Exploring Life through Logic Programming

Alessandro Dal Palú Agostino Dovier
Andrea Formisano Enrico Pontelli

Dept. Computer Science, New Mexico State University, USA

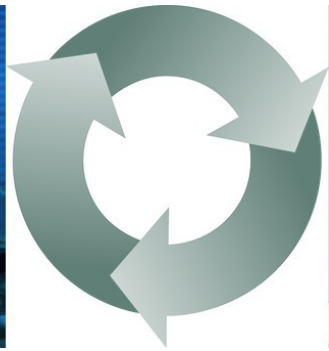
University of Udine, Italy
December 2015

Introduction

- Computer Science is now the root of all sciences

  "All Science is Computer Science" G. Johnson, New York Times, 3.25.2001

- Biology is an incredible source of challenging problems for computer science
- Problems are often hidden or vaguely defined and emerge only after cycles of feedback with biologists, physicists, chemists, etc

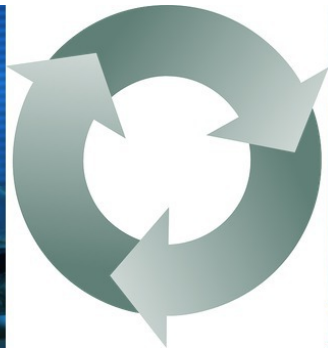




Introduction

- Computer Science is now the root of all sciences

  "All Science is Computer Science" G. Johnson, New York Times, 3.25.2001

- Biology is an incredible source of challenging problems for computer science
- Problems are often hidden or vaguely defined and emerge only after cycles of feedback with biologists, physicists, chemists, etc


- Solving one of these problems can be of unpredictable importance for life sciences and medicine

Introduction

- Intuitively: Bioinformatics = Computer Science $\cap$ Biology

Bioinformatics

Bioinformatics deals with modeling and solving problems, analyzing and filtering data, from biology and related life sciences.

- Computations are extensive.
- Data availability is huge.
- Data is affected by experimental errors.
- Computer science tools should help in analyzing and filtering.

Introduction

Bioinformatics applications can be divided in three categories:

1) Support infrastructure for analysis and experiments

Computational tools for automated environments for workflow management, description and annotation of experiments, reporting requirements, . . .

2) Polynomial time solvable problems

The input size is large: e.g. string matching problems over DNA sequences.

3) Intractable problems

NP-complete or worse problems. Mainly covered by this lecture.

Introduction

Bioinformatics applications can be divided in three categories:

1) Support infrastructure for analysis and experiments

Final Lecture

2) Polynomial time solvable problems

The input size is large: e.g. string matching problems over DNA sequences.

3) Intractable problems

NP-complete or worse problems. Mainly covered by this lecture.

Introduction

Bioinformatics applications can be divided in three categories:

1) Support infrastructure for analysis and experiments

Computational tools for automated environments for workflow management, description and annotation of experiments, reporting requirements, . . .

2) Polynomial time solvable problems

The input size is large: e.g. string matching problems over DNA sequences.

3) Intractable problems

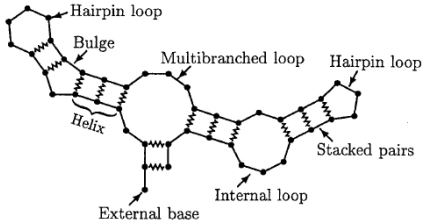
Mainly covered by the next two lectures.

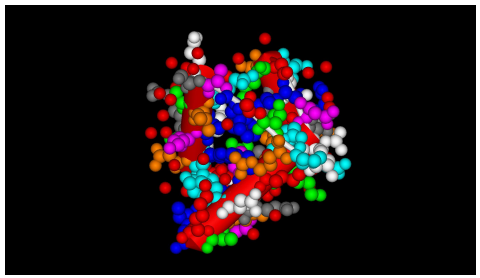
Areas of Bioinformatics

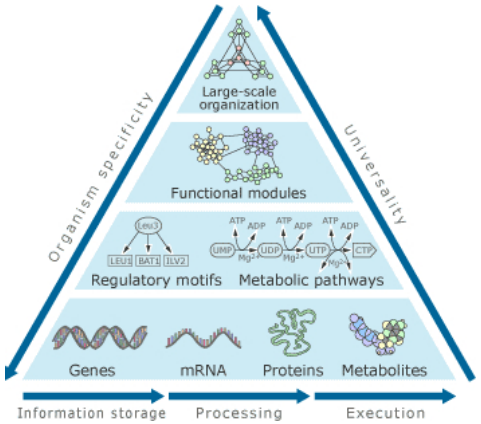
1. Genomics. Study of the genomes. Huge amount of data, fast algorithms (not always), limited to sequence analysis.

