Complexity of Pure Parsimony and Maximum Likelihood Haplotype Phasing

Sorin Istrail

October 3, 2014

Haplotype Phasing by Pure Parsimony

- Pure Parsimony Haplotype Phasing (PPHP) aims to find the smallest set of haplotypes which explains the genotype data
- Potential Issues
  - There could be many optimal solutions
  - Biology is not parsimonious
- PPHP is NP-hard[1]
- Reduction from Minimum Clique Partition

[1] E. Hubbell Personal Communication (2001)

One Slide Introduction to NP-hardness Proofs

- To prove PPHP is NP-hard we have to describe how to solve a different problem which uses an *unknown* algorithm to solve PPHP as a subroutine.
- That is, **To prove PPHP is NP-hard we must reduce a known NP-hard problem to PPHP**

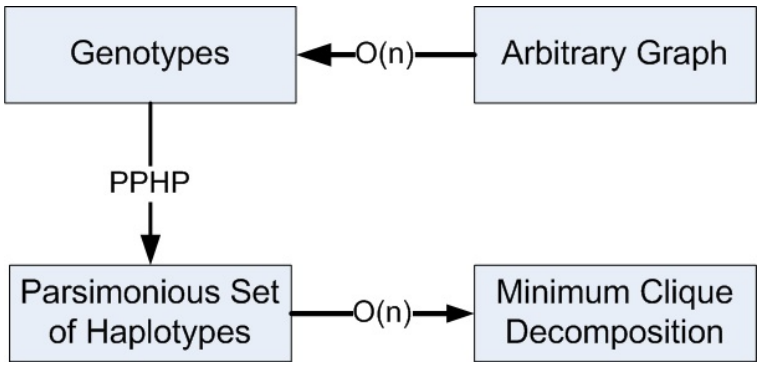
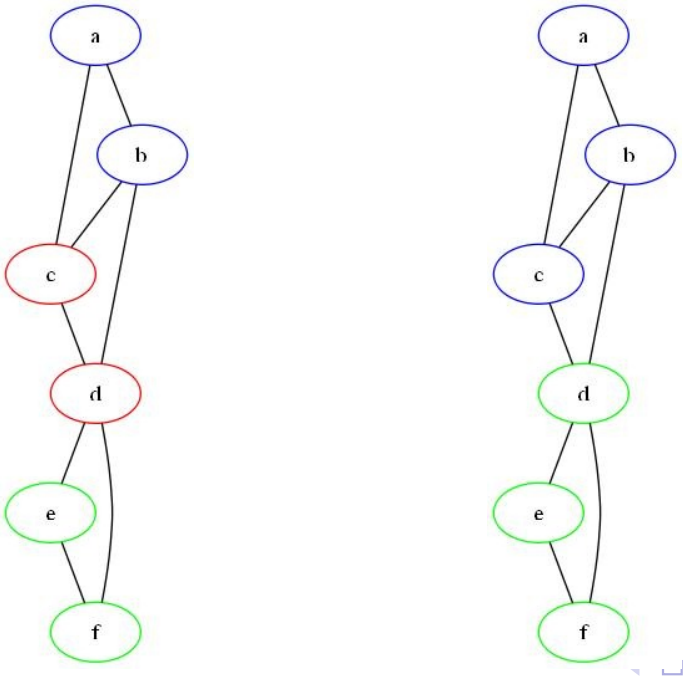


Figure : A sketch of the NP-hardness proof.

