

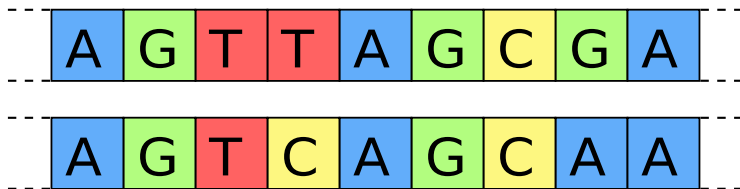
Extended Islands of Tractability for Parsimony Haplotyping

Rudolf Fleischer¹, Jiong Guo², Rolf Niedermeier³, Johannes
Uhlmann³, Yihui Wang¹, Mathias Weller³, and Xi Wu¹

¹Fudan University Shanghai, ²Universität des Saarlandes,
and ³Friedrich-Schiller-Universität Jena

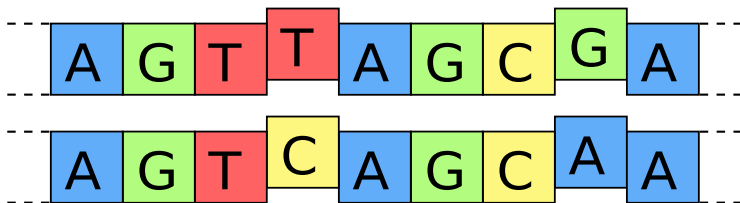
June 22, 2010

First, Some Biology...



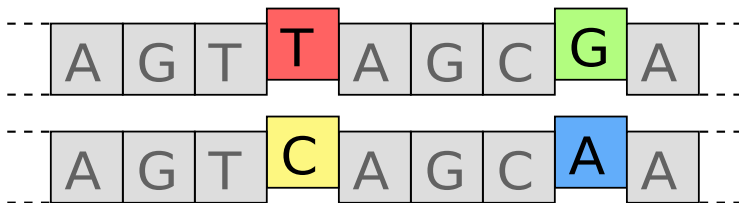
approx. 0.1% of human nucleotide sites differ between individuals

First, Some Biology...



approx. 0.1% of human nucleotide sites differ between individuals

First, Some Biology...



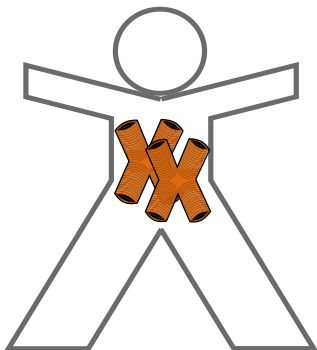
the sequence of SNPs is called a [haplotype](#)

First, Some Biology...

TGCA

the sequence of SNPs is called a **haplotype**

First, Some Biology...

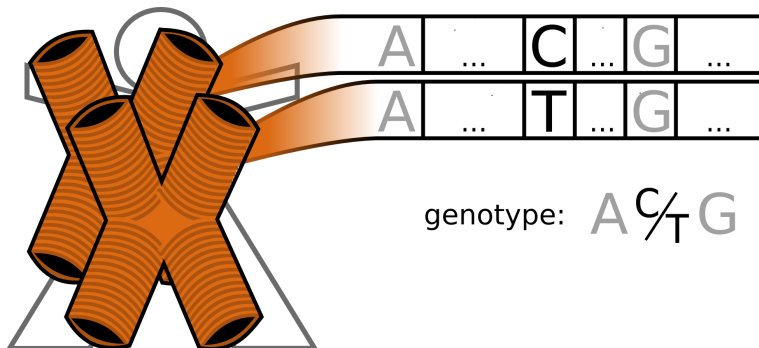


humans are diploid $\leadsto$ 2 chromosome sets

haplotype = SNP sequence in **one** chromosome set

genotype = SNP sequence in the combined chromosome sets

First, Some Biology...



humans are diploid $\leadsto$ 2 chromosome sets

haplotype = SNP sequence in **one** chromosome set

genotype = SNP sequence in the combined chromosome sets

Motivation

Haplotype Inference

goal: find relation between certain SNPs and genetic diseases

problem: difficult (expensive) to sequence both haplotypes

but: easy (cheap) to sequence the genotype instead

~→ **idea:** sequence genotype and computationally infer haplotypes

Motivation

Haplotype Inference

goal: find relation between certain SNPs and genetic diseases

problem: difficult (expensive) to sequence both haplotypes

but: easy (cheap) to sequence the genotype instead

~> **idea:** sequence genotype and computationally infer haplotypes

Problems

- impossible to infer haplotypes of just 1 genotype
~> sequence and infer groups/populations
- which explanation should be preferred if there are multiple?
~> parsimony ("Ockham's razor")
- how to perform the actual computation fast?