...	G	A	T	C	T	G	T	A	C	T	G	A	G	T	...
...	G	A	T	C	T	G	T	A	C	T	G	A	A	T	...

2. Structural Bioinformatics. Study of the folding process of bio-molecules. Less structural data than sequence data available.







3. Systems Biology. Study of complex interactions in biological systems. High level of representation.

Why Logic Programming?

(At least) three main reasons:

- Models are rarely stable and static. Logic programming provides the level of elaboration-tolerance to support model modifications and incremental addition of new knowledge.
- Linear Programming is not enough (in particular for modeling energy models)
- Declarative formalism is elegant and concise!
- Extensive possibilities for refinement of models—parallelism, heuristics, . . .

What we'll see in more details

Challenging Problems

- Genomics:
  - ✓ Haplotype Inference
  - ✓ Phylogenetic trees
- Structural Bioinformatics:
  - ✓ RNA secondary structure prediction
  - ✓ Protein structure prediction (on lattice)
- Systems Biology:
  - ✓ Reasoning on Biological Networks

What we'll see in more details

Challenging Problems

Support Infrastructure

- Evolutionary Informatics
- The EvoIO Stack

What we'll see in more details

Challenging Problems

Support Infrastructure

Parallelism in Logic Programming

- The Past of Parallel Logic Programming
- The Present of Parallel Logic Programming
- Current Directions in Parallel Logic Programming

ASP encoding

⇒ We present modelings and working codes in ASP. You can download them at
`www.unipr.it/~dalpalu/corsi/CILC15/index.html`

- The same models can be encoded in CLP(FD) with almost no changes. It requires Prolog encoding. Left as exercise :)

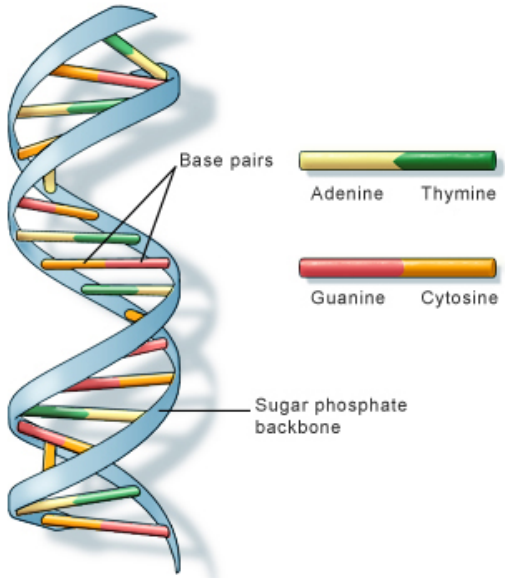
Some introductory references

- P. Clote and R. Backofen. *Computational Molecular Biology*. An Introduction. Wiley, 2000.
- Nice introductory slides by Sebastian Will `math.mit.edu/classes/18.417/Slides/intro.pdf`
- F. Crick. *Central dogma of molecular biology.* Nature, 227, 1970.
- A. Lesk. *Introduction to Bioinformatics.* Oxford Univ. Press, 2008.
- X. Xia. *Bioinformatics and the Cell: Modern Computational Approaches in Genomics, Proteomics and Transcriptomics.* Springer, 2007.

Haplotype inference
and a crash course in genomics...

DNA and Genome in a nutshell

- DNA (DeoxyriboNucleic Acid) is characterized by a string of nucleotides: A, C, G, and T (Adenine, Cytosine, Guanine, Thymine)
- Given a sequence $s \in \{A, C, G, T\}^*$ the complementary sequence $\bar{s}$ is deterministically obtained by reversing s and substituting $A \leftrightarrow T$ and $C \leftrightarrow G$
- s and $\bar{s}$ fold together forming the famous double helix

