Phylogeny- and Parsimony-Based Haplotype Inference with Constraints

Michael Elberfeld Till Tantau

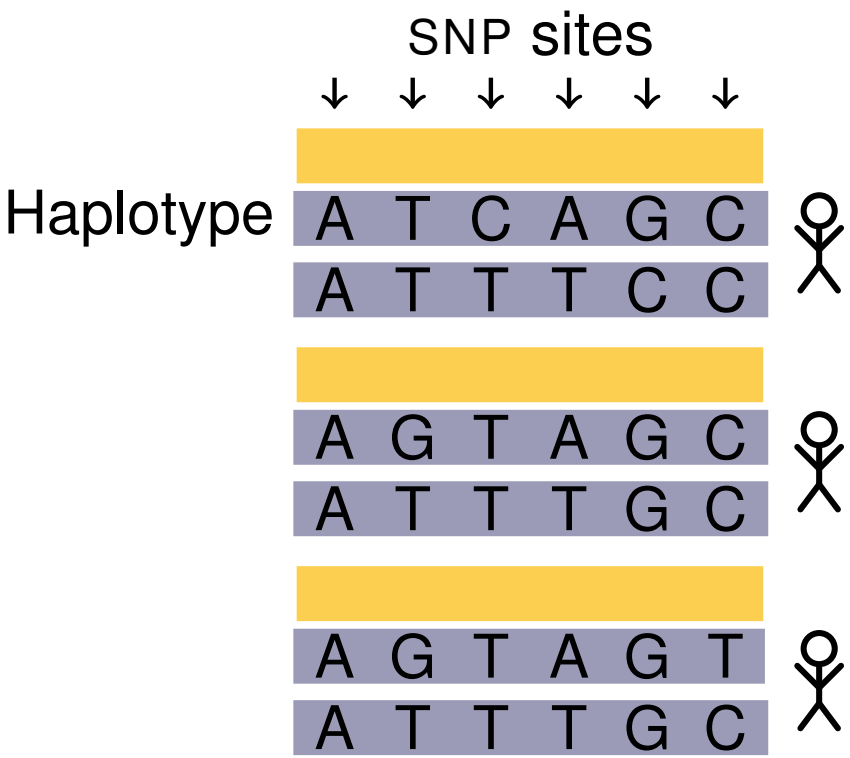
Institute of Theoretical Computer Science
University of Lübeck, Germany

21st Annual Symposium on Combinatorial Pattern Matching
Polytechnic Institute of New York University, USA

