Solving the Haplotype Inference Problem in Bioinformatics with Answer Set Programming

Marvin Beese, mabeese@uni-potsdam.de, 786300

University of Potsdam

Abstract. A field of Computer Science is Bioinformatics, where well-known problems can be solved with the use of Answer Set Programming (ASP). This summary examines the Haplotype Inference by Pure Parsimony Problem with its biological background and looks into an ASP encoding that solves it.

1 Introduction

Bioinformatics is a big field in Computer Science with natural science application. There exist some challenging problems in Bioinformatics and Answer Set Programming (ASP) can be used for effective modelling and solving some of these problems.
This summary is based on the article "ASP Applications in Bio-informatics: A Short Tour" by Alessandro Dal Palù, Agostino Dovier, Andrea Formisano and Enrico Pontelli [1], which gives an overview of the application of ASP to Genomics, Structure Prediction and Systems Biology. In this summary, I look into the field of Genomics regarding Problem Description Genes and the determination of Genotypes. First, I introduce the biological background, on the basis of which I present the definition of Haplotype Inference by Pure Parsimony (HIPP) - Problem. Finally, I provide a look into the general ASP encoding of the HIPP - Problem.

2 Biological Background

Desoxyribo Nucleic Acid (DNA) is a sequence of nucleotides that encodes biologically relevant information. The set s of nucleotides contains Adenine, Cytosine, Guanine and Thymine, here considered as $s \in \{A, C, G, T\}$, whereas the complementary sequence $\overline{s}$ contains the reversed order of s with the substitution of A and T, and C and G. Both sequences fold together into the double helix. A genome is the complete string of DNA, which contains some regions, known as genes that encode information needed to define proteins. Chromosomes are regions of the DNA that include the Problem Description Genes. Diploid organisms like humans possess two copies of almost all chromosomes, where each pair consists of inherited chromosomes of the mother and the father. These chromosomes are largely coincident, but some typical positions carry the basis

to mutations, which are referred to as Single Nucleotide Polymorphism (SNP) with the combination of the nucleic bases C-T and A-G. The DNA sequence that is inherited from one parent is called a Haplotype and the pairing of two corresponding Haplotypes of the child is called a Genotype.

3 Haplotype Inference by Pure Parsimony with ASP

As the mutation in the SNP of a Genotype is assumed to be deterministic, this enables a boolean representation of the SNP using 0 for C and A, and 1 for T and G. When the two haplotypes carry different information for the same position, the resulting Genotype then holds the value 2.
The Haplotype Inference by Pure Parsimony (HIPP) Problem therefore is defined for a given set of equal length Genotypes G as the answer to the search for a set of Haplotypes H that "explain" G. The following example shows how two Haplotypes of the parents h_{2*i-1} and h_{2*i} result in one Genotype of the child g_i.

$$h_1 = 01011 \wedge h_2 = 01110 \Rightarrow g_1 = 01212, \quad h_3 = 01101 \wedge h_4 = 01111 \Rightarrow g_2 = 01121$$

But with $h_5 = 01010$, the search for Haplotypes given $G = \{g_1, g_2\}$ not only results in $H_1 = \{h_1, h_2, h_3, h_4\}$ but also in $H_2 = \{h_3, h_4, h_5\}$.

$$h_3 \wedge h_4 \Rightarrow g_2, \quad h_4 \wedge h_5 \Rightarrow g_1$$

In such a situation, the main goal is to find a set of Haplotypes of minimum cardinality, like H_2 in this example.
The general ASP encoding defines a predicate $g(I, J, B)$ and $h(I, J, B)$, where the Genotype g_I and the Haplotype, h_I respectively, hold for position J the value of $bit(B)$, which is either $0, 1$ or 2.
As a set of Genotypes is given per definition, the determination of Haplotypes happens through rules. The following rules define that both Haplotypes hold the value B of the Genotype, which is either 0 or 1 and cannot be both.

```
h(2*I-1,J,B) :- g(I,J,B), bit(B).
h(2*I, J,B) :- g(I,J,B), bit(B).
1 {h(2*I-1,J,0); h(2*I-1,J,1)} 1 :- g(I,J,2).
```

It could also be that the Haplotypes hold different values for B, so that the Genotype holds the value 2, which is handled in the next rule.

```
h(2*I,J,1-B) :- g(I,J,2), h(2*I-1,J,B), bit(B).
```

With the use of given predicates, different Haplotypes $differenthablo(A, B)$ are defined.

```
differenthaplo(A,B) :-
    haplo(A), haplo(B),
    site(S), bit(X), bit(Y),
    h(A,S,X), h(B,S,Y), X != Y.
```

Representative Haplotypes are selected, which are the ones with the lowest index.

```
cover_someone(B) :-
    haplo(A), haplo(B), A < B,
    not differenthaplo(A,B).
representative_haplo(A) :-
    haplo(A),
    not cover_someone(A).
```

Finally, the number of representative Haplotypes is minimized.

```
#minimize {A:representative_haplo(A)}
```

4 Conclusion

The Haplotype Inference Problem shows, that there can be an easy encoding of difficult biology problems using ASP. This is an example of a real-life representation of mutation through the investigation of chromosomes, which has an encoding of eight lines of code. The problem was solved by Erdem [2].

References

1. Alessandro Dal Palù, Agostino Dovier, Andrea Formisano, Enrico Pontelli: ASP Applications in Bio-informatics: A Short Tour. KI 32(2-3): 157-164 (2018)
2. Erdem E, Türe F (2008) Efficient haplotype inference with answer set programming.