Minimum Clique Partition

- For some graph $G(V, E)$, a **clique partition** is a partitioning of G into V disjoint subsets $V_1, ..., V_k$ such that V_i is a clique (complete subgraph) for $1 \leq i \leq k$.
- A **minimum clique partition** is a clique partitioning of minimum number of disjoint subsets k
- Known to be NP-hard



PPHP is NP-Hard

- New notation for genotypes: 0 is the major allele, 1 is the ambiguous site, and 2 is the minor allele
- Haplotypes remain the same: 0 is the major allele, 1 is the minor allele

  Example:

$$\begin{array}{c} 0\ 1\ 0 \\ 0\ 1\ 1 \\ \hline 0\ 2\ 1 \end{array}$$

- h and h' explains g is denoted as $g = h \oplus h'$ or $g = h + h'$ (because now a genotype is literally the sum of the two haplotypes)

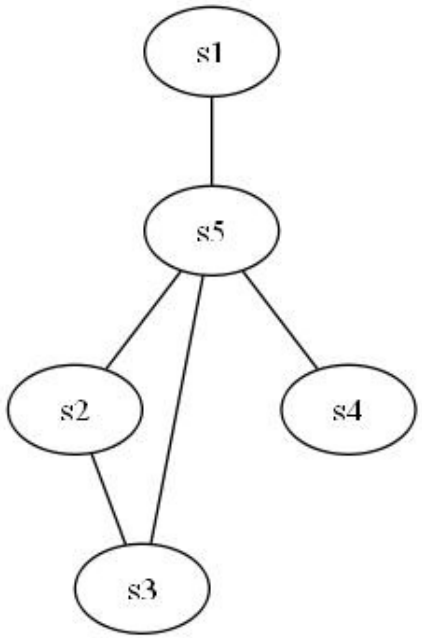
PPHP is NP-hard

- We are given any graph $Z = (V, E)$
- Associate a set of genotypes $G = \{g_1, g_2, ...g_n\}$ with the graph Z
- We use the following algorithm to create the genotypes (which are of length $2n$) [Notation $g_{ij} = g_i[j] = j^{th}$ coordinate of the i^{th} genotype]
  - For the first n positions $g_{ij} = 2$ if $i = j$; $g_{ij} = 1$ if V_i is connected by an edge to V_j; and $g_{ij} = 0$ otherwise.
  - For the next n positions, create an identity matrix: $g_{ij} = 1$ if $j = n + i$ and 0 otherwise, $1 \leq i \leq n$ *and* $n + 1 \leq j \leq 2n$

PPHP is NP-hard

As an example, the following graph corresponds to the accompanying genotype list (n=5).

Vertex	Genotype
s_1	$g(s_1) = 20001\ 10000$
s_2	$g(s_2) = 02101\ 01000$
s_3	$g(s_3) = 01201\ 00100$
s_4	$g(s_4) = 00021\ 00010$
s_5	$g(s_5) = 11112\ 00001$



PPHP is NP-hard

- We now create a haplotype set $H = h_1, h_2, ...h_T$. We can write $g_{i,n+i} = h_{l,n+i} + h_{m,n+i}$ and since $g_{i,n+i} = 1$ for all i, we have either $h_{l,n+i} = 1$ or $h_{m,n+i} = 1$ but not both.
- For example, vertex A or genotype $g_1 = 2000110000$ has the property $g_{1,6} = 1$ while all other genotypes have the property $g_{\alpha,6} = 0$ for $2 \leq \alpha \leq n$.
- WLOG, we may assume that $h_{m,n+i} = 1$.
- These n haplotypes $h_{m_1}, ..., h_{m_n}$ are *unique* and cannot explain any other genotype (because every other haplotype has a 0 in that SNP position).

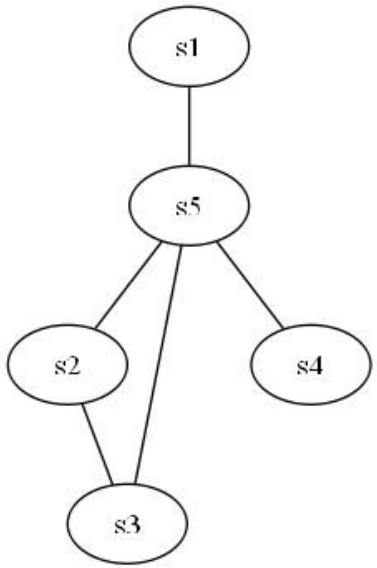
PPHP is NP-hard

Let the first n values be represented by genotype characters and the last n values be represented by haplotype characters. We now have the following partial haplotypes:

