# Efficient Haplotype Inference on Pedigrees and Applications

Tao Jiang

Dept of Computer Science

University of California – Riverside

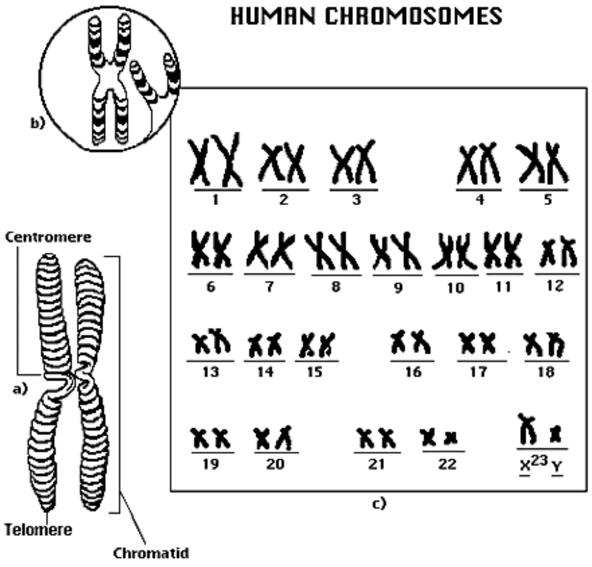
(joint work with Jing Li, CWRU)

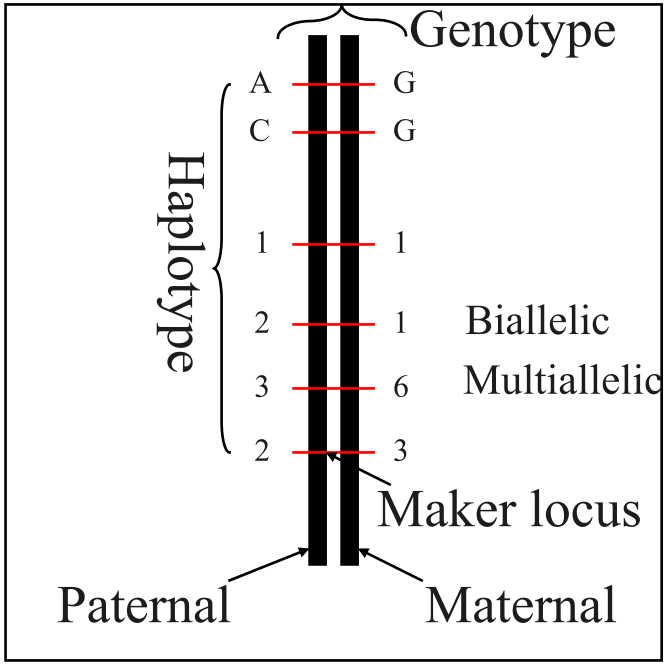
# Outline

- Background
- The MRHC problem and complexity
- An exact algorithm for 0-recombinant data
- A heuristic algorithm (block-extension)
- Integer linear programming formulation and solution for MRHC with missing alleles
- Experimental results and application in disease gene association mapping
- Inference of haplotypes on population data

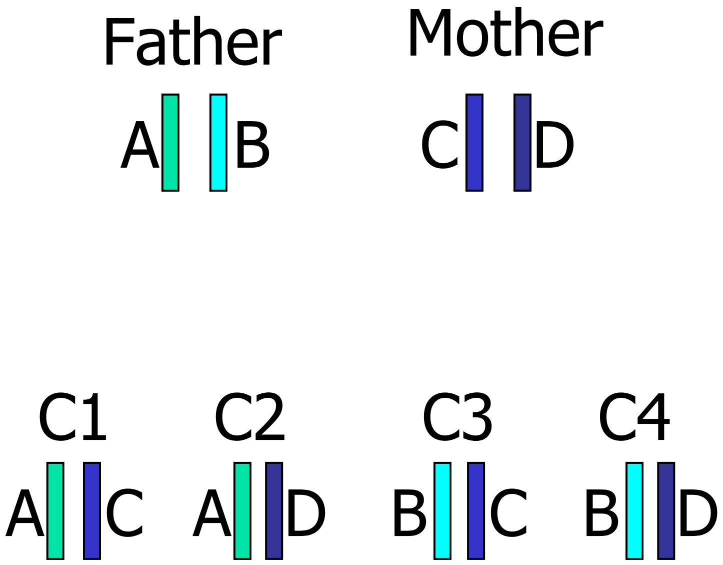
# Terms

- Diploid
- Polymorphisms, marker, allele, and SNP
- Genotype, homozygous & heterozygous
- Haplotype, paternal & maternal haplotypes





# Mendelian Law of Inheritance and Recombination



Father

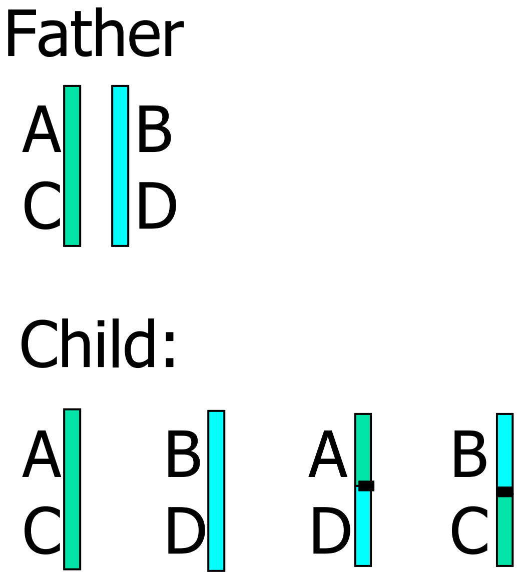
A B

Mother

C D

C1 C2 C3 C4

A C A D B C B D



Father

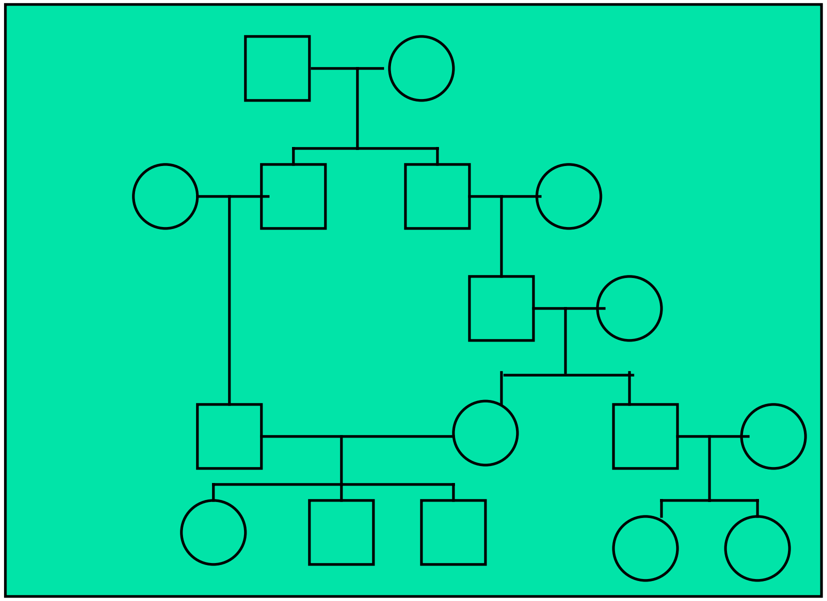
A B
C D

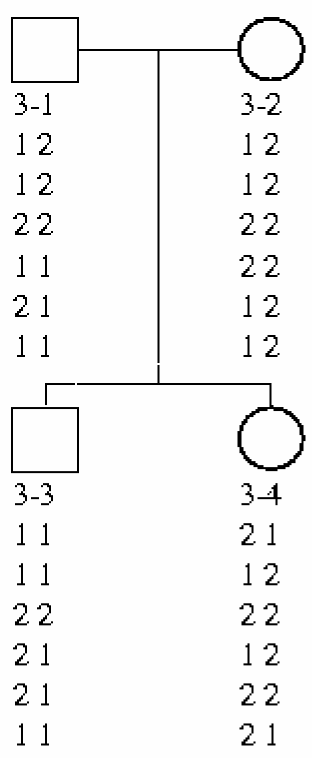
Child:

A B A B
C D D C

# Pedigree

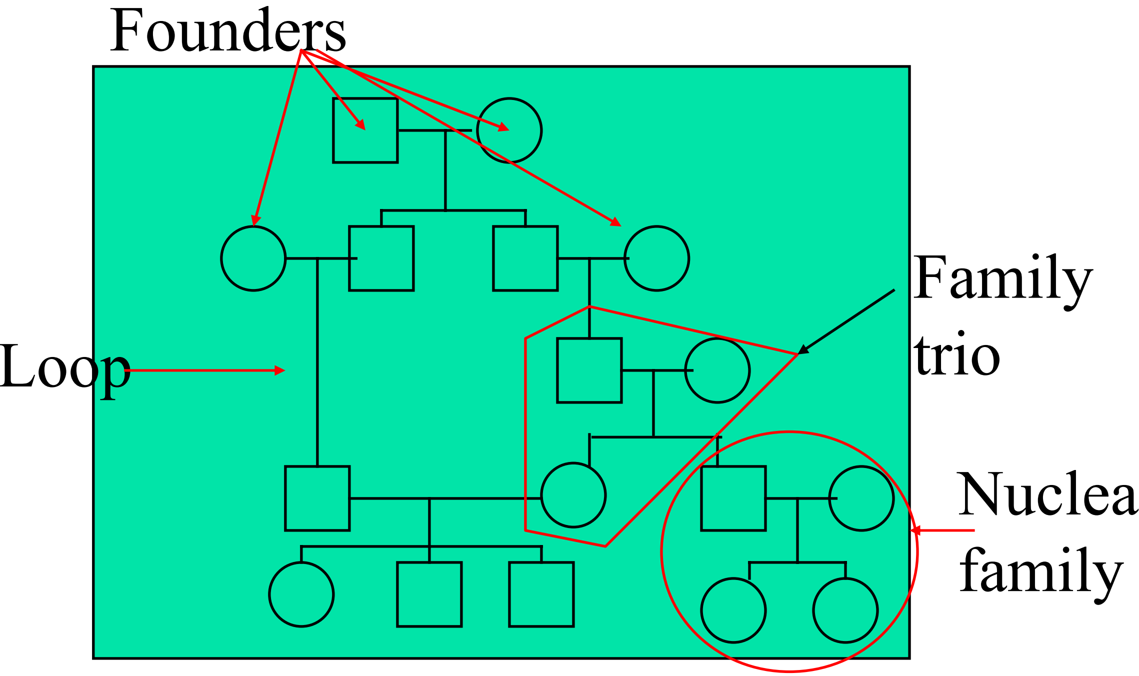
- **Pedigree, nuclear family, founder**





# Pedigree

- **Pedigree, nuclear family, founder**

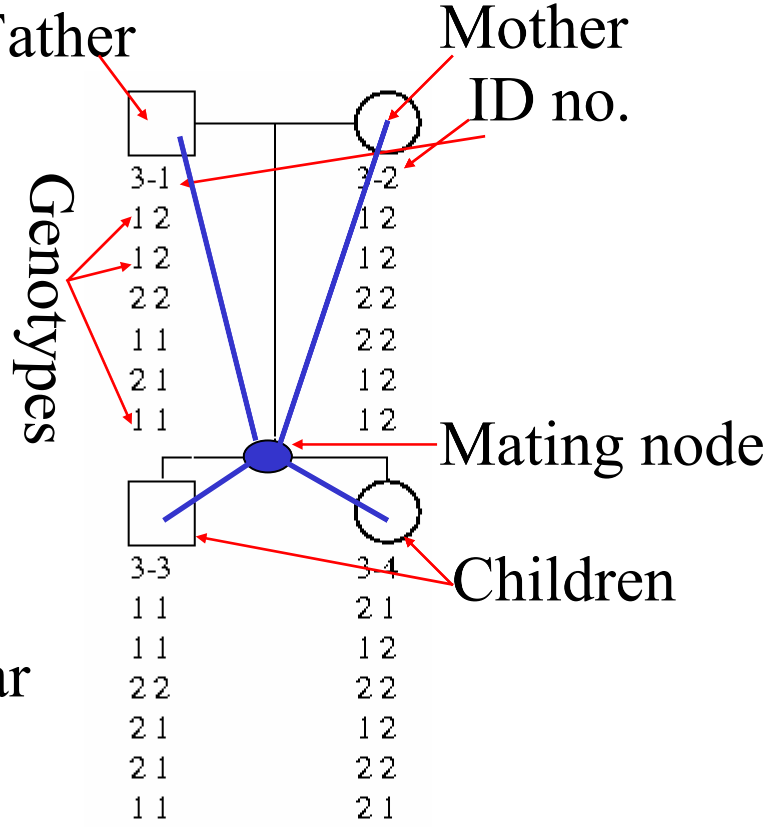


Founders

Loop

Family trio

Nuclear family



Father

Mother

ID no.

Genotypes

Mating node

Children

# Haplotyping from Genotypes: The Problem & Methods

- **Problem:**
  - **Input: genotype data (possibly with missing alleles).**
  - **Output: haplotypes.**
- **Input data:**
  - **Data with pedigree (dependent individuals).**
  - **Data without pedigree info (independent individuals).**

- **Statistical methods**
  - **Find the most likely haplotypes based on genotype data.**
  - **Adv: solid theoretical bases**
  - **Disadv: computation intensive**

- **Rule-based (*i.e.* combinatorial) methods**
  - **Define rules/objective functions based on some plausible assumptions and find haplotypes consistent with the rules or optimizing the obj. fun.**
  - **Adv: usually simple thus very fast**
  - **Disadv: no numerical assessment of the reliability of the results**

# Motivations

- Haplotype is more biologically meaningful than genotype since haplotypes are directly inherited from parents. Haplotype data is more informative in the studies of association between diseases and genes, and human history.

- The human genome project gives us the consensus genotype sequence of humans, but in order to understand the genetic effects on many complex diseases such as cancers, diabetes, osteoporoses, genetic variations are more important, which is best refecledt in haplotypes.

- Current experimental techniques collect genotype data. Computational methods deriving haplotypes from genotypes are highly demanded.

- The ongoing international HapMap project.

# Motivations (cont'd)

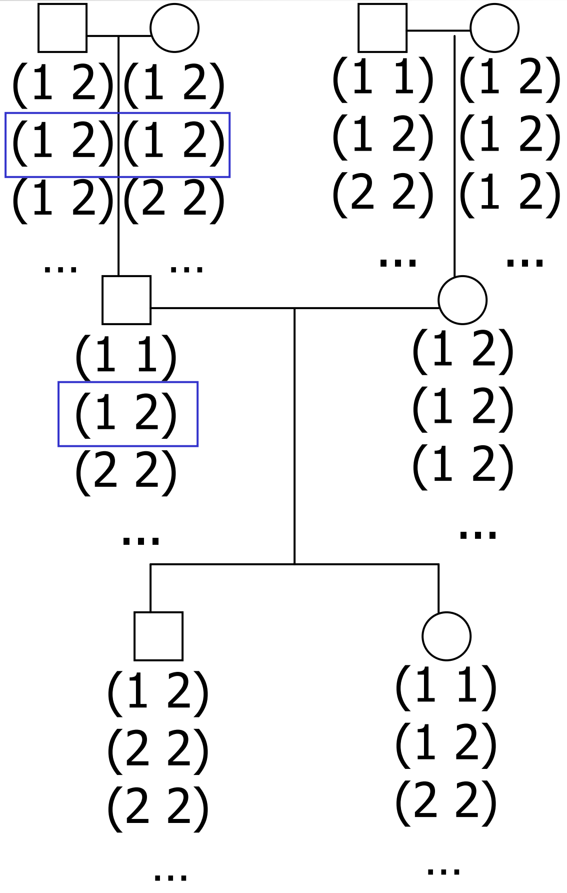
- It is generally believed that with parents/pedigree information, we could get more accurate haplotype and frequency estimations than from data without such information (*i.e.* population data).
- Family-based association studies have been widely used. We would expect more family-based gene mapping methods that assume accurate haplotype information.
- Not only computation intensive, model-based statistical methods may use assumptions that may not hold in real datasets.

# MRHC Problem

Find a minimum recombinant haplotype configuration from a given pedigree with genotype data.

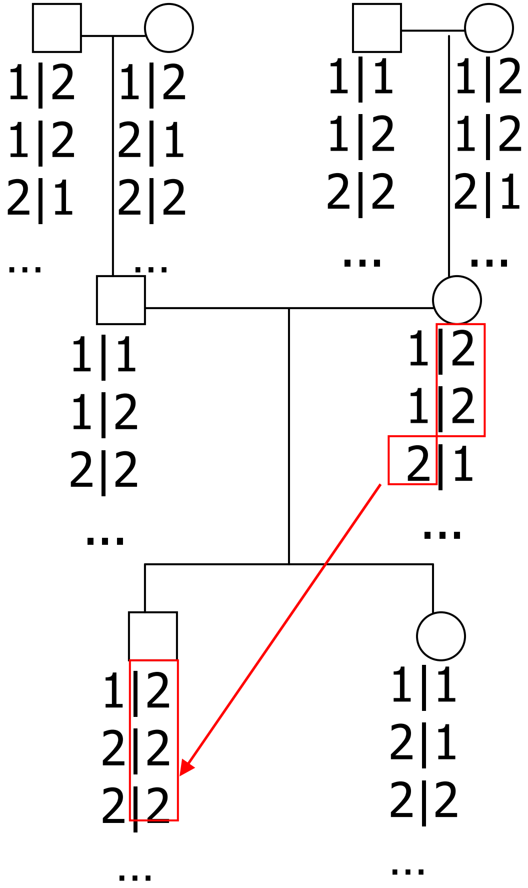
Assumptions:

- Mendelian law (no mutations)
- Recombination events are **rare**



Input (phase unknown)

# The MRHC Problem



Output (phase known)

- PS: parental source of the two alleles at the locus (*i.e.* phase)
- Haplotype configuration = assignment of PS values at each locus of every individual.

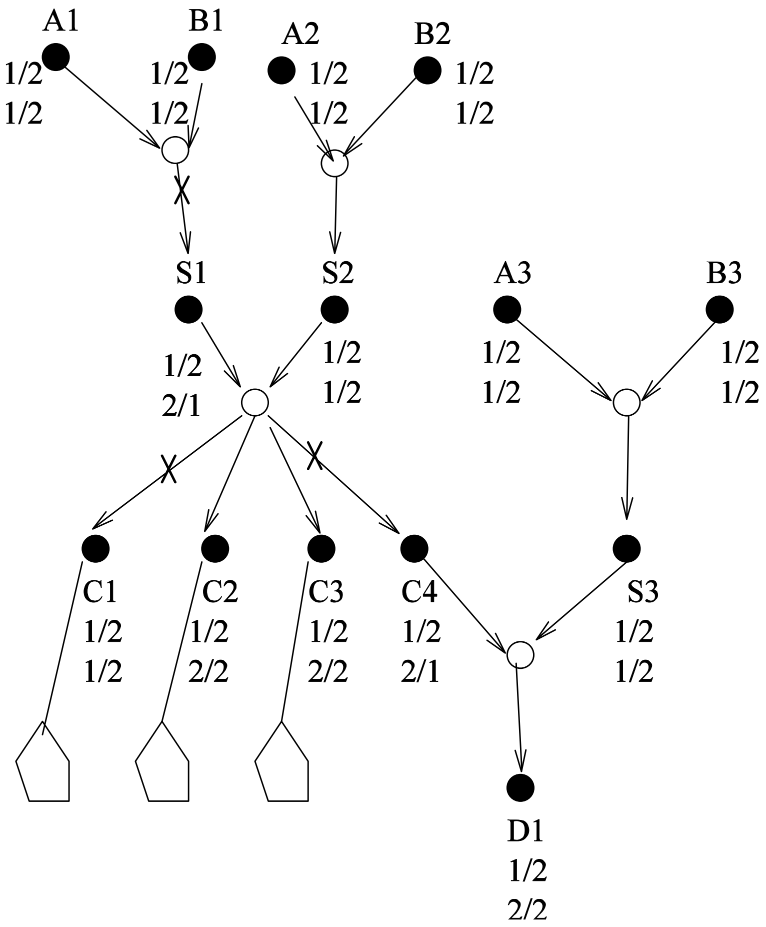
**PS = 1 (because the allele with the smaller index is maternal)**

# Previous Results

- Genotype elimination (O’Connell & Weeks’99).
  - Can only find haplotype configurations requiring no recombinant in the pedigree, exhaustive elimination.
- Genetic algorithm (Tapadar *et al.*’00).
  - Still time consuming, needs many iterations before convergence.
- MRH (Qian & Beckmann’02).
  - Six step rule-based algorithm.
  - Locus by locus at every step, extremely slow for biallelic (*e.g.* SNP) markers.