Motivation

Haplotype Inference

goal: find relation between certain SNPs and genetic diseases

problem: difficult (expensive) to sequence both haplotypes

but: easy (cheap) to sequence the genotype instead

~> **idea:** sequence genotype and computationally infer haplotypes

Problems

- impossible to infer haplotypes of just 1 genotype
~> sequence and infer groups/populations
- which explanation should be preferred if there are multiple?
~> parsimony ("Ockham's razor")
- how to perform the actual computation fast?

Preliminary Definitions

Definition (Haplotype, Genotype, Resolve)

haplotype h = string over $\{0, 1\}$

genotype g = string over $\{0, 1, 2\}$

h_1, h_2 **resolve** $g \Leftrightarrow$

for all $i \in \mathbb{N}$, $g[i] = h_1[i] = h_2[i]$ or $g[i] = 2$ and $h_1[i] \neq h_2[i]$

multiset $H \rightsquigarrow \text{res}(H)$

multiset H **resolves** multiset $G \Leftrightarrow G \subseteq \text{res}(H)$

Example

haplotype1:	0	0	1	0	1	1	1	0	0	0	1
haplotype2:	0	0	1	1	0	1	0	0	1	1	1
genotype:	0	0	1	2	2	1	2	0	2	2	1

Preliminary Definitions

Definition (Haplotype, Genotype, Resolve)

haplotype h = string over $\{0, 1\}$

genotype g = string over $\{0, 1, 2\}$

h_1, h_2 **resolve** $g \Leftrightarrow$

for all $i \in \mathbb{N}$, $g[i] = h_1[i] = h_2[i]$ or $g[i] = 2$ and $h_1[i] \neq h_2[i]$

multiset $H \rightsquigarrow \text{res}(H)$

multiset H **resolves** multiset $G \Leftrightarrow G \subseteq \text{res}(H)$

Example

haplotype1:	0	0	1	0	1	1	1	0	0	0	1
haplotype2:	0	0	1	1	0	1	0	0	1	1	1
genotype:	0	0	1	2	2	1	2	0	2	2	1

Preliminary Definitions

Definition (Haplotype, Genotype, Resolve)

haplotype h = string over $\{0, 1\}$

genotype g = string over $\{0, 1, 2\}$

h_1, h_2 **resolve** $g \Leftrightarrow$

for all $i \in \mathbb{N}$, $g[i] = h_1[i] = h_2[i]$ or $g[i] = 2$ and $h_1[i] \neq h_2[i]$

multiset $H \rightsquigarrow \text{res}(H)$

multiset H **resolves** multiset $G \Leftrightarrow G \subseteq \text{res}(H)$

Example

haplotype1:	0	0	1	1	1	1	1	0	0	0	1
haplotype2:	0	0	1	0	0	1	0	0	1	1	1
genotype:	0	0	1	2	2	1	2	0	2	2	1

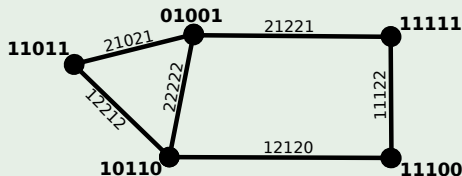
Preliminary Definitions

Definition (Haplotype Graph)

Given H and $G \subseteq \text{res}(H)$, define the **haplotype graph** of H and G :

- $|H|$ vertices (labeled by H)
- $|G|$ edges (labeled by G)
- haplotypes of the endpoints of an edge resolve its genotype

Example



Haplotype Inference by Parsimony

Definition (Haplotype Inference by Parsimony)

Input: multiset G of length- m genotypes, integer $k \geq 0$

Question: $\exists$ multiset H of k haplotypes that resolves G ?