ID	Haplotype
h_{m_1}	20001 10000
h_{m_2}	02101 01000
h_{m_3}	01201 00100
h_{m_4}	00021 00010
h_{m_5}	11112 00001
h_{l_1}	20001 00000
h_{l_2}	02101 00000
h_{l_3}	01201 00000
h_{l_4}	00021 00000
h_{l_5}	11112 00000

PPHP is NP-hard

- As we observed in the previous slide, let h_{l_i} be the haplotype that is consistent with a set of Q genotypes $\{g_{s_t}, ..., g_{s_u}\}$.
- We want to show that $D = \{s_t, ..., s_u\}$ is a clique in Z.
- We can explain genotype g_i by two haplotypes: $g_i = h_{l_i} + h_{m_i}$. (remember: h_m is not shared)
- In our example, let $D = \{s_2, s_3, s_5\}$ for some h_{l_i}.



PPHP is NP-hard

- We have $g_{i,i} = 2$ for $1 \leq i \leq n$ which means both h_{l_i} and h_{m_i} have a 1 at position i
- Therefore, for the shared haplotype, we must have $h_{l,k} = 1$ for all $k = t, ..., u$.
- This in turn means that $g_{i,j} = 1$ for all i≠j since a 2 in the genotype occurs *only* on the diagonal and we need a 1 at that position for the shared haplotype to be consistent with the given genotype.
- The set of genotypes such that $g_{i,j} = 1$ corresponds to vertices s_i and s_j being adjacent on the graph Z while if there is 0 at that index, the vertices are not adjacent.

PPHP is NP-hard

Vertex	Genotype
s_1	$g(s_1) = 20001\ 10000$
s_2	$g(s_2) = 02101\ 01000$
s_3	$g(s_3) = 01201\ 00100$
s_4	$g(s_4) = 00021\ 00010$
s_5	$g(s_5) = 11112\ 00001$

ID	Haplotype
h_{m_1}	10000 10000
h_{m_2}	01000 01000
h_{m_3}	00100 00100
h_{m_4}	00010 00010
h_{m_5}	00001 00001
h_{l_1}	10001 00000
h_{l_2}	01101 00000
h_{l_3}	01101 00000
h_{l_4}	00011 00000
h_{l_5}	11111 00000

PPHP is NP-hard

- Therefore, the set of genotypes such that $g_{i,j} = 1$ for all i≠j corresponds to a set of vertices which all share edges with each other, i.e. D is a clique in graph Z as $s_i \leftrightarrow s_i$ for all i≠j.
- The cliques that are created in this manner will be *disjoint*, since each clique will have a different haplotype in common.

PPHP is NP-hard

- The number of haplotypes constructed is a constant factor of the number of edges in the graph.
- So, to minimize the number of haplotypes needed to phase the genotypes is equivalent to minimizing the number of cliques to partition the graph Z

PPHP is NP-hard

- We now show that given a clique partitioning of Z with C cliques $\{c_1, ...c_c\}$, we can construct the set of haplotypes B=$\{b_1, ..., b_{n+c}\}$ and vice versa. Given C, construct B according to the following algorithm:

for $1 \leq k \leq c$

$$b_{n+k,j} = \begin{cases} 1 & \text{if } s_j \in C_k \\ 0 & \text{otherwise} \end{cases}$$

for each $s_i \in C_k$

$$b_{i,j} = g_{i,j} - b_{n+k,j}$$

- By construction B explains $G(Z)$. Thus a most parsimonious set of haplotypes must have exactly $n + p$ elements where p is the number of cliques in a minimal clique partition.
- Note: in the above representation, b_i $1 \leq i \leq n$ is the complement haplotype to $b_{n+k,j}$ and will not be shared among

PPHP is NP-hard

- Let's use the algorithm to generate haplotypes for our example: $c_1 = \{s_2, s_3, s_5\}$, $c_2 = \{s_1\}$, $c_3 = \{s_4\}$