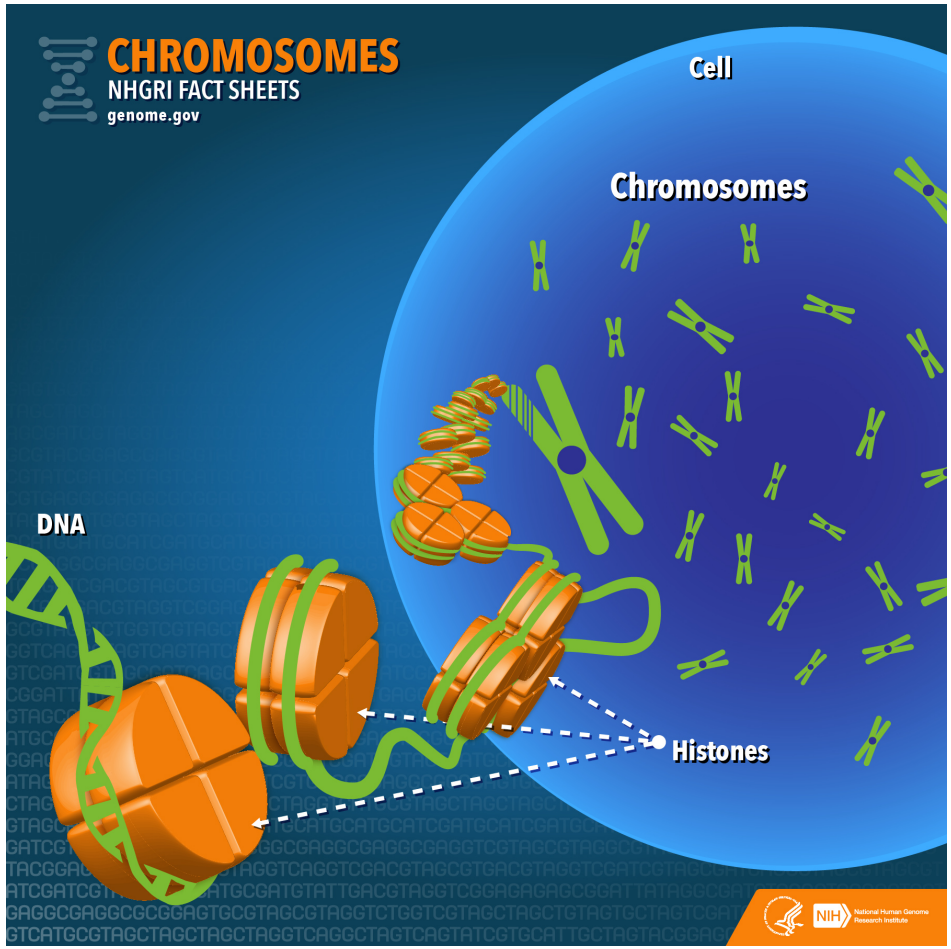


DNA and Genome in a nutshell

- DNA strings are long (10^6–10^{10} nucleotides).
- Differences between the DNAs of two members of the same specie are small (e.g., 1 in 1000 for humans)
- Some fragments of the DNA, called genes, encode proteins (More on this later).
- The set of all genes of an individual is called genome
- The Human Genome is estimated to contain between 20,000 and 25,000 genes.
- Differences of some nucleotides in the same gene define variants of proteins that characterize a property of an individual w.r.t. another.

Haplotype Inference

- DNA is packaged in compact units, called Chromosomes
- Genes are packaged in bundles inside chromosomes.



- In Diploid organisms (like humans) there are almost identical chromosome pairs.
  - Each pair is made of an inherited chromosome from the father and another from the mother.
  - Homologous chromosomes carry same genes in the same order, but with small trait differences.
- A haplotype is a DNA sequence that has been inherited from one parent.
- A genotype is a pairing of two corresponding haplotypes.

Haplotype Inference

Each person inherits two haplotypes for most regions of the genome:

- One from the mother
- One from the father

...	G	A	T	C	T	G	T	A	C	T	G	A	G	T	...
...	G	A	T	C	T	G	T	A	C	T	G	A	A	T	...

Haplotype Inference

Each person inherits two haplotypes for most regions of the genome:

- One from the mother
- One from the father

···	G	A	T	C	T	G	T	A	C	T	G	A	G	T	···
···	G	A	T	C	T	G	T	A	C	T	G	A	A	T	···

In some typical positions, the bases are subject to mutations.

Haplotype Inference

Each person inherits two haplotypes for most regions of the genome:

- One from the mother
- One from the father

···	G	A	T	C	T	G	T	A	C	T	G	A	G	T	···
···	G	A	T	C	T	G	T	A	C	T	G	A	A	T	···

In some typical positions, the bases are subject to mutations.
In the most common case, there is a Single Nucleotide Polymorphism (SNP).

Haplotype Inference

Each person inherits two haplotypes for most regions of the genome:

- One from the mother
- One from the father

$\cdots$	G	A	T	C	T	G	T	A	C	T	G	A	G	T	$\cdots$
$\cdots$	G	A	T	C	T	G	T	A	C	T	G	A	A	T	$\cdots$

In some typical positions, the bases are subject to mutations.
In the most common case, there is a Single Nucleotide Polymorphism (SNP).