22 June 2010

Outline

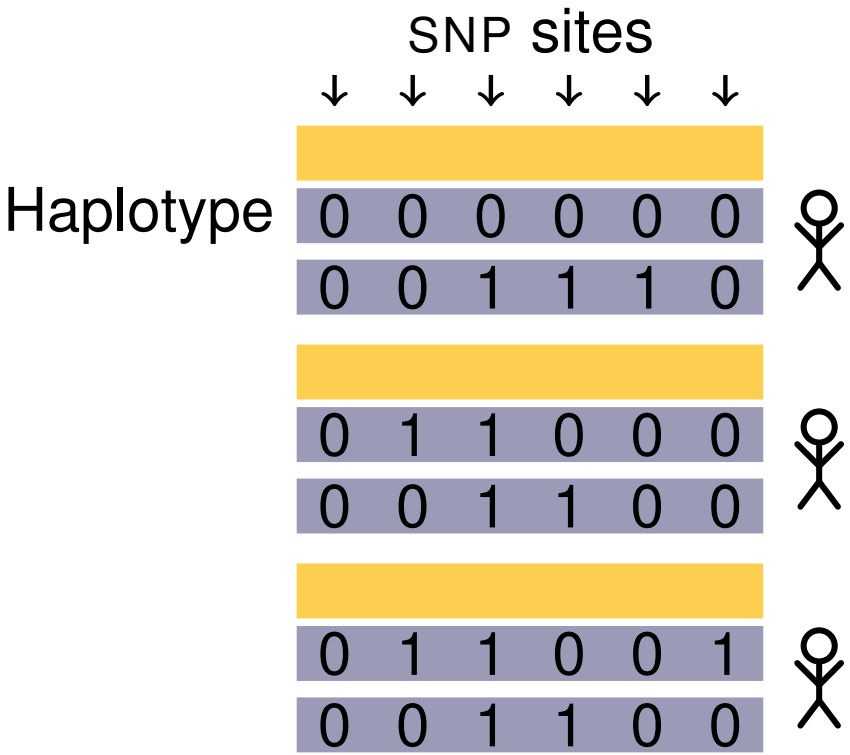
Haplotype Inference With Constraints



- Haplotypes describe genetic information on chromosomes.
- Laboratories provide only genotypes.
- Haplotypes for some genotypes are constrained.

Objective: Predict biologically reasonable haplotypes.

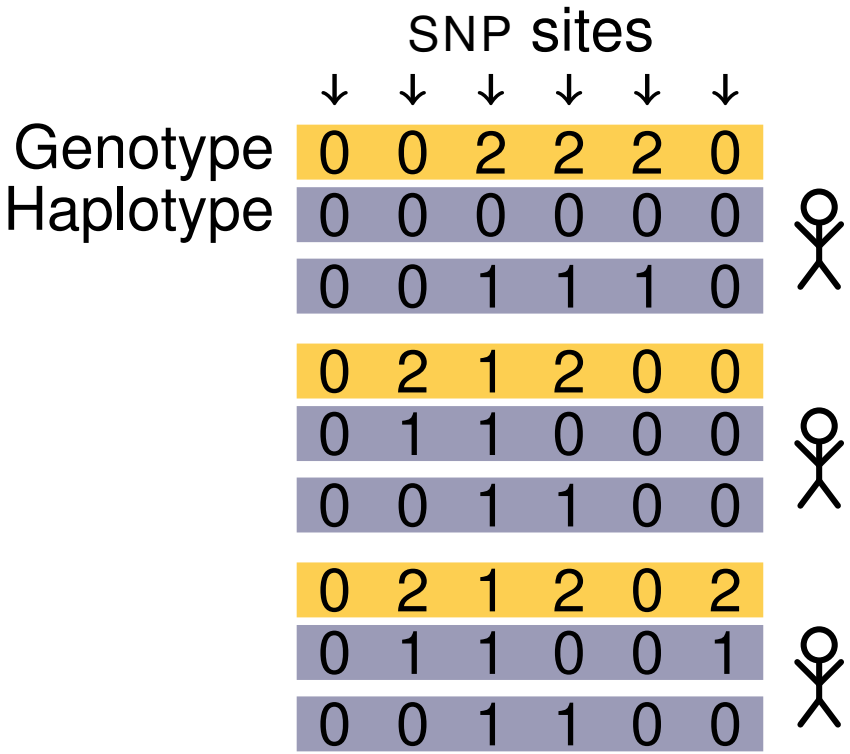
Haplotype Inference With Constraints



- Haplotypes describe genetic information on chromosomes.
- Laboratories provide only genotypes.
- Haplotypes for some genotypes are constrained.

Objective: Predict biologically reasonable haplotypes.