Vertex	Genotype
s_1	$g(s_1) = 20001\ 10000$
s_2	$g(s_2) = 02101\ 01000$
s_3	$g(s_3) = 01201\ 00100$
s_4	$g(s_4) = 00021\ 00010$
s_5	$g(s_5) = 11112\ 00001$

- $g(s_2) : b_6 = 0110100000 \oplus b_2 = 0100001000$
- $g(s_3) : b_6 = 0110100000 \oplus b_3 = 0010000100$
- $g(s_5) : b_6 = 0110100000 \oplus b_5 = 1001100001$
- $g(s_1) : b_7 = 1000000000 \oplus b_1 = 1000110000$
- $g(s_4) : b_8 = 0001000000 \oplus b_4 = 0001100010$

PPHP is NP-hard

- Transforming a parsimonious set of haplotypes into a minimum clique decomposition is trivial.
- Assign all genotypes which share a haplotype into a set
- These sets define the minimum clique decomposition

PPHP is NP-hard

- So, we can transform any graph into a polynomial sized set of genotypes in polynomial time ($\longleftarrow$).
- We assumed the existence of an algorithm to solve the PPHP problem ($\downarrow$).
- We showed that we can transform the minimum set of haplotypes needed to explain the genotypes to the minimum set of disjoint cliques in polynomial time ($\longrightarrow$).

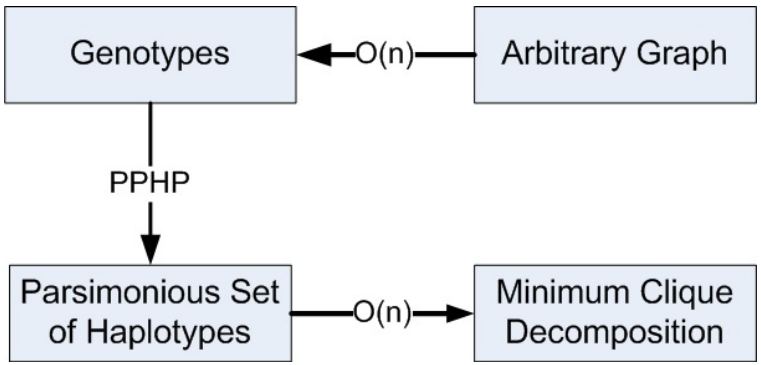


Figure : A sketch of the NP-hardness proof.

PPHP is NP-hard

- This reduction implies that if we had an algorithm to solve PPHP in polynomial time then we would also know how to solve the minimum clique partitioning problem in polynomial time (implying P=NP)
- Thus, PPHP is NP-hard. □

Maximum Likelihood

- Maximum Likelihood is the process of finding the parameter values of a statistic which makes the observed likelihood distribution a maximum.
- For example, let the set of observations of people's heights in this classroom be denoted as $x_1, ..., x_n$ and are distributed normally. By solving the first derivative of the likelihood function of the Normal Distribution in respect to μ we can find the ML estimator of the mean is $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$

Maximum Likelihood

- We will now present the proof that finding a ML solution to the haplotype phasing problem is NP-hard
- This does not imply that the expectation-maximization algorithm is NP-hard as it only finds local maximum rather than the global maximum
- Because we assume random mating, the probability of selecting a haplotype is proportional to its frequency, so, an ML solution corresponds to a set of haplotypes and frequencies such that the probability of generating the observed genotypes is maximum
- We reduce from maximum clique

Maximum Clique

- Reminder: A clique is a complete subgraph.

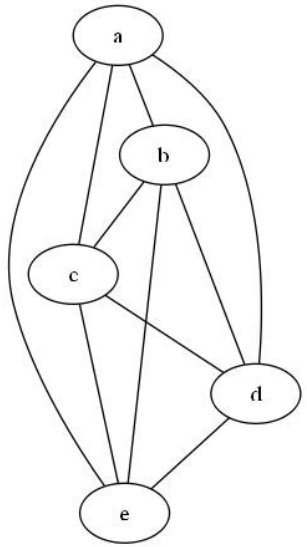


Figure : Clique of size 5

- Finding the clique with the largest number of vertices in an arbitrary graph is NP-hard.