Mutations are $C \leftrightarrow T$ and $A \leftrightarrow G$

Haplotype Inference

Each person inherits two haplotypes for most regions of the genome:

- One from the mother
- One from the father

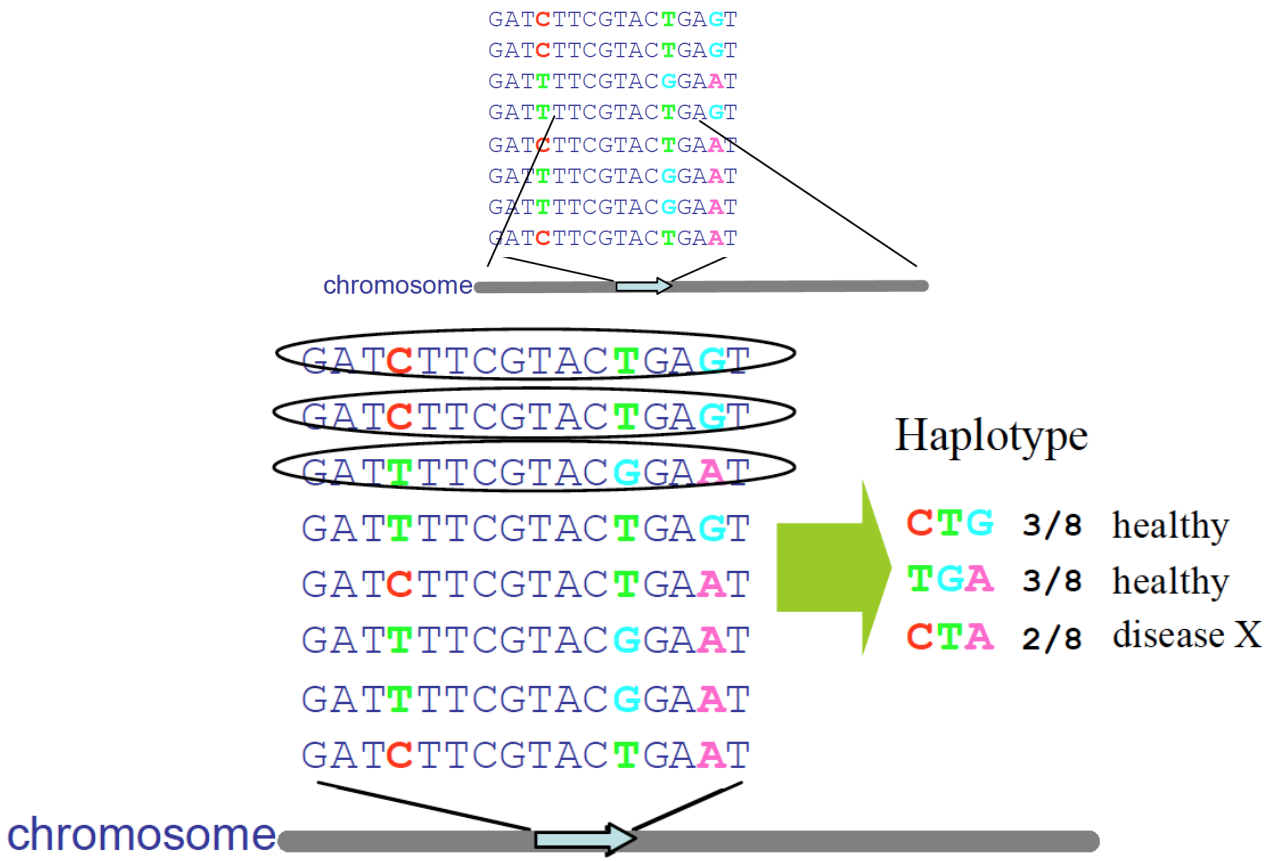
···	G	A	T	C	T	G	T	A	C	T	G	A	G	T	···
···	G	A	T	C	T	G	T	A	C	T	G	A	A	T	···
				⇑						⇑			⇑		

In some typical positions, the bases are subject to mutations.
In the most common case, there is a Single Nucleotide Polymorphism (SNP).

Mutations are $C \leftrightarrow T$ and $A \leftrightarrow G$

A variant a gene is called an Allele.

Haplotype Inference



Haplotype Inference

- A good introduction in `http://csiflabs.cs.ucdavis.edu/~gusfield/gusfieldorzack.pdf`
- Typically the genome (genotype) is easier and cheaper to be obtained.
- Haplotypes are more powerful discriminators between cases and controls in disease association studies
- Use of haplotypes in disease association studies reduces the number of tests to be carried out.
- We need computational methods to guess haplotypes
- The Haplotype Inference problem is introduced to investigate genetic variations in a population.
- Typically SNPs sites are the target of the analysis

Haplotype Inference

Single Nucleotide Polymorphism (SNP)

Each person has two haplotypes (from the mother and from the father) for most regions of the genome:

G	A	A	T	C	T	T	C	G	T	A	C	T	G	A	G	T
G	A	A	T	C	T	T	C	G	T	A	C	T	G	A	A	T

Let us focus on the SNPs:

A	C	T	G
A	C	T	A

We encode SNPs according to: $A \mapsto 0$ $C \mapsto 0$ $G \mapsto 1$ $T \mapsto 1$

Haplotype Inference

Single Nucleotide Polymorphism (SNP)

Each person has two haplotypes (from the mother and from the father) for most regions of the genome:

G	A	A	T	C	T	T	C	G	T	A	C	T	G	A	G	T
G	A	A	T	C	T	T	C	G	T	A	C	T	G	A	A	T

Let us focus on the SNPs:

A	C	T	G
A	C	T	A

We encode SNPs according to: $A \mapsto 0$ $C \mapsto 0$ $G \mapsto 1$ $T \mapsto 1$

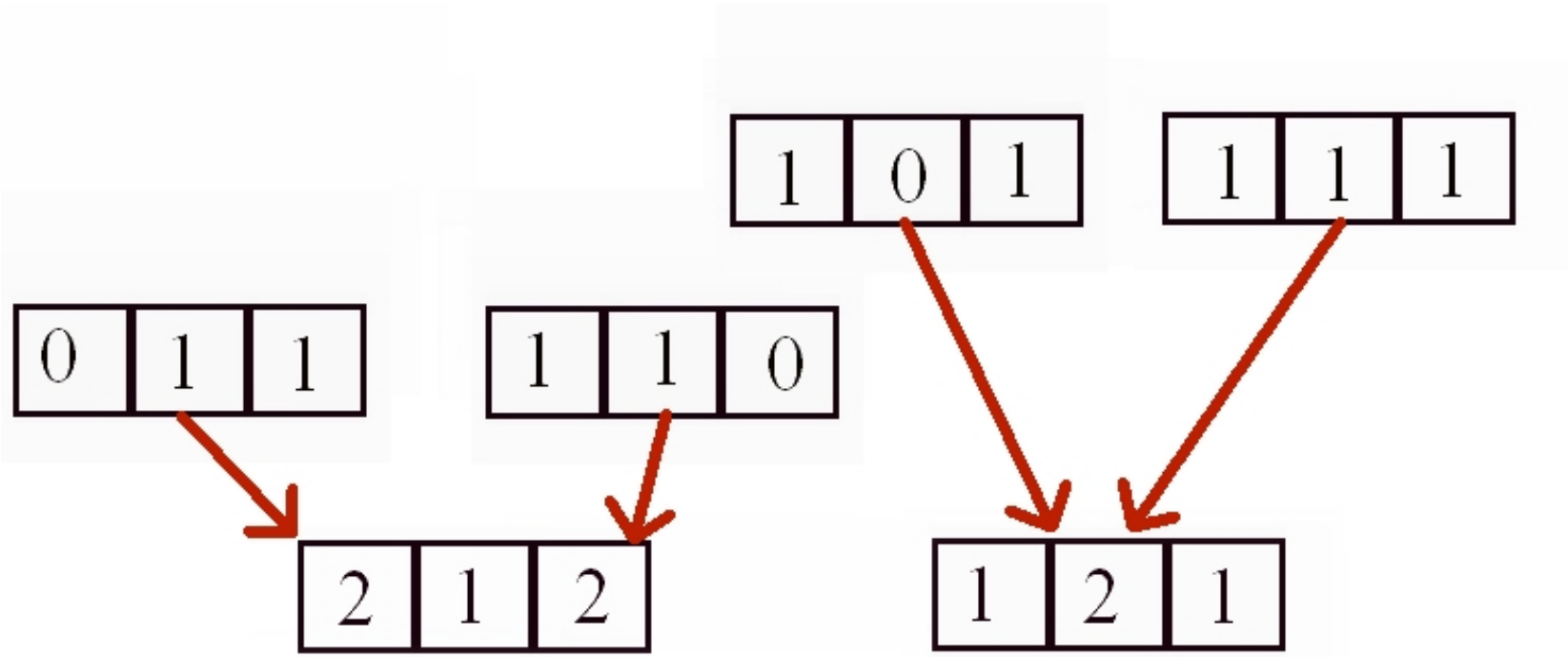
0	0	1	1
0	0	1	0

0	0	1	2

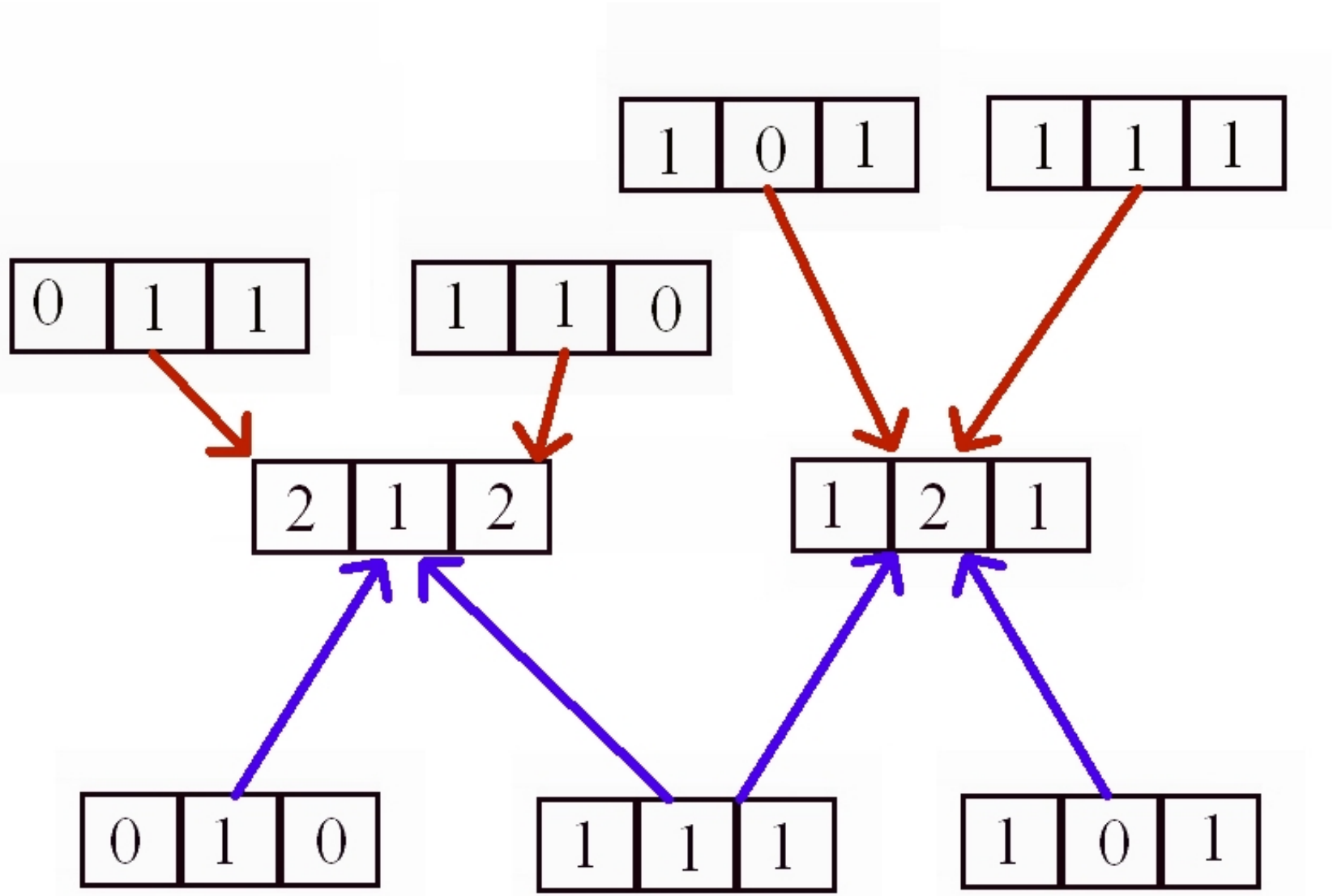
The genotype is set to 2 if there is a mismatch

Haplotype Inference

Haplotype Inference



Haplotype Inference



Haplotype Inference

- A string of $\{0,1\}^*$ is called a *haplotype*
- A string of $\{0,1,2\}^*$ is called a *genotype*
- Two equal length haplotypes generate a unique genotype