# Thm. MRHC is NP-hard.

- Idea: Reduction from a variant of set cover.
- First complexity result concerning the problem.
- Remains hard when there are only two loci.
- Remains hard when no loops in a pedigree.



# An Exact Algorithm for 0-Recombinant Data Based on Resolution of Constraints

- Assumptions:
  - Zero recombinants.
  - No missing alleles, no errors.

- Idea: finding all feasible (*i.e.* 0-recombinant) haplotype configurations is equivalent to reducing the degree of freedom in PS assignment.

- Steps:
  - formulate all the constraints, as linear equations over GF(2)
  - solve the equations by Gaussian elimination
  - enumerate all feasible haplotype configurations

# Four Levels of Constraints

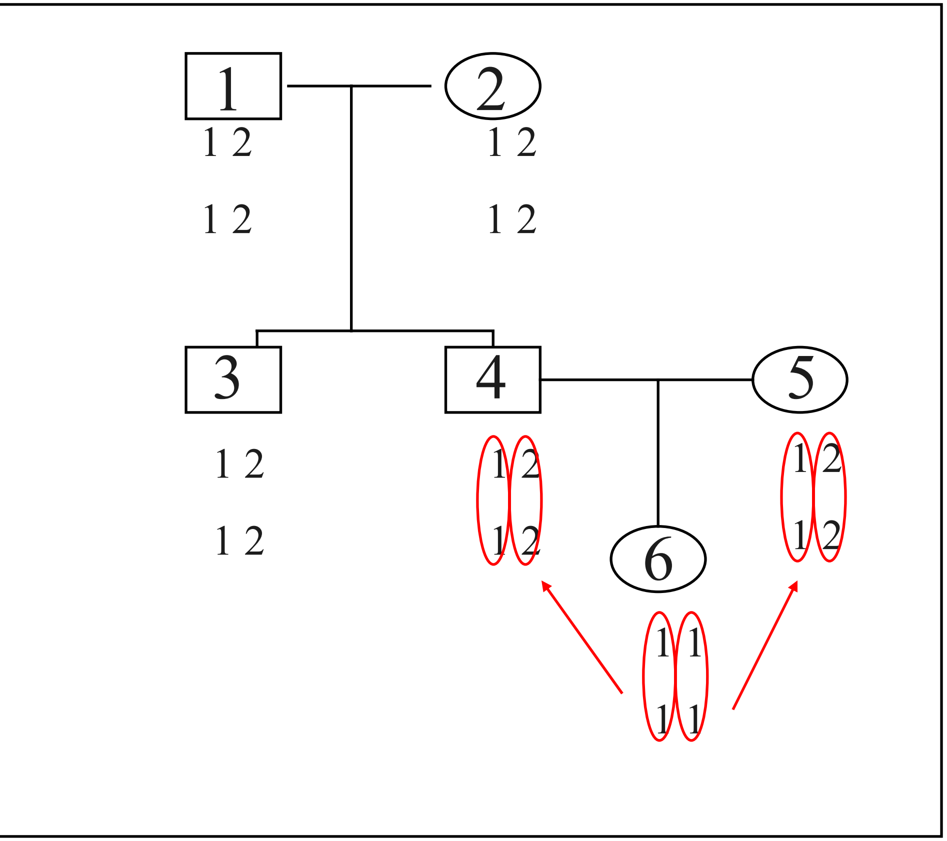
Based on Mendelian law
(for single locus) :

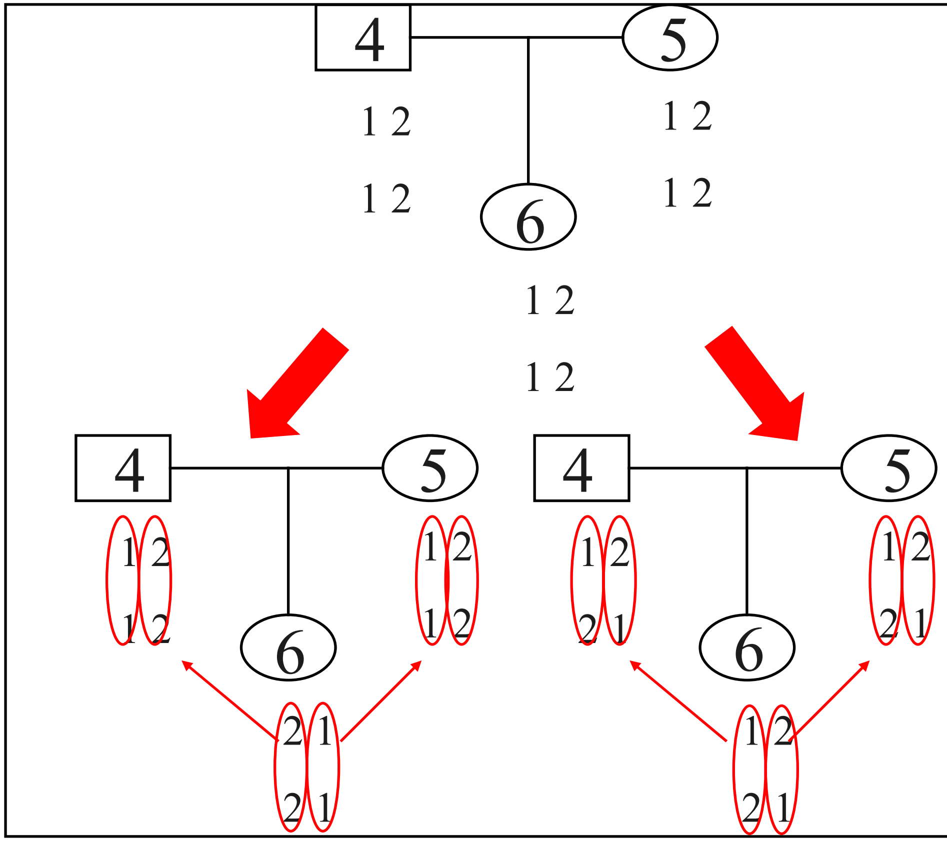
- Level 1: GS (grantparental source) constraint
- Level 2: PS constraint

Based on 0-recombinant assumption
(for a pair of loci):

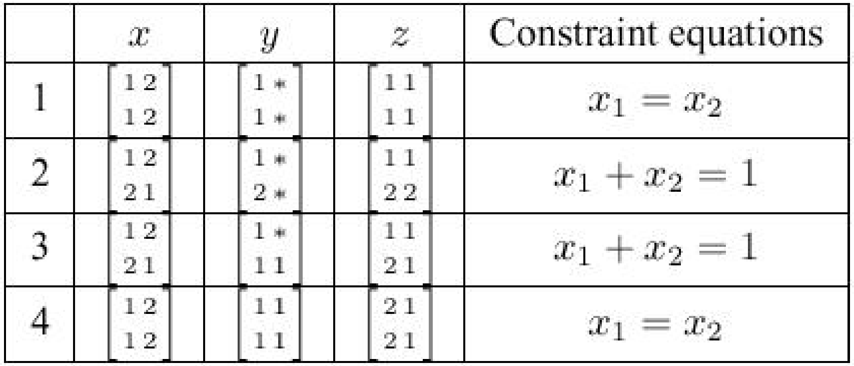
- Level 3: Haplotype constraint
- Level 4: Grouping constraint

# Level 3 and Level 4 Constraints



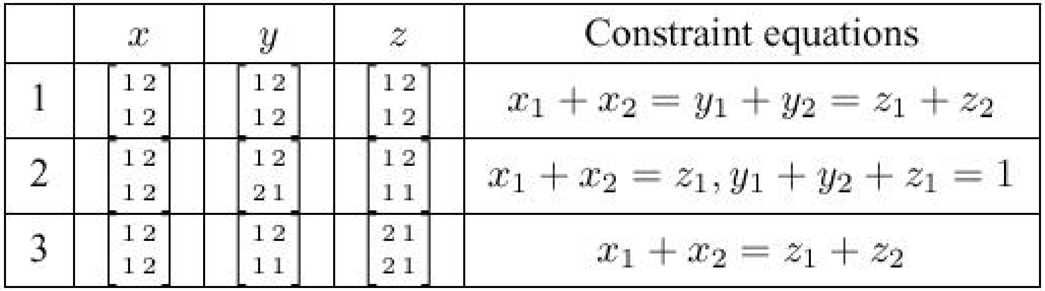


# Level 3 and level 4 Constraints



| | $x$ | $y$ | $z$ | Constraint equations |
|---|---|---|---|---|
| 1 | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&*\\1&*\end{bmatrix}$ | $\begin{bmatrix}1&1\\1&1\end{bmatrix}$ | $x_1 = x_2$ |
| 2 | $\begin{bmatrix}1&2\\2&1\end{bmatrix}$ | $\begin{bmatrix}1&*\\2&*\end{bmatrix}$ | $\begin{bmatrix}1&1\\2&2\end{bmatrix}$ | $x_1 + x_2 = 1$ |
| 3 | $\begin{bmatrix}1&2\\2&1\end{bmatrix}$ | $\begin{bmatrix}1&*\\1&1\end{bmatrix}$ | $\begin{bmatrix}1&1\\2&1\end{bmatrix}$ | $x_1 + x_2 = 1$ |
| 4 | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&1\\1&1\end{bmatrix}$ | $\begin{bmatrix}2&1\\2&1\end{bmatrix}$ | $x_1 = x_2$ |

Table 1: The possible level 3 constraints.



| | $x$ | $y$ | $z$ | Constraint equations |
|---|---|---|---|---|
| 1 | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $x_1 + x_2 = y_1 + y_2 = z_1 + z_2$ |
| 2 | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&2\\2&1\end{bmatrix}$ | $\begin{bmatrix}1&2\\1&1\end{bmatrix}$ | $x_1 + x_2 = z_1, y_1 + y_2 + z_1 = 1$ |
| 3 | $\begin{bmatrix}1&2\\1&2\end{bmatrix}$ | $\begin{bmatrix}1&2\\1&1\end{bmatrix}$ | $\begin{bmatrix}2&1\\2&1\end{bmatrix}$ | $x_1 + x_2 = z_1 + z_2$ |

Table 2: The possible level 4 constraints.