Maximum Likelihood Haplotype Phasing is NP-Hard

- Starting from an arbitrary graph Z, construct a new graph Y by making c copies of Z and connecting all the copies of each vertex
- Each vertex in Z corresponds to a c-clique in Y

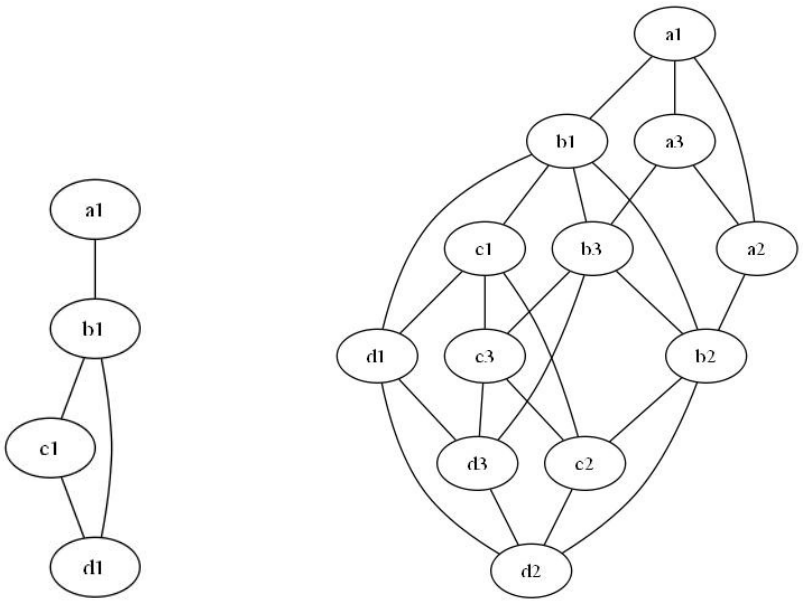


Figure : Left: Z. Right: Y with $c = 2$.