- The rules are $0 \oplus 0 = 0$, $1 \oplus 1 = 1$, $0 \oplus 1 = 2$

Haplotype Inference

- A string of $\{0,1\}^*$ is called a *haplotype*
- A string of $\{0,1,2\}^*$ is called a *genotype*
- Two equal length haplotypes generate a unique genotype
- The rules are $0 \oplus 0 = 0$, $1 \oplus 1 = 1$, $0 \oplus 1 = 2$
  E.g., $0010, 0101 \Rightarrow 0222$

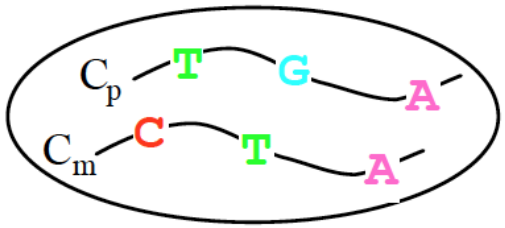
Haplotype Inference

- A string of $\{0,1\}^*$ is called a *haplotype*
- A string of $\{0,1,2\}^*$ is called a *genotype*
- Two equal length haplotypes generate a unique genotype
- The rules are $0 \oplus 0 = 0$, $1 \oplus 1 = 1$, $0 \oplus 1 = 2$
  E.g., $0010, 0101 \Rightarrow 0222$
- If we have a genotype, we can only conjecture (potentially exponentially many) *haplotype*s that generated it
  (observe that, e.g., $0110, 0001 \Rightarrow 0222$)