Note: The variables represent PS values and the equations are over GF(2) (in fact, addition mod 2).

# Constraint-Based Algorithm

**Thm.** Every solution consistent with the constraint equations is a feasible solution and vice versa.

We can adapt the classical Gaussian elimination algorithm to find all consistent solutions in O($n^3m^3$) time.

Previously, only an exponential time algorithm is known due to O'Connell and Weeks (1999).

The algorithm is useful for solving 0-recombinant data and may serve as a subroutine in a general haplotyping algorithm.

# Block-Extension Algorithm

Iterative, heuristic, five steps. Rules are derived from Mendelian law, MR principle, block concept and some greedy ideas based on the following observations:

- Block structures are common in haplotypes.
- Double recombination events are rare.
- Common haplotype blocks shared in siblings.
- …

# Steps in the BE algorithm

- Missing allele imputation by the Mendelian Law of inheritance and allele frequency
- PS assignment by Mendelian Law
  - Locus by locus, member by member, in a top-down scan
- Greedy assignment of PS
  - Bottom-up, infer PS value from PS of adjacent loci.
- Block-Extension
  - Iteratively extend the longest block to the same region of other members.
- Finishing the gaps between blocks by enumeration.

# Analysis of the BE Algorithm

- Advantage:
  - Simple and efficient.
  - Accurate when the number of recombination events is small.
- Disadvantage:
  - Potential errors in steps 3 and 4. Accuracy could decrease with the increase of the number of recombination events.

# More Exact Algorithms

- Locus-based dynamic programming algorithm
  - Linear time in the number of the members
  - Applicable to only tree pedigrees
- Member-based dynamic programming algorithm
  - Linear time in the number of the loci
  - Applicable to general pedigrees with small size

- Integer linear programming (ILP) with branch-and-bound
  - Combines missing data imputation and haplotype inference together.
  - It also implicitly checks Mendelian consistency for pedigree genotype data with missing alleles, which is also an NPC problem.
  - Effective for practical size problems, regardless of the pedigree structure

# ILP for MRHC with Missing Data

1. Alleles are represented as binary variables.
2. Genotype info and the Mendelian law of inheritance are enforced by linear constraints.
3. The objective of minimizing the total number of recombinants is encoded as a linear function of the variables.
4. Effective preprocessing of constraints by taking advantage of special properties in our ILP formulation to reduce the number of variables.
5. Branch-and-bound strategy to find solutions. The branch step guided by a partial order relationship (and some other special relationships) identified during the preprocessing step.
6. Non-trivial bounds are estimated to prune the search tree.
7. A maximum likelihood approach is used to select the best haplotype configuration from multiple optimal solutions.

# Formulation: variables

- Possible alleles (totally $t_j$) at marker locus j: $M_j = \{m_1^j, \ldots, m_{t_j}^j\}$

- Define 2$t_j$ (*f* and *m*) vars and 2 *g* vars for each paternal allele and maternal allele at locus *j* for individual *i*

$$f_{i,k}^j, m_{i,k}^j \ (1 \le k \le t_j) \quad g_{i,1}^j, g_{i,2}^j$$

- Var $f_k$ (or $m_k$)=1 iff the allele is $m_k$. Var $g_1$ = 0 (or 1) iff paternal allele is copied from father's paternal (or maternal) allele. Var $g_2$ defined similarly.

- Define *r* vars:

$$r_{i,1}^j, r_{i,2}^j \ (1 \le j \le m-1)$$

$$r_{i,1}^j = 1 \text{ iff } g_{i,1}^j \ne g_{i,1}^{j+1}$$

# Formulation

- Objective function:

$$\sum_{\text{Non-Founders}} \sum_{j=1}^{m-1} (r_{i,1}^{j} + r_{i,2}^{j})$$

Subject to

Genotype constraints: (0 means missing allele)

$$\{0,0\} \Rightarrow \{\sum_{k=1}^{t_j} f_{i,k}^{j} = 1, \sum_{k=1}^{t_j} m_{i,k}^{j} = 1\}$$

$$\{m_r^j, 0\} \Rightarrow \{f_{i,r}^{j} + m_{i,r}^{j} \geq 1\}$$

$$\{m_r^j, m_r^j\} \Rightarrow \{f_{i,r}^{j} = m_{i,r}^{j} = 1\}$$

$$\{m_r^j, m_s^j\} \Rightarrow \{f_{i,r}^{j} + f_{i,s}^{j} = m_{i,r}^{j} + m_{i,s}^{j} = f_{i,r}^{j} + m_{i,r}^{j} = f_{i,s}^{j} + m_{i,s}^{j} = 1\}$$

# Formulation

- Mendelian law of inheritance constraints (for a child *i* and its father *f*):

$$f_{i,k}^{j} - f_{f,k}^{j} - g_{i,1}^{j} \leq 0$$

$$f_{i,k}^{j} - m_{f,k}^{j} + g_{i,1}^{j} \leq 1$$

- Constraints for *r* vars:

$$r_{i,l}^{j} - g_{i,l}^{j} - g_{i,l}^{j+1} \leq 0$$

$$r_{i,l}^{j} + g_{i,l}^{j} + g_{i,l}^{j+1} \leq 2$$

$$-r_{i,l}^{j} + g_{i,l}^{j} - g_{i,l}^{j+1} \leq 0$$

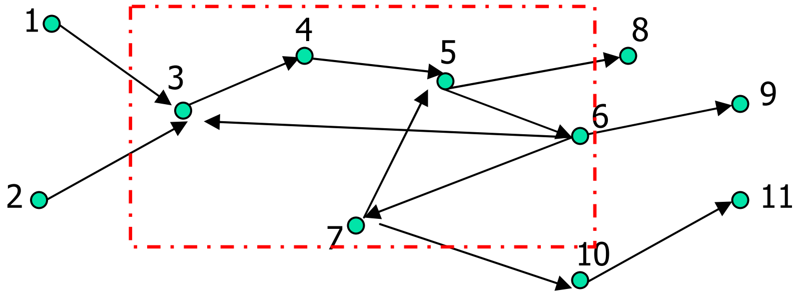
$$-r_{i,l}^{j} - g_{i,l}^{j} + g_{i,l}^{j+1} \leq 0$$

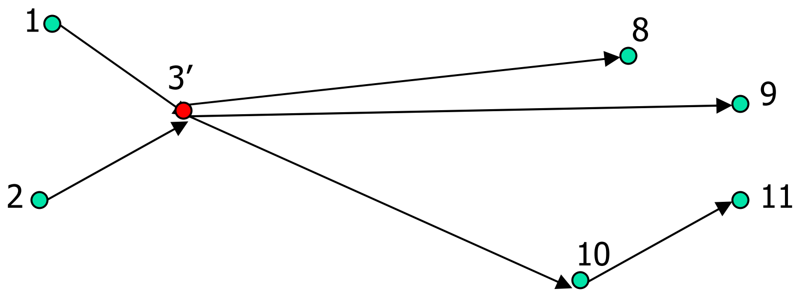
# A Partial Order Relationship

Denote: $$y^{\gamma} = \begin{cases} y & \gamma = 1 \\ 1 - y & \gamma = 0 \end{cases}$$

Inequalities with 2 variables:

$$y_i^{\alpha} \le y_j^{\beta}$$





# Forced Variables

- Rule 1: $y^0, y^1 \in S \Rightarrow \text{Inconsistency}$

- Rule 2: $(y_i^\alpha \le y_j^\beta) \wedge (y_i^\alpha \le y_j^{1-\beta}) \Rightarrow y_i^\alpha = 0$

  $(y_i^\alpha \le y_j^\beta) \wedge (y_i^{1-\alpha} \le y_j^\beta) \Rightarrow y_j^\beta = 1$

- Rule 3: $y_i^\alpha \le y_i^{1-\alpha} \Rightarrow y_i^\alpha = 0$

# Lower and Upper Bounds

- Lower bounds
  - Linear relaxation.
  - Sum of minimum number of recombinants in each nuclear family.
  - Effective for data with a large number of recombinants.

- Upper bound
  - Obtained by the Block-Extension algorithm.
  - Effective for data with a small number of recombinants.