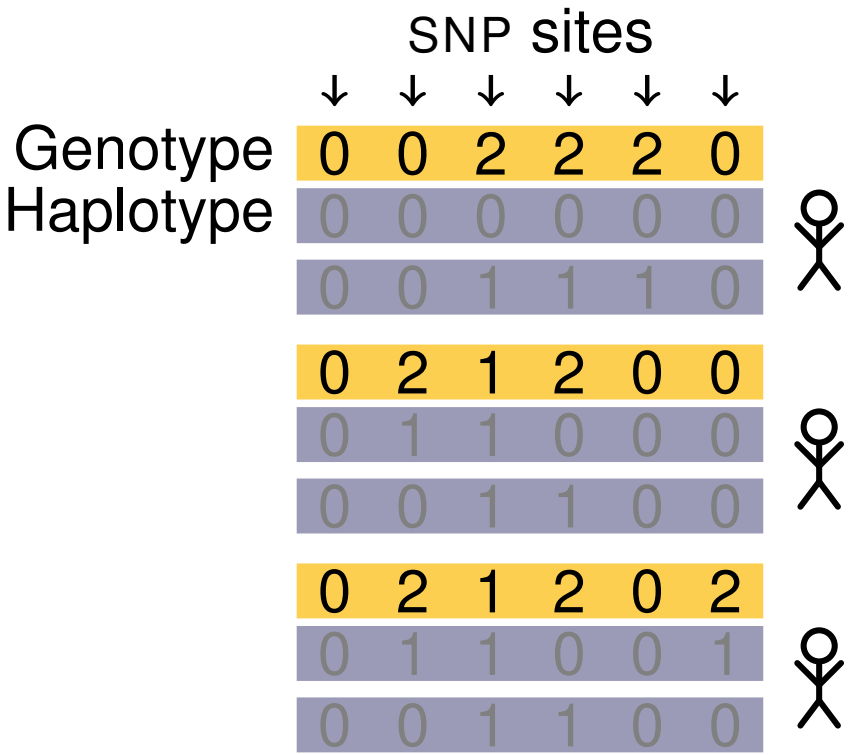
Haplotype Inference With Constraints



- Haplotypes describe genetic information on chromosomes.
- Laboratories provide only genotypes.
- Haplotypes for some genotypes are constrained.

Objective: Predict biologically reasonable haplotypes.

