \{a, b\}$ and $1 \leq i_2 < i_1 \leq n$, is 1 if the p haplotype of genotype i_1 and the q haplotype of genotype i_2 are different. The conditions are all of the following form, for all $1 \leq j \leq m$,

$$\neg(R \leftrightarrow S) \rightarrow x^{p\,q}_{i_1\,i_2},$$

where the predicates R and S depend on the values of the sites (i_1, j) and (i_2, j), and on which of the haplotypes is considered, i.e. either a or b.

- If $g_{i_1\,j} \neq 2$, then $R = (g_{i_1\,j} \leftrightarrow (q \leftrightarrow a))$ and $S = t_{i_2\,j}$.
- If $g_{i_2\,j} \neq 2$, then $R = (g_{i_2\,j} \leftrightarrow (p \leftrightarrow a))$ and $S = t_{i_1\,j}$.
- If $g_{i_1\,j} = 2 \wedge g_{i_2\,j} = 2$, then $R = (p \leftrightarrow q)$ and $S = (t_{i_1\,j} \leftrightarrow t_{i_2\,j})$.

In addition, the model uses variables to denote when one of the haplotypes associated with a given genotype is different from all previous haplotypes. Hence, u_i^p, with $p \in \{a, b\}$ and $1 \leq i \leq n$, is 1 if haplotype p of genotype i is different from all previous haplotypes,

$$\bigwedge_{1 \leq k < i} (x^{p\,a}_{i\,k} \wedge x^{p\,b}_{i\,k}). \rightarrow u_i^p$$

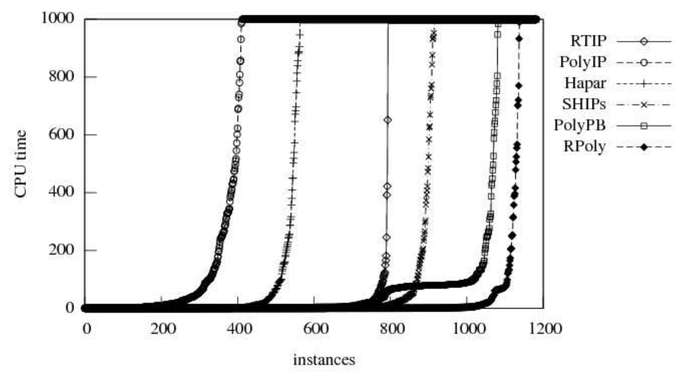
Finally, the cost function is given by

$$\text{minimize} \sum_{i=1}^{n} (u_i^a + u_i^b).$$

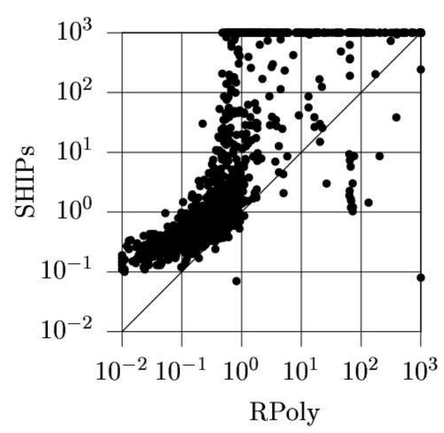
The proposed modifications result in significantly smaller PBO problem instances. The number of terms in RPoly is a factor of 5 to 10 smaller than in PolyPB.

Experimental Results

A comparison of the performance of alternative approaches to the HIPP problem (RTIP, PolyIP, Hapar, SHIPs and the new ones, PolyPB and RPoly) is summarized in the figure below. The run times for each solver were sorted and plotted, the cutoff point being 1000 seconds.A universe of 1183 problem instances is used.



For most problem instances, RPoly is faster than all other solvers and only aborts 43 instances out of 1183. Although, usually, SHIPs is faster than PolyPB, PolyPB only aborts 100 instances, while SHIPs aborts 268. RTIP aborts 389 instances, Hapar aborts 619 instances and PolyIP aborts 771 instances.



This scatter plot, with the run time for RPoly and SHIPs on each of the problem instances with a timeout of 1000 seconds, clearly show that RPoly is significantly more robust than SHIPs. For most problem instances (1089 out of 1183), RPoly is faster than SHIPs and RPoly aborts on a significantly smaller number of instances, being able to solve more than 96% of the problem instances.

Conclusions

We conclude that by replacing the CPLEX ILP solver with the PBO solver MiniSAT+, the existing PolyIP model is shown to be competitive with the state-of-the-art methods. RPoly is a new PBO model for the HIPP problem which entails a number of improvements to the basic PolyIP model. The results for RPoly are significantly more promising: RPoly is most often faster than SHIPs and is also significantly more robust, aborting only on a small number of problem instances.

References

[1] A. Graça, J. Marques-Silva, I. Lynce, and A. Oliveira. Efficient haplotype inference with pseudo-Boolean optimization. In *Algebraic Biology 2007(AB 2007)*, pages 125–139, July 2007.