Example

genotypes	haplotype graph	haplotypes
11122		01001
12120		10110
12212		11011
21021		11100
21221		11111
22222		11111

Previous Work

Complexity

- NP-hard [Halldórsson et al., DMTCS '03]
- APX-hard [Lancia et al., INFORMS '04]
- many special cases in P (e.g. [Lancia & Rizzi, ORL '06])

Algorithms

- Factor- 2^{d-1} -Approximation [Lancia & Rizzi, ORL '06]
- $O(2^{|G| \cdot d})$ Branch&Bound [Wang & Xu, Bioinformatics '03]
- $O(k^{2k^2} \cdot m)$ Algorithm [Sharan et al., TCBB '06]

$k := \# \text{haplotypes}$, $m := \text{genotype length}$, $d := \max \# \text{ 2's in a genotype}$

Our Contribution

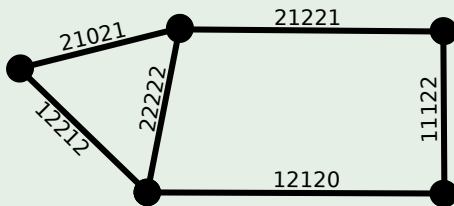
- $k^{4k} \cdot \text{poly}(|G|, m)$ time algorithm
- polynomial-time solvable special case:
“Induced Haplotype Inference”

Inference Graph

Definition (Inference Graph)

inference graph Γ of G = an order- k graph with edges **consistently** labeled by the genotypes in G

Example



If Only We Had an Inference Graph...

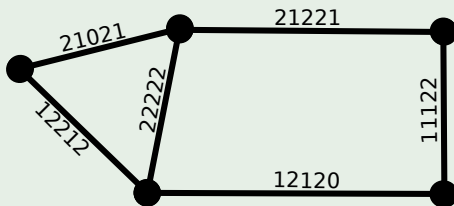
Observation (non-bipartite components of Γ)

$C = \text{cycle in } \Gamma$

$\leadsto$ for all i , $|\{g \in C \mid g[i] = 2\}|$ is even

non-bipartite components of $\Gamma \leadsto O(|\Gamma| \cdot m)$ time

Example



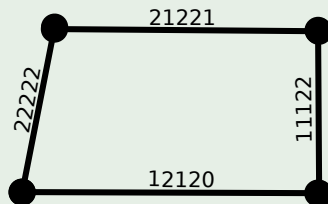
If Only We Had an Inference Graph...

Observation (bipartite components of Γ)

all $g[i] = 2$ for some $i \Rightarrow$ choose arbitrarily

bipartite components of $\Gamma \rightsquigarrow O(|\Gamma| \cdot m)$ time

Example



If Only We Had an Inference Graph...

Observation

(G, k) *yes-instance* with *solution* H

$\Rightarrow \exists \Gamma$ *extendable* ($O(|\Gamma| \cdot m)$ *time*) to a *haplotype graph* of H and G

If Only We Had an Inference Graph...

Observation

(G, k) yes-instance with solution H

$\Rightarrow \exists \Gamma$ extendable ($O(|\Gamma| \cdot m)$ time) to a haplotype graph of H and G

algorithmic idea: guess Γ

If Only We Had an Inference Graph...

Observation

(G, k) yes-instance with solution H

$\Rightarrow \exists \Gamma$ extendable ($O(|\Gamma| \cdot m)$ time) to a haplotype graph of H and G

algorithmic idea: guess Γ

better idea: guess a “spanning” subgraph of Γ

Algorithm Outline

Algorithm

- 1 guess “spanning” subgraph of Γ
- 2 infer the haplotype multiset $H \rightsquigarrow O(k \cdot m)$ time
- 3 check whether H resolves $G \rightsquigarrow O(k^2 \cdot m)$ time

Algorithm Outline

Algorithm

- ① guess “spanning” subgraph of Γ
 - ① guess a size- k genotype subset of $G \rightsquigarrow O(k^{2k})$ possibilities
 - ② for these genotypes, guess 2 (of k) vertices $\rightsquigarrow O(k^{2k})$ possibilities
- ② infer the haplotype multiset $H \rightsquigarrow O(k \cdot m)$ time
- ③ check whether H resolves $G \rightsquigarrow O(k^2 \cdot m)$ time