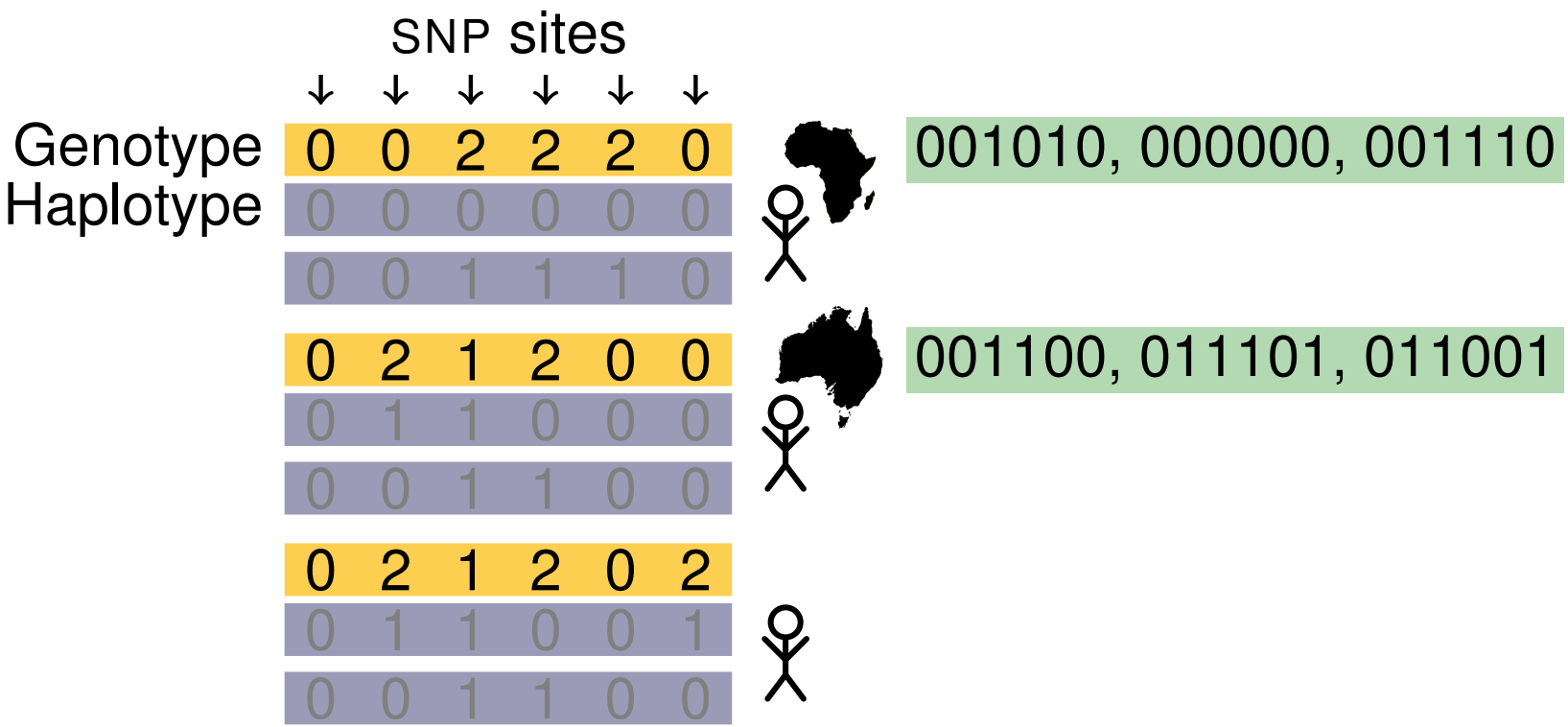
Haplotype Inference With Constraints



- Haplotypes describe genetic information on chromosomes.
- Laboratories provide only genotypes.
- Haplotypes for some genotypes are constrained.

Objective: Predict biologically reasonable haplotypes.

Haplotype Inference With Constraints



- Haplotypes describe genetic information on chromosomes.
- Laboratories provide only genotypes.
- Haplotypes for some genotypes are constrained.

Objective: Predict biologically reasonable haplotypes.

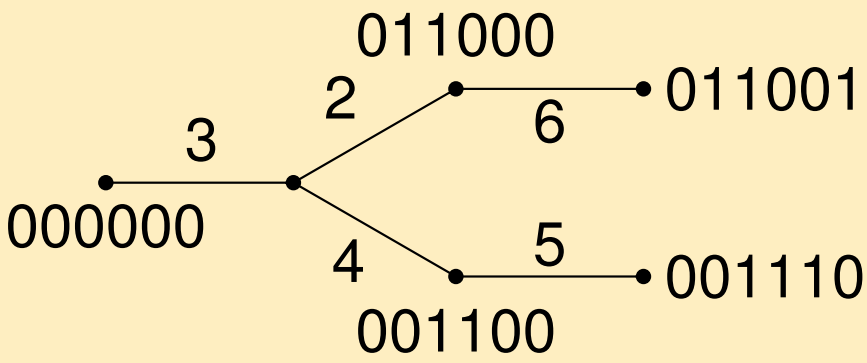
Phylogeny-Based Haplotype Inference

Definition (Perfect Phylogeny, PP)

Haplotypes

000000
011000
011001
001110
001100

Perfect Phylogeny



At most one mutation per site.

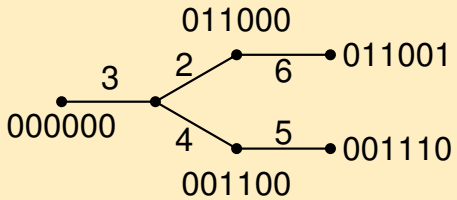
Problem (CONSTRAINED-PPH)

Input: Genotypes and haplotype sets.

002220 H_1
021200 H_2
021202

Output: Haplotypes with a perfect phylogeny, if any exist.

$000000 \in H_1$
$001110 \in H_1$
$011000 \in H_2$
$001100 \in H_2$
011001
001100



Problem Variants and Their Complexity

PPH	No haplotype sets.
POOL-PPH	Same haplotype set for every genotype.
CONSTRAINED-PPH	The general variant.

Theorem ([Gusfield, 2002])

PPH $\in$ P.

Theorem ([Fellows et al., 2009, CPM])

POOL-PPH *for some cases with few 2-entries is in* P.

Theorem

CONSTRAINED-PPH $\in$ P.