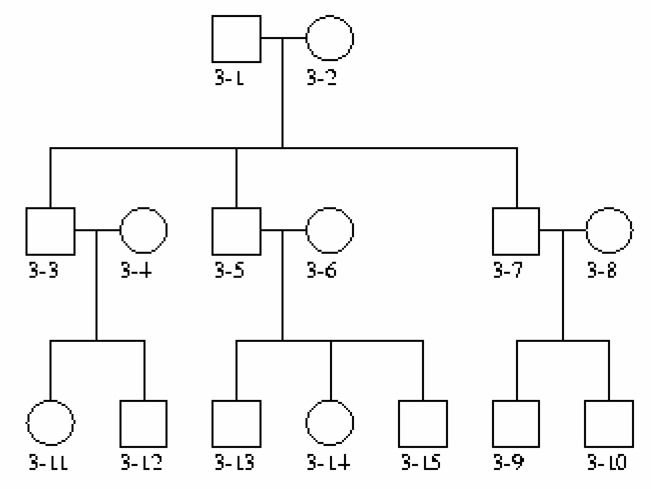
# ILP

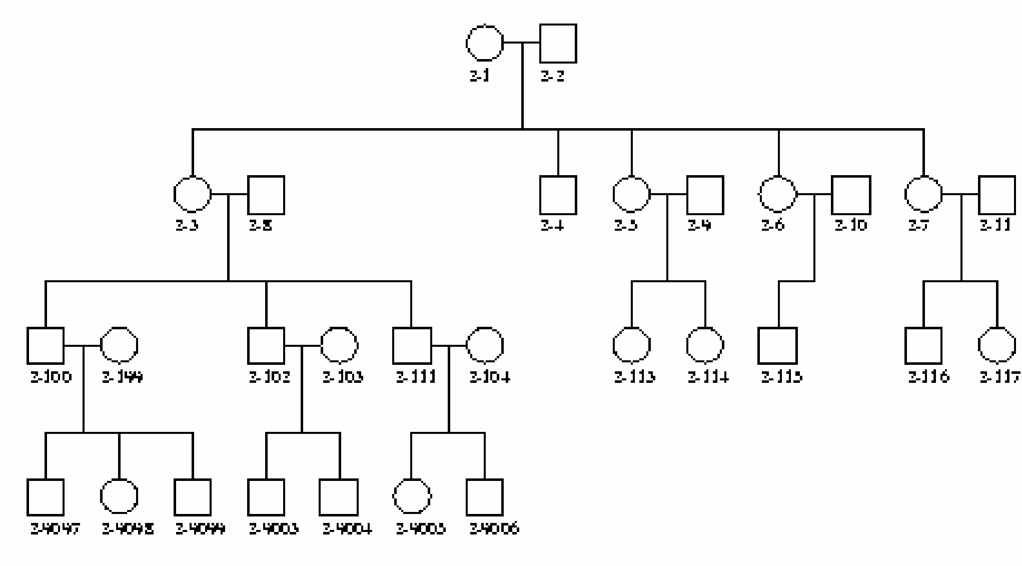
- Practical in terms of time efficiency
- Could find all possible optimal solutions
- Very effective in terms of missing allele imputation.

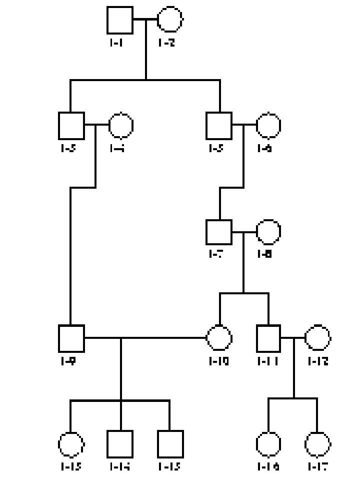
# Simulation Studies

- The algorithms have been implemented in a program called PedPhase in C++.
- Simulated data were generated to compare our algorithms, and with MRH (Qian&Beckmann'02)
- Three different pedigree structures.
- Multiallelic and biallelic data.
- Numbers of loci: 10, 25 and 50.
- Number of recombinants: 0-4.
- 100 runs per data set.

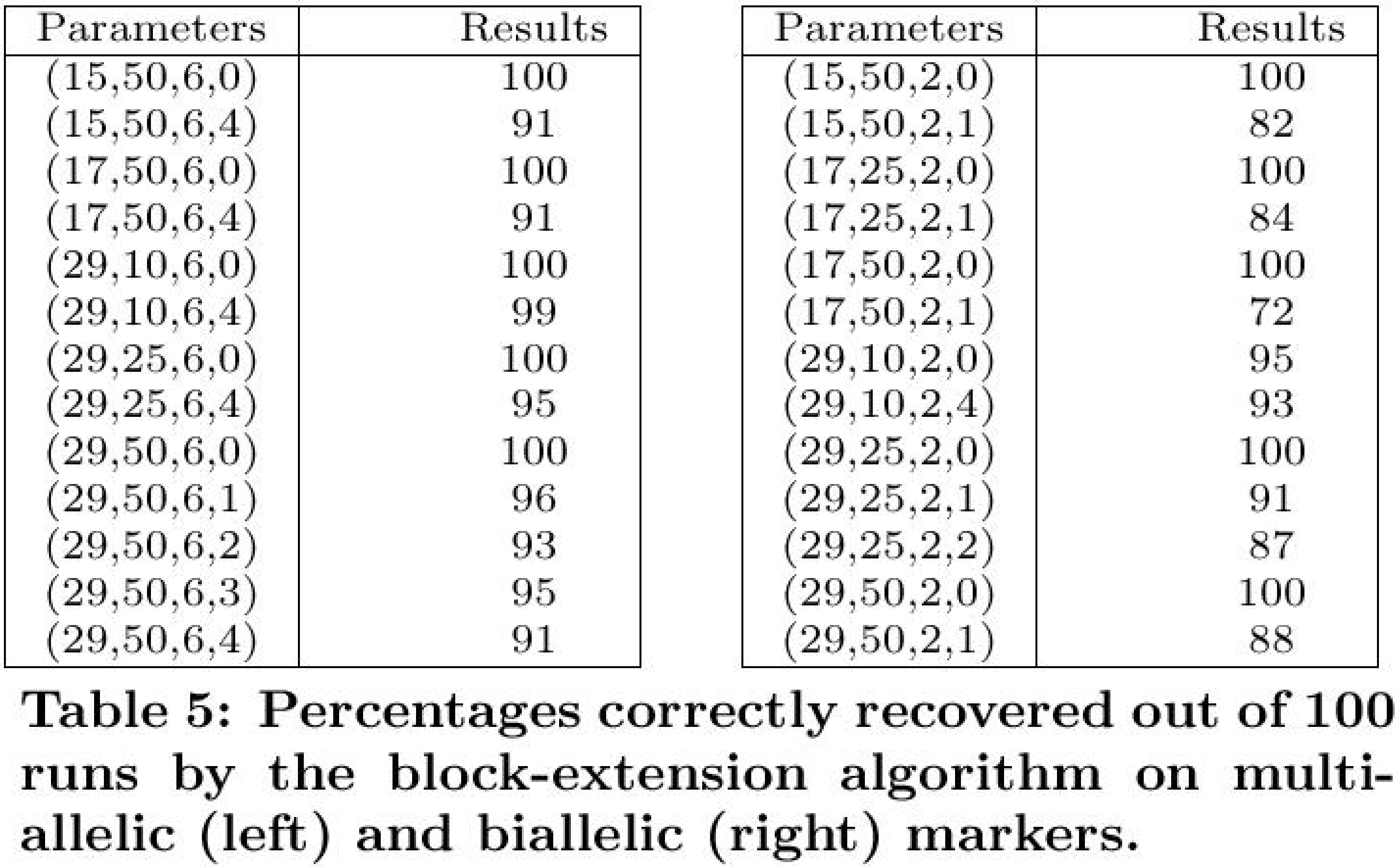
# Pedigree Structures







# Accuracy Results of Algrotihm Block-Extension



| Parameters | Results |
|---|---|
| (15,50,6,0) | 100 |
| (15,50,6,4) | 91 |
| (17,50,6,0) | 100 |
| (17,50,6,4) | 91 |
| (29,10,6,0) | 100 |
| (29,10,6,4) | 99 |
| (29,25,6,0) | 100 |
| (29,25,6,4) | 95 |
| (29,50,6,0) | 100 |
| (29,50,6,1) | 96 |
| (29,50,6,2) | 93 |
| (29,50,6,3) | 95 |
| (29,50,6,4) | 91 |

| Parameters | Results |
|---|---|
| (15,50,2,0) | 100 |
| (15,50,2,1) | 82 |
| (17,25,2,0) | 100 |
| (17,25,2,1) | 84 |
| (17,50,2,0) | 100 |
| (17,50,2,1) | 72 |
| (29,10,2,0) | 95 |
| (29,10,2,4) | 93 |
| (29,25,2,0) | 100 |
| (29,25,2,1) | 91 |
| (29,25,2,2) | 87 |
| (29,50,2,0) | 100 |
| (29,50,2,1) | 88 |

**Table 5: Percentages correctly recovered out of 100 runs by the block-extension algorithm on multi-allelic (left) and biallelic (right) markers.**

# Efficiency Results

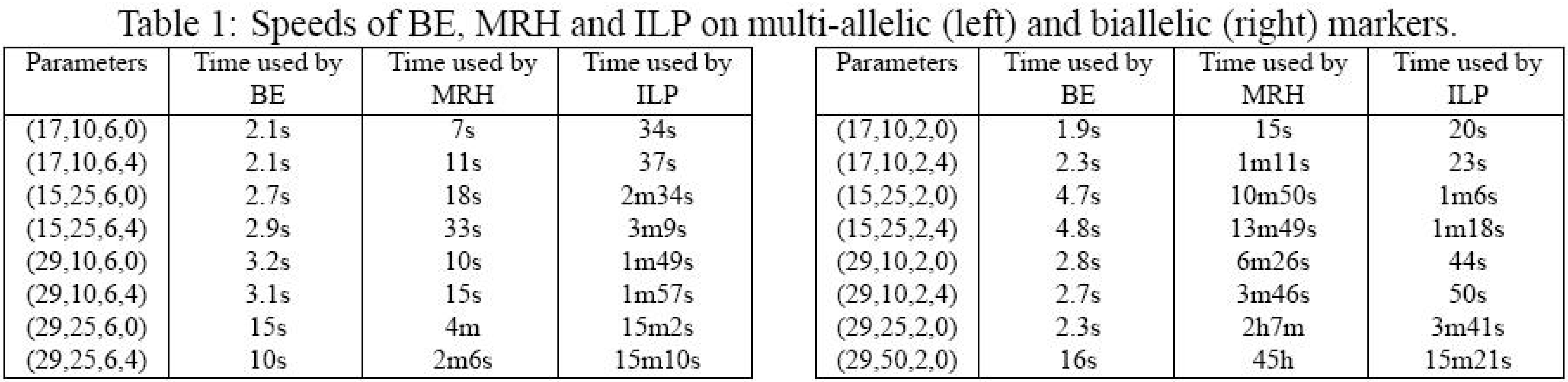


Table 1: Speeds of BE, MRH and ILP on multi-allelic (left) and biallelic (right) markers.