Algorithm Outline

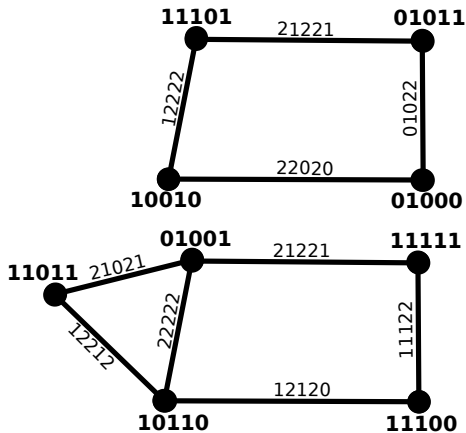
Algorithm

- ① guess “spanning” subgraph of Γ
 - ① guess a size- k genotype subset of $G \rightsquigarrow O(k^{2k})$ possibilities
 - ② for these genotypes, guess 2 (of k) vertices $\rightsquigarrow O(k^{2k})$ possibilities
- ② infer the haplotype multiset $H \rightsquigarrow O(k \cdot m)$ time
- ③ check whether H resolves $G \rightsquigarrow O(k^2 \cdot m)$ time

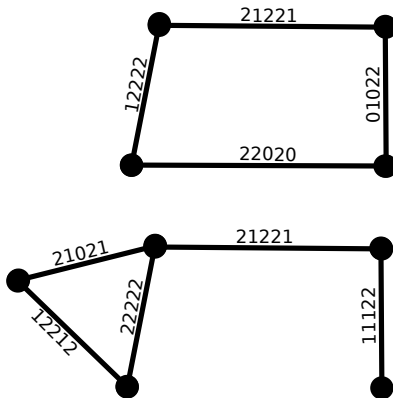
Theorem

Haplotype Inference by Parsimony can be solved in $O(k^{4k+2} \cdot m)$ time.

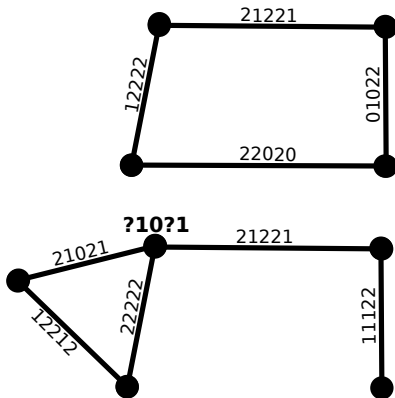
Example



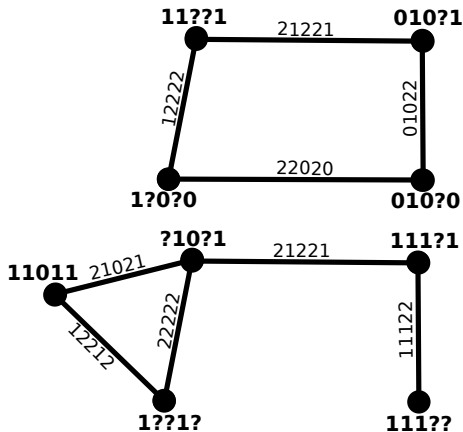
Example



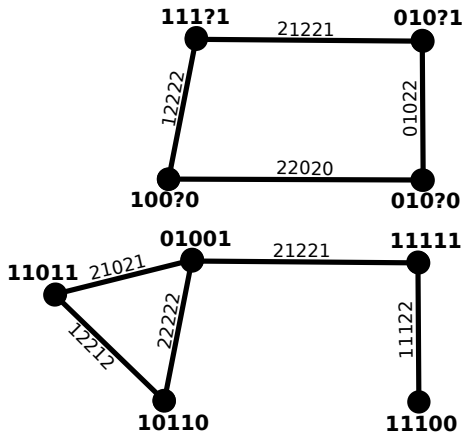
Example



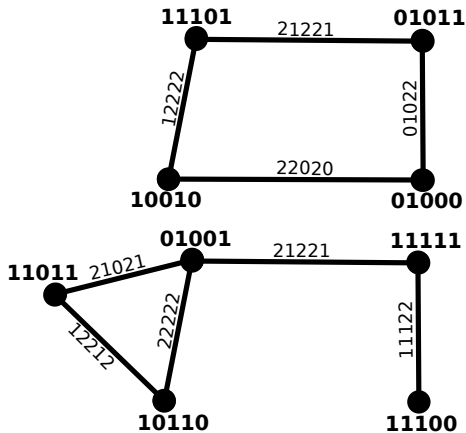
Example



Example



Example



Induced Haplotype Inference

recall: H resolves $G \Leftrightarrow G \subseteq \text{res}(H)$

what if $G = \text{res}(H)$? $\leadsto$ haplotype graph is a clique

Definition

$G = \text{res}(H) \leadsto H$ induces G

Induced Haplotype Inference

recall: H resolves $G \Leftrightarrow G \subseteq \text{res}(H)$

what if $G = \text{res}(H)$? $\leadsto$ haplotype graph is a clique

Definition

$G = \text{res}(H) \leadsto H$ induces G

Definition (Induced Haplotype Inference by Parsimony)

Input: multiset G of length- m genotypes

Question: $\exists$ multiset H of haplotypes that induces G ?