Corollary

POOL-PPH $\in$ P.

Parsimony-Based Haplotype Inference

Problem (CONSTRAINED-MINH)

Input: Genotypes and haplotype sets.
Output: Solution with minimum number of different haplotypes.

Problem (CONSTRAINED-MINPPH)

Input: Genotypes and haplotype sets.
Output: Solution with perfect phylogeny and minimum number of different haplotypes.

Complexity with Solution Size as Parameter

MINH $\in$ FPT
[Sharan et al., 2006]

POOL-MINH $\in$ FPT
[Fellows et al., 2009, CPM]

CONSTRAINED-MINH W[2]-hard

MINPPH $\in$ FPT

POOL-MINPPH $\in$ FPT

CONSTRAINED-MINPPH $\in$ FPT

Known results and new results.

Outline

Phylogeny-Based Haplotype Inference

Problem (CONSTRAINED-PPH)

Input: Genotypes and haplotype sets.

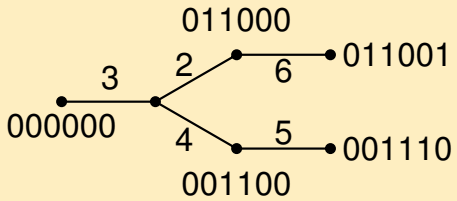
002220 H_1

021200 H_2

021202

Output: Haplotypes with a perfect phylogeny, if any exist.

$000000 \in H_1$
$001110 \in H_1$
$011000 \in H_2$
$001100 \in H_2$
011001
001100



Theorem

CONSTRAINED-PPH $\in$ P.

Proof Plan.

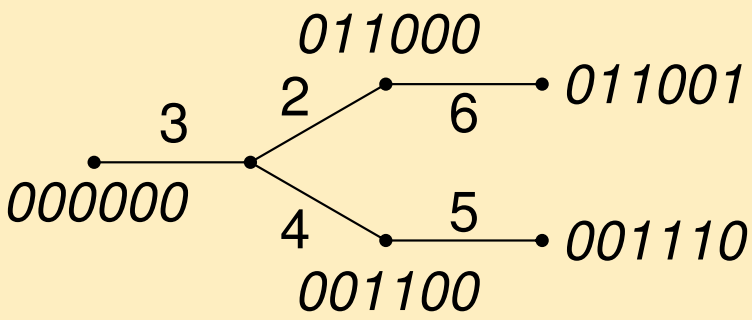
1. 4-gamete characterization of perfect phylogenies.
2. Initial partition into all-2-column instances.
3. Recursive partition into PPH instances.

□

Phylogenies and 4-Gamete Characterization

Lemma (4-Gamete Characterization, Folklore)

Perfect Phylogeny *iff* *No column pair contains all gamete strings 00, 01, 10 and 11.*

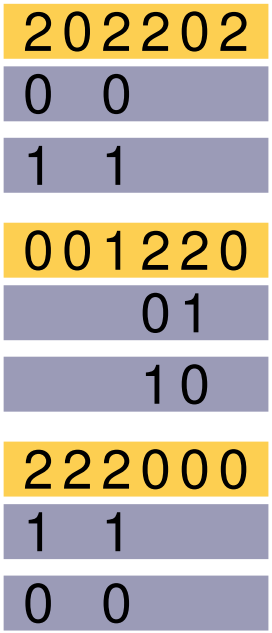


0 0 0 0 0 0
0 1 1 0 0 0
0 1 1 0 0 1
0 0 1 1 1 0
0 0 1 1 0 0

We use the 4-gamete characterization instead of perfect phylogenies.

What are Phase Constraints?

Genotypes



Phase Constraints

unequal(c_1, c_4),
equal(c_3, c_4),
unequal(c_1, c_3)

1. For two 2's in a genotype: Either gametes 00, 11 (equal phase) or 01, 10 (unequal phase) are present.
2. Genotypes with common two 2's are phased in the same way.
3. Some phases are determined by the genotypes.

We store known phases in phase constraints.