| Parameters | Time used by BE | Time used by MRH | Time used by ILP |
|---|---|---|---|
| (17,10,6,0) | 2.1s | 7s | 34s |
| (17,10,6,4) | 2.1s | 11s | 37s |
| (15,25,6,0) | 2.7s | 18s | 2m34s |
| (15,25,6,4) | 2.9s | 33s | 3m9s |
| (29,10,6,0) | 3.2s | 10s | 1m49s |
| (29,10,6,4) | 3.1s | 15s | 1m57s |
| (29,25,6,0) | 15s | 4m | 15m2s |
| (29,25,6,4) | 10s | 2m6s | 15m10s |

| Parameters | Time used by BE | Time used by MRH | Time used by ILP |
|---|---|---|---|
| (17,10,2,0) | 1.9s | 15s | 20s |
| (17,10,2,4) | 2.3s | 1m11s | 23s |
| (15,25,2,0) | 4.7s | 10m50s | 1m6s |
| (15,25,2,4) | 4.8s | 13m49s | 1m18s |
| (29,10,2,0) | 2.8s | 6m26s | 44s |
| (29,10,2,4) | 2.7s | 3m46s | 50s |
| (29,25,2,0) | 2.3s | 2h7m | 3m41s |
| (29,50,2,0) | 16s | 45h | 15m21s |

# More Results on ILP

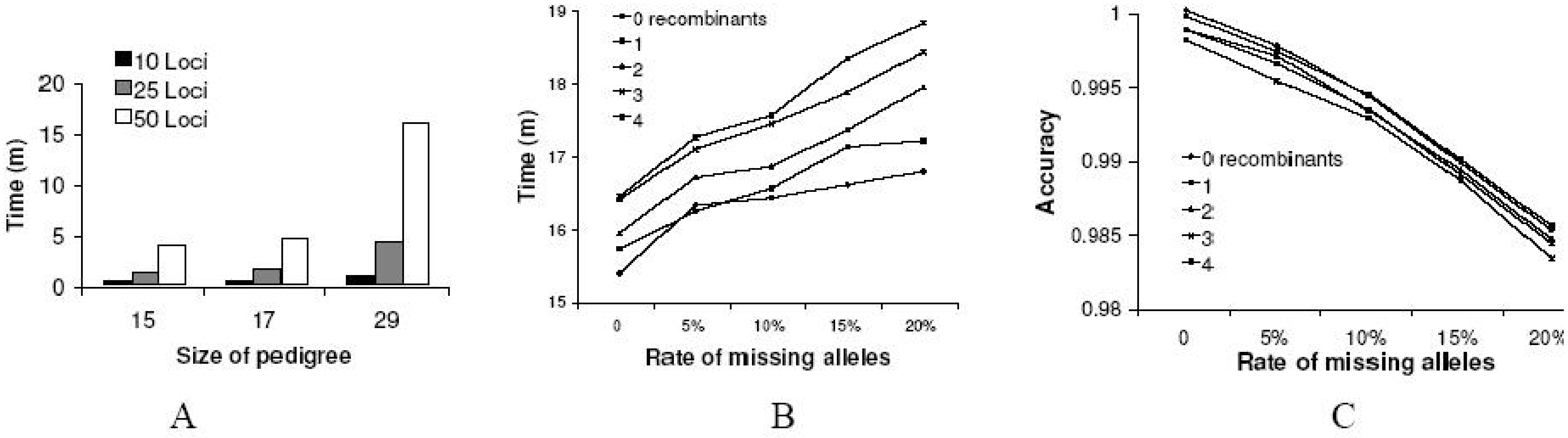
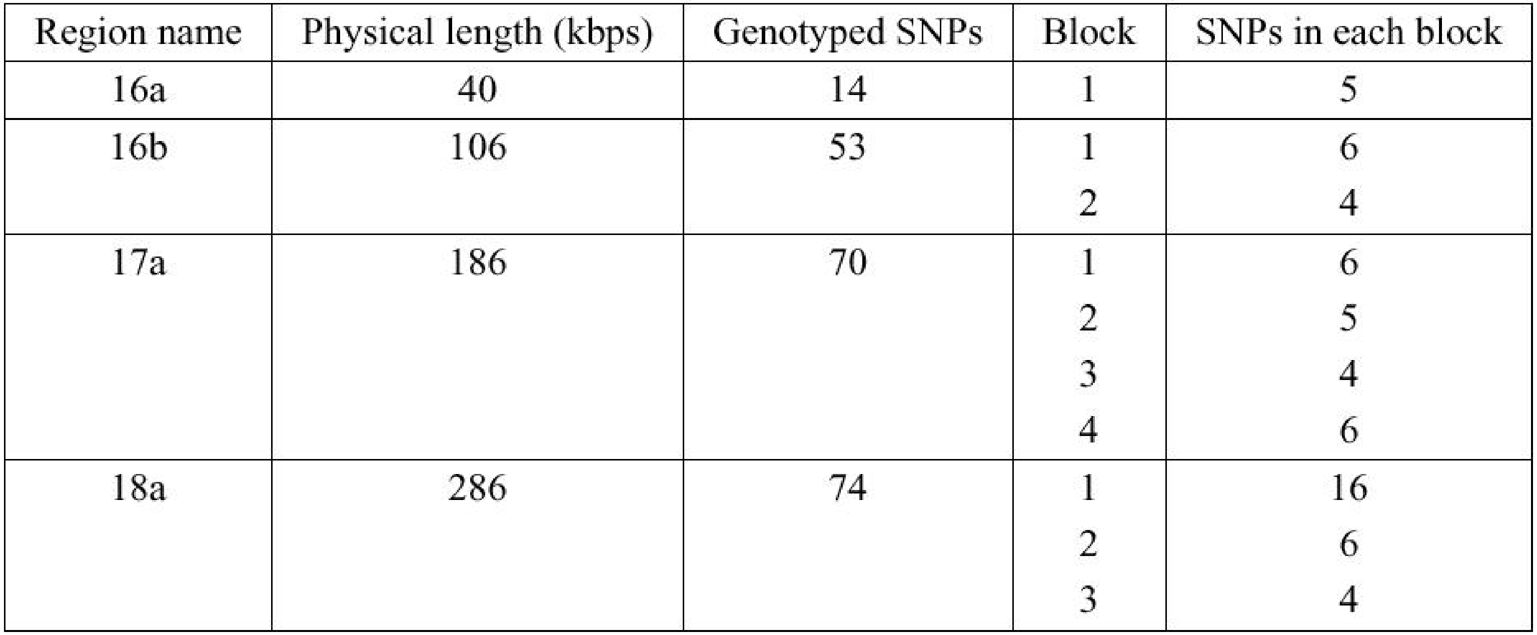


Figure 4: Some simulation results on ILP. A. Effect of problem size on speed. B. Effect of number of recombinants and rate of missing alleles on speed. C. Effect of number of recombinants and rate of missing alleles on accuracy.

# Real Data Analysis

- Data set (Gabriel *et al.*'02)
  - 93 members, 12 pedigrees (each with 7-8 members);
  - chromosome 3, 4 regions, each region 1-4 blocks.



| Region name | Physical length (kbps) | Genotyped SNPs | Block | SNPs in each block |
|---|---|---|---|---|
| 16a | 40 | 14 | 1 | 5 |
| 16b | 106 | 53 | 1 | 6 |
| | | | 2 | 4 |
| 17a | 186 | 70 | 1 | 6 |
| | | | 2 | 5 |
| | | | 3 | 4 |
| | | | 4 | 6 |
| 18a | 286 | 74 | 1 | 16 |
| | | | 2 | 6 |
| | | | 3 | 4 |

Table 8: The regions and blocks on chromosome 3.