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Strategy: Divide & Conquer

divide-step

base cases

merge-step

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Strategy: Divide & Conquer

divide-step $\leadsto$ OK

base cases

merge-step

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Strategy: Divide & Conquer

divide-step $\leadsto$ OK

base cases $\leadsto$ OK

merge-step

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Strategy: Divide & Conquer

divide-step $\leadsto$ OK

base cases $\leadsto$ OK

merge-step $\leadsto$ **problem!**

Observations & Algorithm

Observation

G can be partitioned “nicely” into G_0 , G_1 , and G_2 .

Observation

$G_2 \neq \emptyset$ but $G_0 = \emptyset$ or $G_1 = \emptyset \leadsto \text{poly}$

$|G_0| = |G_1| = 1 \leadsto \text{poly}$ (although we may get 2 solutions)

Strategy: Divide & Conquer

divide-step $\leadsto$ OK

base cases $\leadsto$ OK

merge-step $\leadsto$ **problem!**

need to find a way to compute H_1 for given H_0 and G_2

Extending a Subclique Solution

Observation

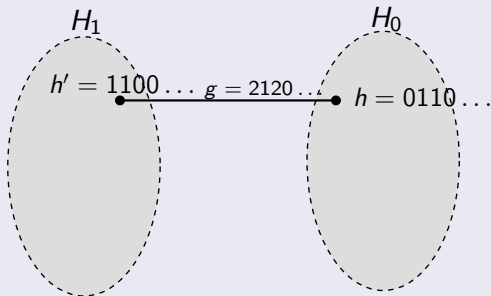
Let H_0 induce G_0 and let g be a genotype in G_2 with the smallest number of 2's. $\leadsto$ All $h \in H_0$ that are consistent with g are equal.

Extending a Subclique Solution

Observation

Let H_0 induce G_0 and let g be a genotype in G_2 with the smallest number of 2's. $\leadsto$ All $h \in H_0$ that are consistent with g are equal.

Proof Idea

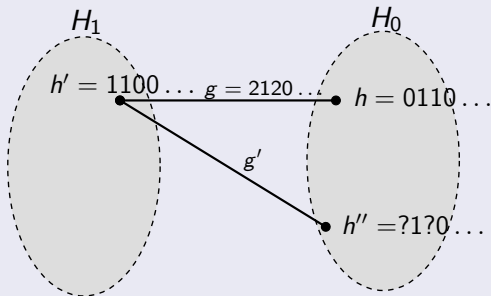


Extending a Subclique Solution

Observation

Let H_0 induce G_0 and let g be a genotype in G_2 with the smallest number of 2's. $\leadsto$ All $h \in H_0$ that are consistent with g are equal.

Proof Idea



Concluding The Induced Problem

Divide & Conquer like algorithm

divide steps	$O(G \cdot k)$
base solutions	$O(G \cdot m)$
extends (merges)	$O(G \cdot k \cdot m)$
all in all	$O(G \cdot k \cdot m)$

Conclusion

what we saw...

- Induced Haplotyping in $O(|G| \cdot k \cdot m)$ time
- $2^{O(k^2 \log k)}$ -time algorithm improved to $2^{O(k \log k)}$

Conclusion

what we saw...

- Induced Haplotyping in $O(|G| \cdot k \cdot m)$ time
- $2^{O(k^2 \log k)}$ -time algorithm improved to $2^{O(k \log k)}$

also in the paper

- results also hold for sets instead of multisets
- results basically also hold for constrained variant
- data reduction ($O(2^k \cdot k^2)$ -bit kernel)

Conclusion

what we saw...

- Induced Haplotyping in $O(|G| \cdot k \cdot m)$ time
- $2^{O(k^2 \log k)}$ -time algorithm improved to $2^{O(k \log k)}$

also in the paper

- results also hold for sets instead of multisets
- results basically also hold for constrained variant
- data reduction ($O(2^k \cdot k^2)$ -bit kernel)

future work

- find polynomial kernel (or prove nonexistence)
- distance from triviality measures
- find $2^{O(k)}$ time algorithm

Thank you