Complete Parsimony Haplotype Inference Problem and Algorithms

Gerold Jäger
Christian-Albrechts-University Kiel (Germany)

joint work with
Sharlee Climer, Weixiong Zhang
Washington University St. Louis (United States)

September 8, 2009

Overview

Overview

Haplotypes and genotypes

- Haplotypes: set of nucleotides in physical proximity on a chromosome strand.
- Genotypes: conflations of haplotype pairs.

HIPP

- Identifying individual haplotypes in a laboratory setting: feasible only for small studies.
- Observation: Number of unique haplotypes in a given population is small.

$\hookrightarrow$ Minimize number of unique haplotypes.

$\hookrightarrow$ Haplotype Inference by Pure Parsimony (HIPP)

Haplotypes and genotypes

- Let $m \in \mathbb{N}$.
- Haplotype: $h \in \{0, 1\}^m$
- Genotype: $g \in \{0, 1, 2\}^m$
- Explanation of genotypes by haplotypes:

  h, h' explain g, if for each $i = 1, 2, \ldots, m$:

  1. $g_i = 0 \;\Rightarrow\; h_i = h'_i = 0$
  2. $g_i = 1 \;\Rightarrow\; h_i = h'_i = 1$
  3. $g_i = 2 \;\Rightarrow\; h_i = 0, h'_i = 1$ or $h_i = 1, h'_i = 0$

  Example: Haplotypes $h = (1, 0, 1, 1)$ and $h' = (0, 0, 1, 0)$ explain genotype $(2, 0, 1, 2)$.

HIPP

- Input: Set of genotypes $G = \{g_1, g_2, \ldots, g_n\}$with $g_i \in \{0, 1, 2\}^m$.
- Output: Set of haplotypes $H = \{h_1, h_2, \ldots, h_p\}$ with $h_j \in \{0, 1\}^m$, where
  1. Each genotype in G can be explained by two haplotypes in H.
  2. $|H| = p$ is minimal.
- Remark: HIPP is NP-hard.

Example

- Input: $g_1 = (2, 1), g_2 = (2, 2)$.
- Solution: $h_1 = (0, 1), h_2 = (0, 0), h_3 = (1, 1)$, because
  1. g_1 can be explained by h_1, h_3.
  2. g_2 can be explained by h_2, h_3.
  3. No set of haplotypes exists with this condition and cardinality 1 or 2.

Disadvantage of HIPP

- HIPP may have multiple optimal solutions.
- Arbitrary algorithm returns arbitrary optimal solution.
- Found optimal solution may be not the biological true solution.

Idea

- Find all optimal HIPP solutions.
- Purpose: Increase probability of finding the biologically true solution.

$\hookrightarrow$ Complete Haplotype Inference by Pure Parsimony (CHIPP)

$\hookrightarrow$ Biological Study:

[Climer, J., Templeton, Zhang; Bioinformatics; 2009]

CHIPP

- Input: Set of genotypes $G = \{g_1, g_2, \ldots, g_n\}$with $g_i \in \{0, 1\}^m$
- Output: Find all HIPP solutions.

Example

- Input: $g_1 = (2, 2)$.
- Solution 1: $h_1 = (0, 0), h_2 = (1, 1)$.
- Solution 2: $h_3 = (0, 1), h_4 = (1, 0)$.

Integer Programming (IP) Model for HIPP

- [Gusfield; 2003]
- Consider all haplotypes $h_1, h_2, \ldots, h_r$ explaining at least one genotype.
- Define for $i = 1, 2, \ldots, r$:
  $$x_i = \begin{cases} 1, & \text{if haplotype } h_i \text{ appears in the HIPP solution} \\ 0, & \text{otherwise} \end{cases}$$
- Structure of IP model
  $$\min \sum_{i=1}^{r} x_i \ \text{ subject to}$$
  $$x_i \in \{0, 1\} \text{ for } i = 1, \ldots, r,$$
  further constraints

IP Algorithm for CHIPP

1. Solve HIPP using the IP model.
   Let $i_1, \ldots, i_p$ be the indices of these haplotypes.
2. Add the following inequality to the IP model:

$$\sum_{s=1}^{p} x_{i_s} \leq p - 1$$

3. Solve the newly expanded IP.

4. Case 1: The new IP has an objective value larger than p.
   $\hookrightarrow$ No new optimal solution exists.
5. Case 2: The new IP has an objective value equal to p.
   $\hookrightarrow$ Another optimal solution has been found.
6. Repeat this process, until all optimal solutions have been found.

Branch-and-Bound (BnB) Algorithm for HIPP

- [Wang, Xu; 2003]
- The algorithm starts with a heuristic solution leading to the initial upper bound for the BnB search.
- The search implicitly considers all possible explaining haplotype pairs for each genotype.
- The best solution found is the optimal solution to be returned.
- If during the search the node cost is equal to or exceeds the upper bound, move on to the next branch.

BnB Algorithm for CHIPP

- Pruning is applied only when the node cost strictly exceeds the upper bound.
- This allows to explore a branch that may lead to another optimal solution.