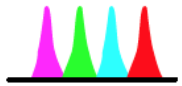
Haplotype Inference

- A string of $\{0,1\}^*$ is called a *haplotype*
- A string of $\{0,1,2\}^*$ is called a *genotype*
- Two equal length haplotypes generate a unique genotype
- The rules are $0 \oplus 0 = 0$, $1 \oplus 1 = 1$, $0 \oplus 1 = 2$
  E.g., $0010, 0101 \Rightarrow 0222$
- If we have a genotype, we can only conjecture (potentially exponentially many) *haplotype*s that generated it
  (observe that, e.g., $0110, 0001 \Rightarrow 0222$)
- Biological experiments allow us to know genotypes!
- Investigating *sets* of genotypes for a population, helps in understanding the relationships between SNPs and physical features as well as medical information
- Since genotypes are introduced in evolution, it is reasonable to find minimal sets of haplotypes explaining the known genotypes.

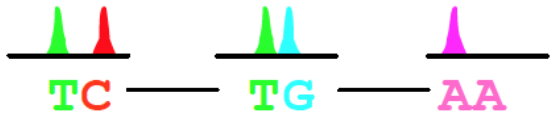
Haplotype Inference



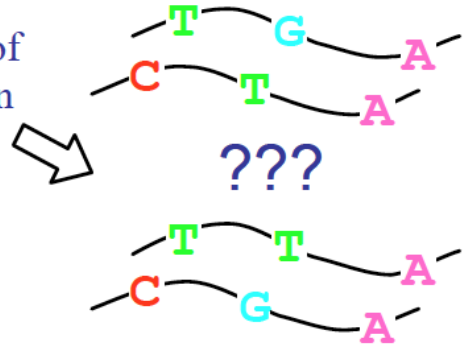
Heterozygous diploid individual



sequencing



Genotype g
pairs of alleles with association of alleles to chromosomes unknown



haplotype $h \equiv (h_1, h_2)$
possible associations of alleles to chromosome

This is a mixture modeling problem!

Haplotype Inference

- Let $H = \{\{0,1\}^n\}$ be the set of *haplotypes* and
- $G = \{\{0,1,2\}^n\}$ be the set of *genotypes*.
- Given $h_1, h_2 \in H$ and $g \in G$, $\{h_1, h_2\}$ explains g if and only if $|h_1| = |h_2| = |g|$ and $\forall i \in [1..n]$:

$$\begin{array}{lcl} g[i] \leq 1 & \longrightarrow & h_1[i] = h_2[i] = g[i] \\ g[i] = 2 & \longrightarrow & h_1[i] \neq h_2[i] \end{array}$$

- A set of haplotypes H explains a set of genotypes G if for all $g \in G$ there are $h_1, h_2 \in H$ such that $\{h_1, h_2\}$ explains g.
- Given a set of genotypes G and an integer k, the *haplotype inference problem (HIP)* by pure parsimony is the problem of finding a set H that explains G and such that $|H| = k$ (decision version).

Haplotype Inference

Use of such a parsimony criterion is consistent with the fact that the number of distinct haplotypes observed in most natural populations is vastly smaller than the number of possible haplotypes; this is expected given the plausible assumptions that the mutation rate at each site is small and recombinations rate are low.

[Gusfield and Orzack, 2006]

- The problem is NP-complete. Reduction from vertex cover [LPR04].

Haplotype Inference

The ASP modeling

- m genotypes ($1 \ldots m$) in input; facts:
  - `geno(1 ... m).`
  - `site(1... n).`
  - `g(i,j,k).`
    - i is the i-th genotype ($1 \leq i \leq m$)
    - j is the position within the genotype ($1 \leq j \leq n$) and
    - k is the value in $\{0, 1, 2\}$
- $2m$ inferred haplotypes ($1 \ldots 2m$, not necessarily distinct); facts:
  - `haplo(1 ... 2m).`
- The i-th genotype g_i is explained by haplotypes h_{2i} and h_{2i-1}
- Haplotype `h(i,j).` is in the model if the value at position j is 1

Haplotype Inference

The ASP modeling: Deterministic Case

```
  % If g(i,j)=0 or 1 assign deterministically
  % h(2i) and h(2i-1).
(1)     h(2*I-1,J) :- g(I,J,1).
(2)     h(2*I,J)   :- g(I,J,1).
(3)     :- h(2*I-1,J), g(I,J,0).
(4)     :- h(2*I,J),   g(I,J,0).
```

```
(5) h(2*I-1,J) :- g(I,J,2), not h(2*I,J). (6) h(2*I,J) :- g(I,J,2), not h(2*I-1,J).
```

Haplotype Inference

The ASP modeling

```
(7)    representative_haplo(A) :-
                haplo(A), not cover_someone(A).
(8)    cover_someone(B) :-
                haplo(A), haplo(B), A < B, samehaplo(A,B).
```

- Define the representatives.
- The lowest index haplotype of a set of equal ones is selected as representative.