Phase Constraints Permit Haplotype Deletion

Genotypes	Haplotype Sets	Phase Constraints
202202	000000, 001100, 100001	unequal(c_1, c_4), equal(c_3, c_4), unequal(c_1, c_3)
001220		
222000	000000, 111000	

1. Haplotype determines phases for all two 2's of a genotype.
2. Phase constraints forbid the use of haplotypes with different phases.

We delete haplotypes that violate phase constraints.

Phase Constraints Permit Haplotype Deletion

Genotypes	Haplotype Sets	Phase Constraints
202202	~~000000~~, 001100, 100001	unequal(c_1, c_4),
001220		equal(c_3, c_4),
222000	~~000000~~, 111000	unequal(c_1, c_3)

1. Haplotype determines phases for all two 2's of a genotype.
2. Phase constraints forbid the use of haplotypes with different phases.

We delete haplotypes that violate phase constraints.

Initial Partition Into All-2-Column Instances

Input consists of genotypes and haplotype sets.

Instance

Genotypes	Haplotype sets
002220	000000, 001110
021200	
021202	001000, 001001, 011101, 001010
001202	101000, 011101, 111000, 001101
201200	
001220	001000, 001110

Initial Partition Into All-2-Column Instances

Sort columns by leaf count [Gusfield, 2002].

Instance

Genotypes	Haplotype sets
220002	000000, 110001
122000	
122200	100000, 100100, 111100, 100100
120200	100010, 111100, 101010, 110100
120020	
120002	100000, 110001

Initial Partition Into All-2-Column Instances

Deduce phase constraints from genotypes.

Instance

Genotypes	Haplotype sets
220002	000000, 110001
122000	
122200	100000, 100100, 111100, 100100
120200	100010, 111100, 101010, 110100
120020	
120002	100000, 110001

Phase Constraints

$\text{equal}(c_1, c_2)$,
$\text{equal}(c_1, c_6)$,
$\text{equal}(c_2, c_6)$,
$\text{unequal}(c_3, c_4)$

Initial Partition Into All-2-Column Instances

Common two 2's have equal phase.

Subinstance for Column c_1

all-2-column

↓

220002	000000, 110001	equal(c_2, c_6), …

Subinstance for Column c_2

all-2-column

↓

122000		equal(c_1, c_2), equal(c_1, c_6), equal(c_2, c_6), unequal(c_3, c_4)
122200	100000, 100100, 111100, 100100	
120200	100010, 111100, 101010, 110100	
120020		
120002	100000, 110001	

Recursive Partition Into PPH Instances

Solve instance with all-2-column.

Instance With All-2-Column

all-2-column ↓

122000	
122200	100000, 100100, 111100, 100100
120200	100010, 111100, 101010, 110100
120020	
120002	100000, 110001

equal(c_1, c_2),
equal(c_1, c_6),
equal(c_2, c_6),
unequal(c_3, c_4)

Recursive Partition Into PPH Instances

Calculate cover columns and the intersection graph.

Instance With All-2-Column

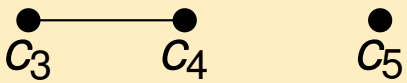
all-2-column ↓

122000	
122200	100000, 100100, 111100, 100100
120200	100010, 111100, 101010, 110100
120020	
120002	100000, 110001

equal(c_1, c_2),
equal(c_1, c_6),
equal(c_2, c_6),
unequal(c_3, c_4)

Cover Columns and Intersection Graph

Cover columns: c_3, c_4, c_5

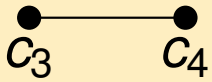
Intersection graph: 

Recursive Partition Into PPH Instances

Solve component instances independently.

Instance for Component 1

Component 1:



122000		equal(c_1, c_2),
122200	100000, 100100, 111100, 100100	equal(c_1, c_6),
120200	100010, 111100, 101010, 110100	equal(c_2, c_6), unequal(c_3, c_4)

Instance for Component 2

Component 2:

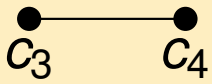
120020	equal(c_2, c_6),
	...