# Reconstruction of Common Haplotypes and Estimation of Their Frequencies

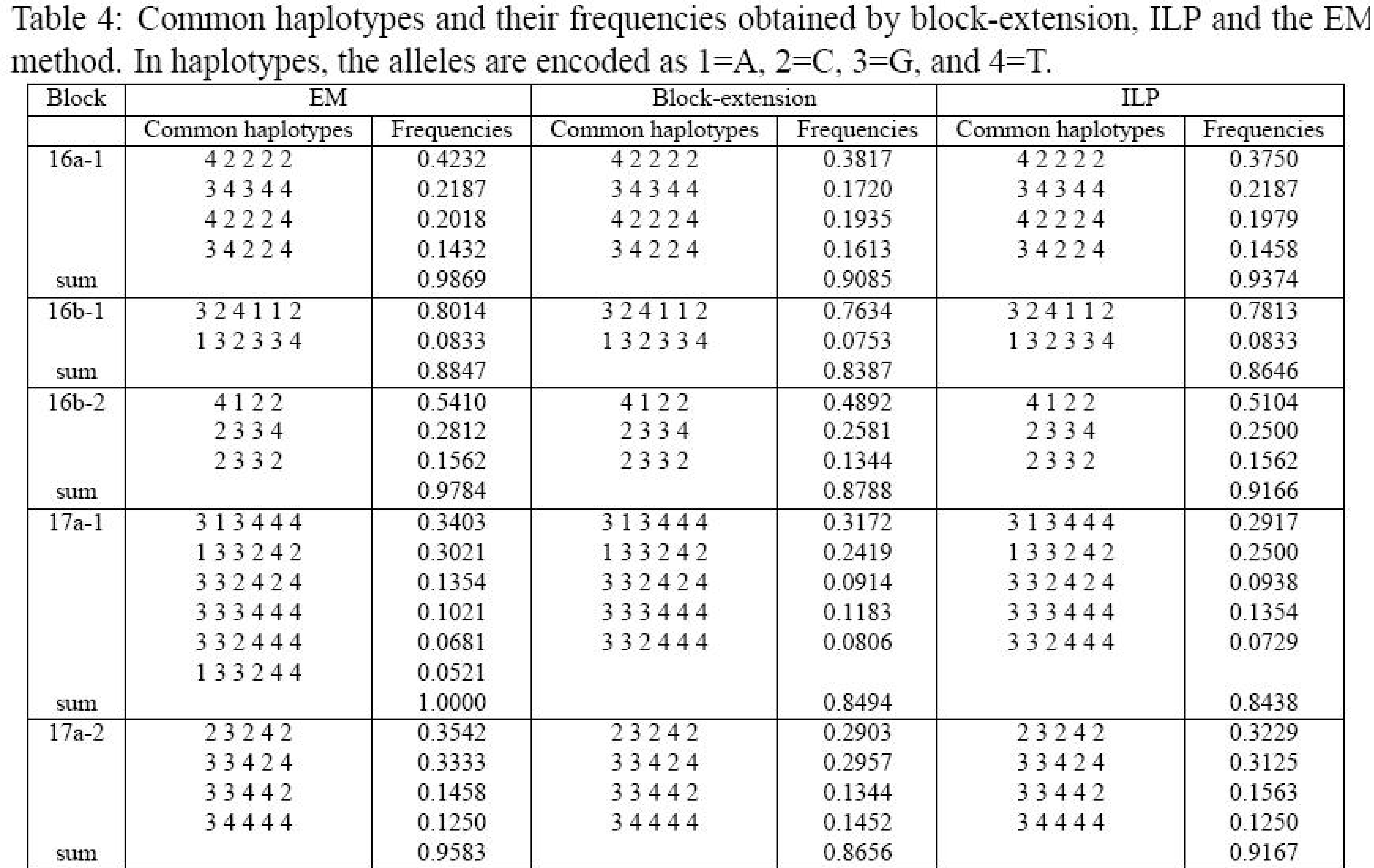


Table 4: Common haplotypes and their frequencies obtained by block-extension, ILP and the EM method. In haplotypes, the alleles are encoded as 1=A, 2=C, 3=G, and 4=T.

| Block | EM | | Block-extension | | ILP | |
|---|---|---|---|---|---|---|
| | Common haplotypes | Frequencies | Common haplotypes | Frequencies | Common haplotypes | Frequencies |
| 16a-1 | 4 2 2 2 2 | 0.4232 | 4 2 2 2 2 | 0.3817 | 4 2 2 2 2 | 0.3750 |
| | 3 4 3 4 4 | 0.2187 | 3 4 3 4 4 | 0.1720 | 3 4 3 4 4 | 0.2187 |
| | 4 2 2 2 4 | 0.2018 | 4 2 2 2 4 | 0.1935 | 4 2 2 2 4 | 0.1979 |
| | 3 4 2 2 4 | 0.1432 | 3 4 2 2 4 | 0.1613 | 3 4 2 2 4 | 0.1458 |
| sum | | 0.9869 | | 0.9085 | | 0.9374 |
| 16b-1 | 3 2 4 1 1 2 | 0.8014 | 3 2 4 1 1 2 | 0.7634 | 3 2 4 1 1 2 | 0.7813 |
| | 1 3 2 3 3 4 | 0.0833 | 1 3 2 3 3 4 | 0.0753 | 1 3 2 3 3 4 | 0.0833 |
| sum | | 0.8847 | | 0.8387 | | 0.8646 |
| 16b-2 | 4 1 2 2 | 0.5410 | 4 1 2 2 | 0.4892 | 4 1 2 2 | 0.5104 |
| | 2 3 3 4 | 0.2812 | 2 3 3 4 | 0.2581 | 2 3 3 4 | 0.2500 |
| | 2 3 3 2 | 0.1562 | 2 3 3 2 | 0.1344 | 2 3 3 2 | 0.1562 |
| sum | | 0.9784 | | 0.8788 | | 0.9166 |
| 17a-1 | 3 1 3 4 4 4 | 0.3403 | 3 1 3 4 4 4 | 0.3172 | 3 1 3 4 4 4 | 0.2917 |
| | 1 3 3 2 4 2 | 0.3021 | 1 3 3 2 4 2 | 0.2419 | 1 3 3 2 4 2 | 0.2500 |
| | 3 3 2 4 2 4 | 0.1354 | 3 3 2 4 2 4 | 0.0914 | 3 3 2 4 2 4 | 0.0938 |
| | 3 3 3 4 4 4 | 0.1021 | 3 3 3 4 4 4 | 0.1183 | 3 3 3 4 4 4 | 0.1354 |
| | 3 3 2 4 4 4 | 0.0681 | 3 3 2 4 4 4 | 0.0806 | 3 3 2 4 4 4 | 0.0729 |
| | 1 3 3 2 4 4 | 0.0521 | | | | |
| sum | | 1.0000 | | 0.8494 | | 0.8438 |
| 17a-2 | 2 3 2 4 2 | 0.3542 | 2 3 2 4 2 | 0.2903 | 2 3 2 4 2 | 0.3229 |
| | 3 3 4 2 4 | 0.3333 | 3 3 4 2 4 | 0.2957 | 3 3 4 2 4 | 0.3125 |
| | 3 3 4 4 2 | 0.1458 | 3 3 4 4 2 | 0.1344 | 3 3 4 4 2 | 0.1563 |
| | 3 4 4 4 4 | 0.1250 | 3 4 4 4 4 | 0.1452 | 3 4 4 4 4 | 0.1250 |
| sum | | 0.9583 | | 0.8656 | | 0.9167 |

# Results from ILP on the Whole Dataset

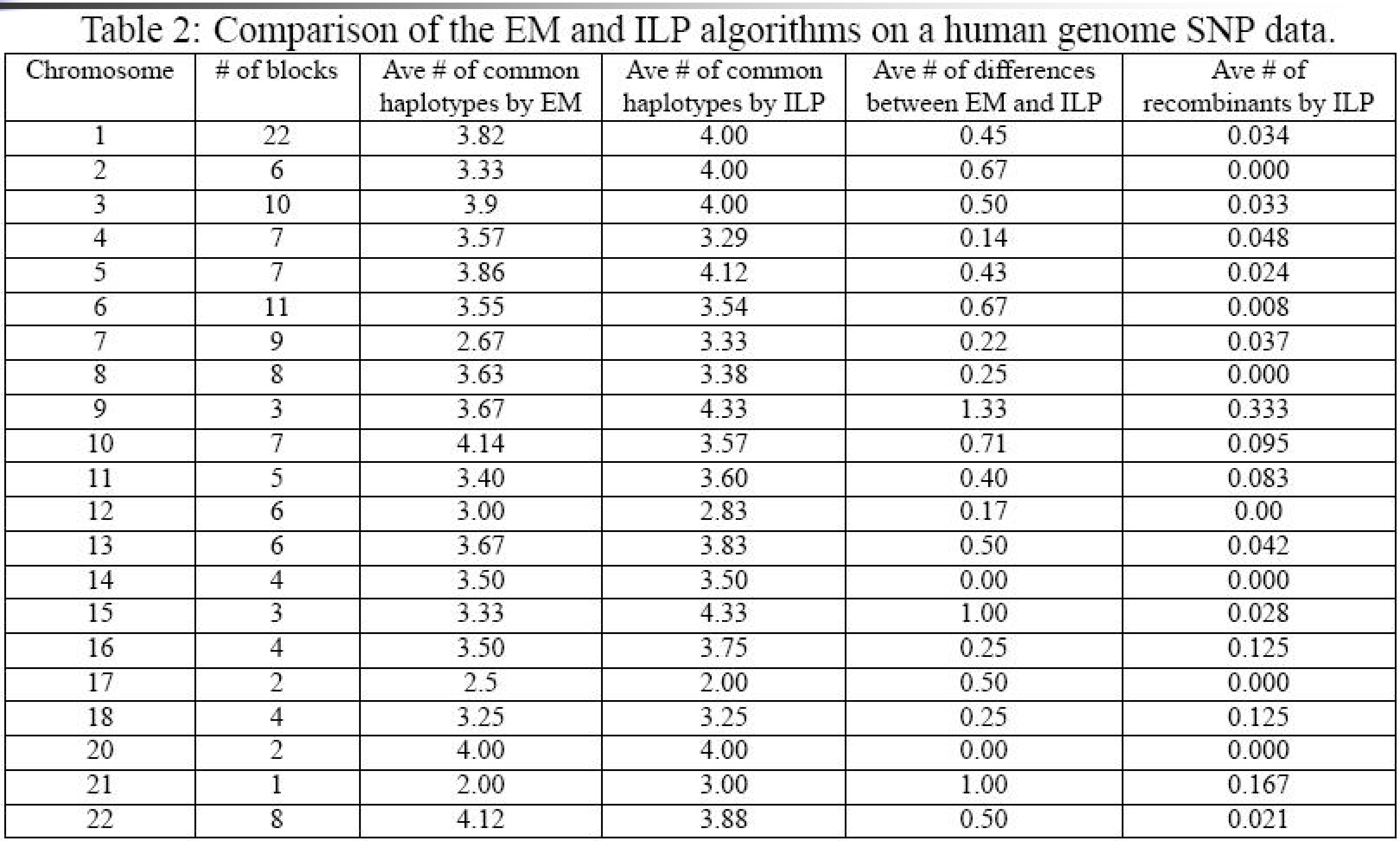


Table 2: Comparison of the EM and ILP algorithms on a human genome SNP data.

| Chromosome | # of blocks | Ave # of common haplotypes by EM | Ave # of common haplotypes by ILP | Ave # of differences between EM and ILP | Ave # of recombinants by ILP |
|---|---|---|---|---|---|
| 1 | 22 | 3.82 | 4.00 | 0.45 | 0.034 |
| 2 | 6 | 3.33 | 4.00 | 0.67 | 0.000 |
| 3 | 10 | 3.9 | 4.00 | 0.50 | 0.033 |
| 4 | 7 | 3.57 | 3.29 | 0.14 | 0.048 |
| 5 | 7 | 3.86 | 4.12 | 0.43 | 0.024 |
| 6 | 11 | 3.55 | 3.54 | 0.67 | 0.008 |
| 7 | 9 | 2.67 | 3.33 | 0.22 | 0.037 |
| 8 | 8 | 3.63 | 3.38 | 0.25 | 0.000 |
| 9 | 3 | 3.67 | 4.33 | 1.33 | 0.333 |
| 10 | 7 | 4.14 | 3.57 | 0.71 | 0.095 |
| 11 | 5 | 3.40 | 3.60 | 0.40 | 0.083 |
| 12 | 6 | 3.00 | 2.83 | 0.17 | 0.00 |
| 13 | 6 | 3.67 | 3.83 | 0.50 | 0.042 |
| 14 | 4 | 3.50 | 3.50 | 0.00 | 0.000 |
| 15 | 3 | 3.33 | 4.33 | 1.00 | 0.028 |
| 16 | 4 | 3.50 | 3.75 | 0.25 | 0.125 |
| 17 | 2 | 2.5 | 2.00 | 0.50 | 0.000 |
| 18 | 4 | 3.25 | 3.25 | 0.25 | 0.125 |
| 20 | 2 | 4.00 | 4.00 | 0.00 | 0.000 |
| 21 | 1 | 2.00 | 3.00 | 1.00 | 0.167 |
| 22 | 8 | 4.12 | 3.88 | 0.50 | 0.021 |

# Application of Haplotype Inference in Gene Association Mapping

- We have developed a new haplotype association mapping method based on density-based clustering for case-control data.
- The method regards haplotype segments as data points in a high dimensional space, and defines a new pairwise haplotype distance measure.
- Clusters are then identified by a density-based clustering algorithm.
- *Z*-scores based on the number of cases and controls in a cluster can be used as an indicator of the degree of association between a cluster and the disease under study.
- Results are very promising.
- But it needs haplotypes as input.