Haplotype Inference

The ASP modeling

```
(7)    representative_haplo(A) :-
                haplo(A), not cover_someone(A).
(8)    cover_someone(B) :-
                haplo(A), haplo(B), A < B, samehaplo(A,B).

(9)    { samehaplo(A,B) } :- haplo(A), haplo(B), A < B.
(10)   :- samehaplo(A,B), haplo(A), haplo(B), A < B,
              site(S), h(A,S), not h(B,S).
(11)   :- samehaplo(A,B), haplo(A), haplo(B), A < B,
              site(S), not h(A,S), h(B,S).
```

- We have a (possibly empty) set of `samehaplo`.
- Constraints: can't be `samehaplo` and a site S with $\neq$ values.
- `not(_,S)` for the case haplotype at site S is 0.
- Symmetry breaking.

Haplotype Inference

The ASP modeling

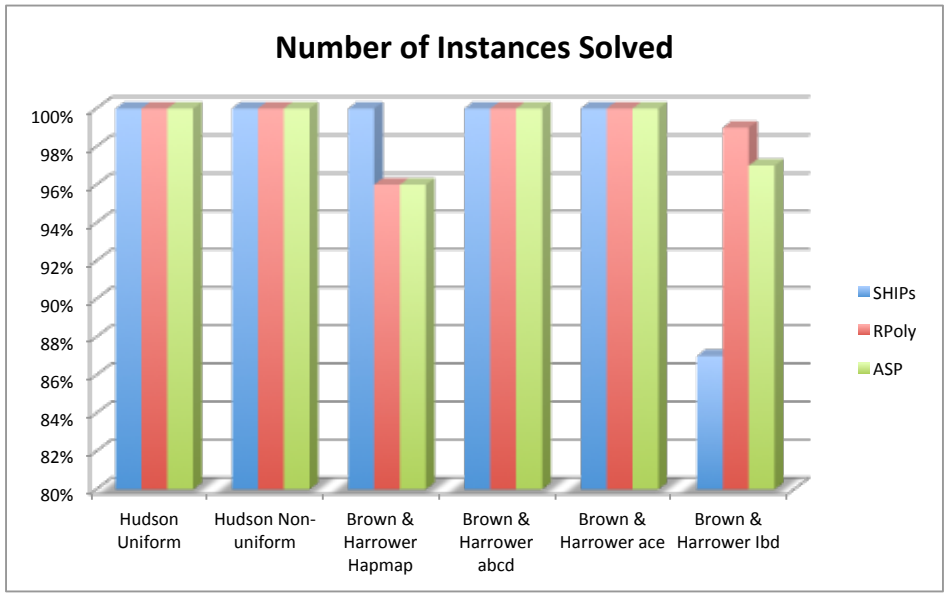
```
  % Count the number of representative and minimize it
(12)   #minimize{ 1,A:representative_haplo(A) }.
```

Considerations

Refined versions of this ASP have been presented in

Erdem, Erdem, Türe. HAPLO-ASP: Haplotype Inference Using Answer Set Programming. LPNMR 2009: 573–578

- Competitive results with state-of-the-art tools



Haplotype Inference

Some References

- Gusfield and Orzack. Haplotype Inference (Survey, and ILP formulations) In CRC Handbook on Bioinformatics, 2006
- Lancia, Pinotti, Rizzi. [LPR04] Haplotyping Populations by Pure Parsimony: Complexity of Exact and Approximation Algorithms. INFORMS Journal on Computing 16(4):348–359, 2004.
- Graça, Marques-Silva, Lynce, Oliveira. Several works on SAT-based and specialized 0-1 ILP for Haplotype Inference. (e.g. WCB 08, WCB 09)
- Di Gaspero, Roli. Stochastic local search for large-scale instances of the haplotype inference problem by pure parsimony. J. Algorithms 63(1-3): 55-69 (2008) (also in WCB08).
- Erdem, Erdem, Türe. HAPLO-ASP: Haplotype Inference Using Answer Set Programming. LPNMR 2009: 573–578
- James Cussens Maximum likelihood pedigree reconstruction using integer programming. WCB 10.