Concept

- Idea: Transform instance to a smaller and easier equivalent one.
- Advantage: New instance easier to solve.
- Disadvantage: Transformation costs extra time.
- Expectation: For difficult instances extra time is much smaller than saved time.

$\hookrightarrow$ Fixed Parameter Tractability [Niedermeier; 2006]

Backbones

- Backbone haplotypes: Haplotypes appearing in each optimal HIPP solution.
- Backbone genotypes: Genotypes that can be explained by two backbone haplotypes.
- Idea: Backbone genotypes can be omitted for HIPP/CHIPP.

  Reason: They can be explained by haplotypes of any optimal solution.

$\hookrightarrow$ Compute backbone haplotypes.

$\hookrightarrow$ With backbone haplotypes compute backbone genotypes (easy).

Computation of trivial backbone haplotypes

- A genotype with no 2 leads to one backbone haplotype.
  Example: $g = (1, 0, 1) \Rightarrow h = (1, 0, 1)$.
- A genotype with one 2 leads to two backbone haplotypes.
  Example: $g = (1, 0, 2) \Rightarrow h = (1, 0, 0)$ and $h' = (1, 0, 1)$.
- This are trivial backbone haplotypes.

Computation of non-trivial backbone haplotypes

- Idea: When a backbone haplotype is omitted, no optimal solution to HIPP can be found any more.
- This haplotype must be contained in the optimal solution.

Example

- Input: $g_1 = (1, 0, 2, 2), g_2 = (2, 2, 1, 0), g_3 = (2, 2, 1, 2)$.
- Solution 1: $h_1 = (1, 0, 1, 0), h_2 = (1, 0, 0, 1)$, $h_3 = (0, 1, 1, 0), h_4 = (0, 1, 1, 1)$, because
  1. g_1 can be explained by h_1, h_2.
  2. g_2 can be explained by h_1, h_3.
  3. g_3 can be explained by h_1, h_4.
  4. No set of haplotypes exists with this condition and cardinality $1, 2$ or 3.
- Question: Is $h_1 = (1, 0, 1, 0)$ a backbone haplotype?
- Answer: Yes, because

  a solution without $h_1 = (1, 0, 1, 0)$ contains $h_5 = (1, 0, 0, 0), h_6 = (1, 0, 1, 1)$, $h_7 = (0, 0, 1, 0), h_8 = (1, 1, 1, 0)$ and another 5-th haplotype.

Example

- Input 1: $g_1 = (2)$.
  Solution: $h_1 = (0), h_2 = (1)$.
- Input 2: $g_2 = (2, 2)$.
  Solution 1: $h_3 = (0, 0), h_4 = (1, 1)$.
  Solution 2: $h_5 = (0, 1), h_6 = (1, 0)$.

Equal Column Technique

- Copy column 1 of the solution of input 1 to column 2.

$\hookrightarrow$ Solution 1 of input 2.

$\hookrightarrow$ Equal column technique for HIPP [Wang, Xu; 2003].

- But: original equal column technique does not work directly for CHIPP because

  solution 2 of input 2 is missed by this method.

$\hookrightarrow$ Idea: For an equal column enumerate all possible combinations of 0 and 1 with the restriction that

  - each resulting haplotype explains at least one genotype.

Decomposability Algorithm

- Two genotypes g and g' are non-overlapping, if $i \in \{1, 2, \ldots, m\}$ exists with:
  - $g_i = 0,\ g'_i = 1$ or $g_i = 1,\ g'_i = 0$.
- Observation: Two non-overlapping genotypes do not share any explaining haplotype.

$\hookrightarrow$ Idea: Compose CHIPP solutions of sub-classes whose genotypes do not overlap.

Example

- Input 1: $g_1 = (0, 2, 2)$.
  Solution 1: $h_1 = (0, 0, 0), h_2 = (0, 1, 1)$.
  Solution 2: $h_3 = (0, 0, 1), h_4 = (0, 1, 0)$.
- Input 2: $g_2 = (1, 2, 2)$.
  Solution 1: $h_5 = (1, 0, 0), h_6 = (1, 1, 1)$.
  Solution 2: $h_7 = (1, 0, 1), h_8 = (1, 1, 0)$.
- Input 3: $g_1 = (0, 2, 2), g_2 = (1, 2, 2)$.
  Solution 1: h_1, h_2, h_5, h_6.
  Solution 2: h_1, h_2, h_7, h_8.
  Solution 3: h_3, h_4, h_5, h_6.
  Solution 4: h_3, h_4, h_7, h_8.

Easy Instances

- Optimized slightly worse than baseline algorithms.

$\hookrightarrow$ Typical example: IP-Bas.: 1 sec.   IP-Opt.: 1.68 sec.

Difficult Instances

- Optimized significantly superior to baseline algorithms.

$\hookrightarrow$ Typical example: BnB-Bas.: $>$ 6 h.   BnB-Opt.: 275.69 sec.

Optimized Versions

IP better performance than BnB.

Optimization Techniques

- Strongly depends on structure of instances, e.g., backbone technique effective for instances with many backbones.

Thanks for your attention!