Recursive Partition Into PPH Instances

Branch for different phases between all-2-column and a cover column.

Instance for Component 1

Component 1:



all-2-column

122000		equal(c_1, c_2), equal(c_1, c_6), equal(c_2, c_6), unequal(c_3, c_4)
122200	100000, 100100, 111100, 100100	
120200	100010, 111100, 101010, 110100	

Branch equal(c_2, c_3)

Gives unequal(c_2, c_4)

Branch unequal(c_2, c_3)

Gives equal(c_2, c_4)

Haplotype deletion partitions haplotypes among branches.

Recursive Partition Into PPH Instances

Solve base case with known algorithms.

Instance for Component 2

Component 2: 

1 2 0 0 2 0

equal(c_2, c_6),
...

Fact ([Eskin et al., 2003, Bafna et al., 2003])

PPH *with phase constraints is in* P*.*

Runtime of the Polynomial-Time Algorithm

Lemma

The algorithm solves CONSTRAINED-PPH *in time* $O(p(n+p)m^2)$, *where*

- n *is the number of genotypes,*
- p *is the sum of the sizes of all haplotype sets, and*
- m *is the number of sites.*

Outline

Parsimony-Based Haplotype Inference

Problem (CONSTRAINED-MINH)

Input: Genotypes and haplotype sets.
Output: Solution with minimum number of different haplotypes.

Theorem

CONSTRAINED-MINH is W[2]-hard

Proof Sketch.

1. Reduction from HITTING-SET.
2. Transform edges to genotypes and incident vertices to haplotype pairs.
3. Hitting set of size k corresponds to haplotype set of size $2k$.

□

Parsimony-Based Haplotype Inference

Problem (CONSTRAINED-MINPPH)

Input: Genotypes and haplotype sets.
Output: Solution with perfect phylogeny and minimum number of different haplotypes.

Theorem

CONSTRAINED-MINPPH $\in$ FPT

Proof Sketch.

1. Size-k haplotype sets explain at most $k(k-1)/2$ different genotypes.
2. Consider all $O(k^{2k^2})$ possible ways of how k haplotypes explain $k(k-1)/2$ genotypes.
3. Recursive partition into MINPPH instances that are solved directly.

□

Summary and Open Problems

Results

- CONSTRAINED-PPH $\in$ P. Implementation works fast on real data.
- Settles open problem from [Fellows et al., 2009, CPM]: POOL-PPH $\in$ P.
- CONSTRAINED-MINH is W[2]-hard
- CONSTRAINED-MINPPH $\in$ FPT

Open Problems and Research Directions

- Solve CONSTRAINED-PPH faster: From $O(p(n+p)m^2)$ to $O(p(n+p)m)$, or even $O((n+p)m)$.
- Work with more general types of haplotype constraints like $*$-haplotypes.

References I

Bafna, V., Gusfield, D., Lancia, G., and Yooseph, S. (2003).
Haplotyping as perfect phylogeny: A direct approach.
Journal of Computational Biology, 10(3–4):323–340.

Brown, D. G. and Harrower, I. M. (2006).
Integer programming approaches to haplotype inference by pure parsimony.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3(2):141–154.

Eskin, E., Halperin, E., and Karp, R. M. (2003).
Efficient reconstruction of haplotype structure via perfect phylogeny.
Journal of Bioinformatics and Computational Biology, 1(1):1–20.

References II


Fellows, M. R., Hartman, T., Hermelin, D., Landau, G. M., Rosamond, F. A., and Rozenberg, L. (2009).
Haplotype inference constrained by plausible haplotype data.
In *Proceedings of CPM 2009*, volume 5577 of *LNCS*, pages 339–352. Springer.


Gusfield, D. (2002).
Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions.
In *Proceedings of RECOMB 2002*, pages 166–175. ACM Press.


Sharan, R., Halldórsson, B. V., and Istrail, S. (2006).
Islands of tractability for parsimony haplotyping.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3(3):303–311.