# An Application of Haplotype Inference

- Haplotypes are inferred by computational methods that we mentioned earlier.

- For example: a real data set that we analyzed consists of 385 nuclear families of size 4 (2 parents with 2 affected children).

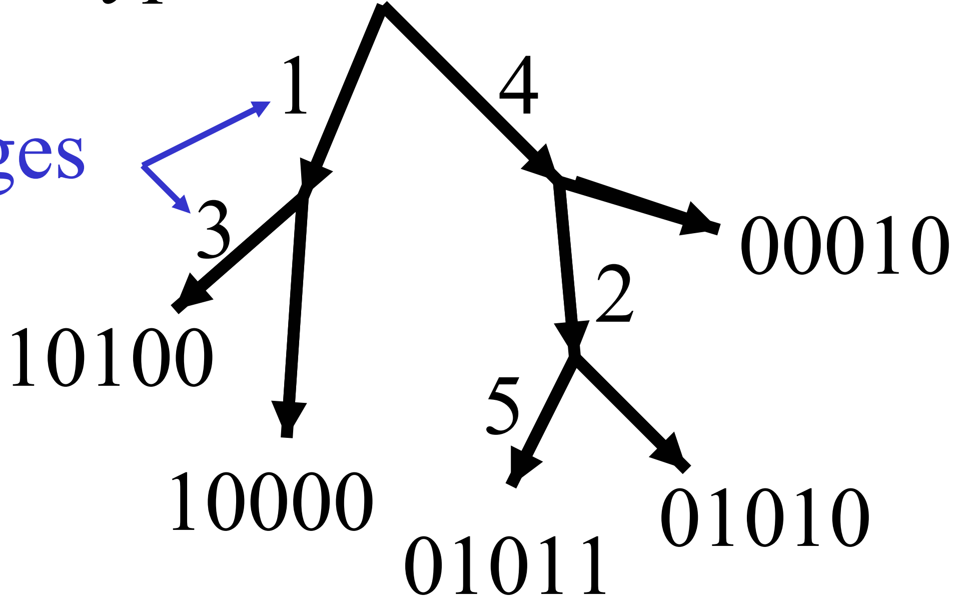
- We do haplotype inference first using our ILP algorithm. The haplotypes transmitted to (affected) children are treated as cases and un-transmitted haplotypes as controls. The haplotype association method was applied then.

# Inference of Haplotypes from Population Data: The Perfect Phylogeny Model

Loci 12345

Ancestral haplotype 00000

Locus **mutations** on edges



Each locus suffers from at most one mutation. No recombination!

Extant haplotypes at the leaves

# Perfect Phylogeny Haplotype (PPH)

Given a set/poplation of genotypes S, find an explanatory set of haplotypes that fits a perfect phylogeny.

Loci

S

|   | 1 | 2 |
|---|---|---|
| a | 2 | 2 |
| b | 0 | 2 |
| c | 1 | 0 |

Genotype matrix

The genotype coding:

|   |   |
|---|---|
| (11): 0 | homozygous |
| (22): 1 | homozygous |
| (12): 2 | heterozygous |

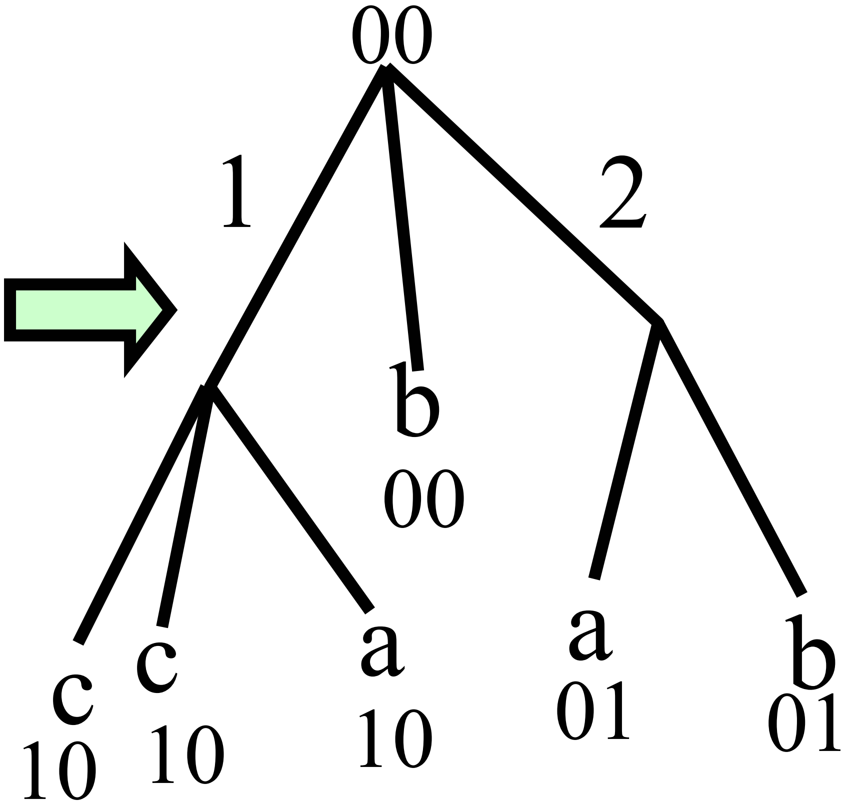
A haplotype pair explains a genotype if the merge of the haplotypes creates the genotype. E.g., merging haplotypes 001 and 100 results in genotype 202.

# Perfect Phylogeny Haplotype (PPH)

Given a set of genotypes S, find an explanatory set of haplotypes that fits a perfect phylogeny.

|   | 1 | 2 |
|---|---|---|
| a | 2 | 2 |
| b | 0 | 2 |
| c | 1 | 0 |

|   | 1 | 2 |
|---|---|---|
| a | 1 | 0 |
| a | 0 | 1 |
| b | 0 | 0 |
| b | 0 | 1 |
| c | 1 | 0 |
| c | 1 | 0 |



# Perfect Phylogeny Haplotype (PPH)

Given a set of genotypes S, find an explanatory set of haplotypes that fits a perfect phylogeny.

|   | 1 | 2 |
|---|---|---|
| a | 2 | 2 |
| b | 0 | 2 |
| c | 1 | 0 |

→

|   | 1 | 2 |
|---|---|---|
| a | 1 | 0 |
| a | 0 | 1 |
| b | 0 | 0 |
| b | 0 | 1 |
| c | 1 | 0 |
| c | 1 | 0 |

# An Alternative Haplotype Explanation

|   | 1 | 2 |
|---|---|---|
| a | 2 | 2 |
| b | 0 | 2 |
| c | 1 | 0 |

|   | 1 | 2 |
|---|---|---|
| a | 1 | 1 |
| a | 0 | 0 |
| b | 0 | 0 |
| b | 0 | 1 |
| c | 1 | 0 |
| c | 1 | 0 |

No perfect phylogeny exists for this explanation

# Efficient Solutions to the PPH Problem with n Individuals and m Loci

- Reduction to a graph realization problem (GPPH), based on Bixby-Wagner or Fushishige solution to graph realization (Gusfield'01).
- Reduction to graph realization, based on Tutte's graph realization method, in O(nm^2) time (Gusfield'02).
- Direct combinatorial approach in O(nm^2) time.

  Bafna *et al.*'03
- Eskin, Halperin and Karp'03: Specialize the Tutte solution to the PPH problem, in O(nm^2) time.

# Summary

- Li, J. and T. Jiang. Efficient Rule-Based Haplotyping Algorithm for Pedigree Data. *RECOMB'03*
  - NP-completeness proof for general pedigrees.
  - An efficient heuristic algorithm: block-extension.
  - An efficient exact algorithm for 0-recombinant data.
- Doi, K., J. Li and T. Jiang. Minimum Recombinant Haplotype Configuration on Tree Pedigrees. *WABI'03*
  - NP-completeness proof for loopless (or tree) pedigrees.
  - Two dynamic programming algorithms
- Li, J. and T. Jiang. An Exact Solution for Finding Minimum Recombinant Haplotype Configurations on Pedigrees with Missing Data by Integer Linear Programming. *RECOMB'04.*
- Li, J. and T. Jiang. Haplotype Association Mapping by Density-Based Clustering in Case-Control Studies. *RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotypes*, CMU, 2004.

# Future Work

- Incorporating mutations and errors into MRHC.
- Incorporating the likelihood of recombination into the objective function of ILP.
- Haplotype inference and missing allele imputation without pedigree information.
- Approximation algorithms for MRHC, especially MRHC on tree pedigrees.
- Efficient fixed-parameter (# of recombinants) algorithms.

# Acknowledgements

Dr. Dajun Qian from City of Hope.

Whitehead/MIT Center for Genome Research
Drs. David Altshuler, Mark Daly, Stacey Gabriel,
and Stephen Schaffner, and their entire group.

Financial support from NSF and CMOST 973 project.