Maximum Likelihood Haplotype Phasing is NP-Hard

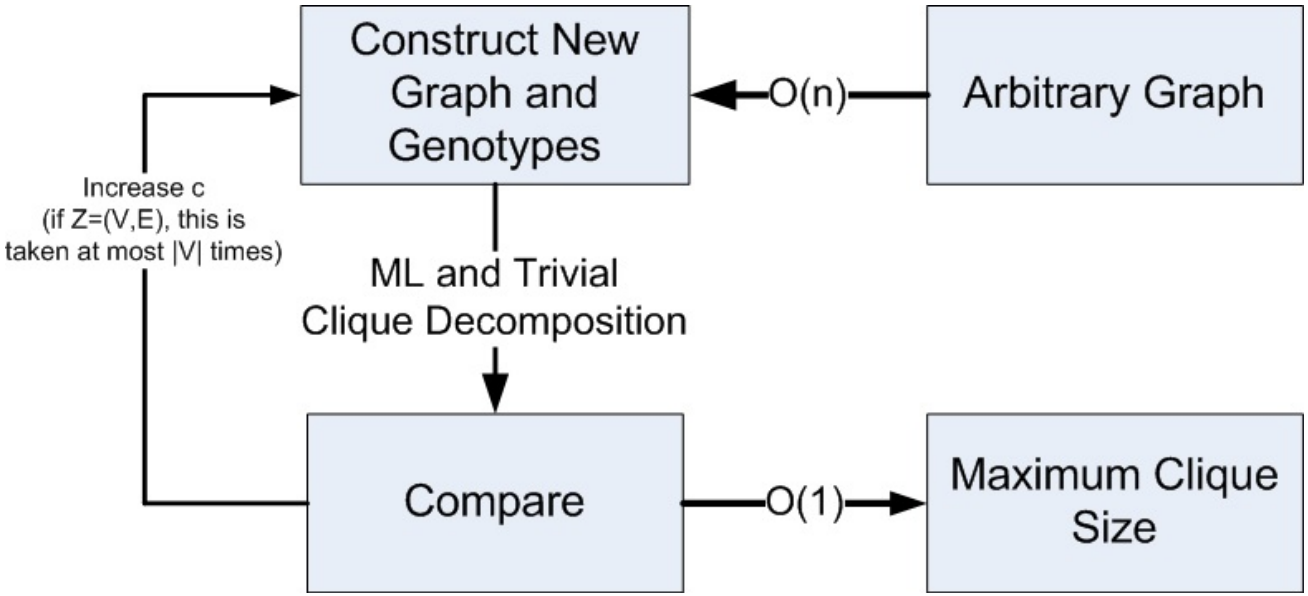


Figure : Outline of ML Phasing is NP-hard proof

Maximum Likelihood Haplotype Phasing is NP-Hard

- Construct a set of genotypes G from Y using the same technique in the Pure Parsimony Haplotype Phasing proof
- For example, consider the simple Z and its Y with $c = 1$

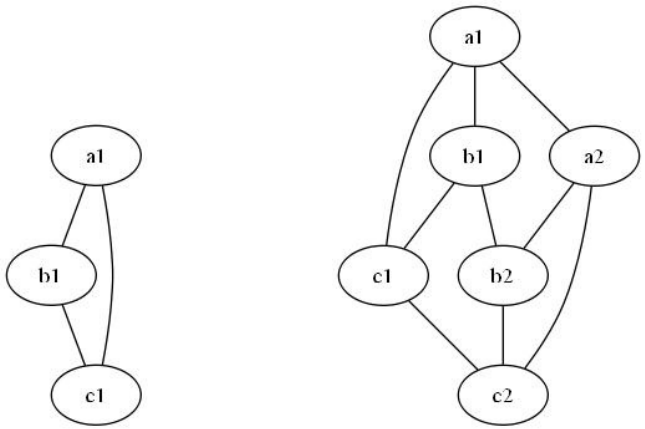


Figure : Left: A simple Z. Right: Corresponding Y with $c = 1$.

Maximum Likelihood Haplotype Phasing is NP-Hard

Example continued:

Vertex	Genotype
a1	$g(a1) = 211100\ 100000$
b1	$g(b1) = 121010\ 010000$
c1	$g(c1) = 112001\ 001000$
a2	$g(a2) = 100211\ 000100$
b2	$g(b2) = 010121\ 000010$
c2	$g(c2) = 001112\ 000001$

Maximum Likelihood Haplotype Phasing is NP-Hard

- Any solution to the haplotype phasing problem for G (this isn't necessarily the ML solution) corresponds to a clique decomposition in Y as we proved previously
- For example, consider these two solutions (we only represent the shared haplotypes here, the non-shared are trivially determined):
  - $sol_1 = \{100100000000, 010010000000, 001001000000\}$
  - $sol_2 = \{111000000000, 000111000000\}$

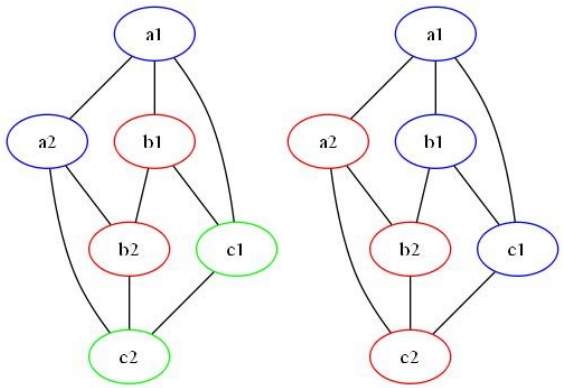


Figure : Left: subgraph induced by sol_1. Right: subgraph induced by sol_2.

Maximum Likelihood Haplotype Phasing is NP-Hard

Question: Is there a relationship between the maximum likelihood solution and the maximum clique size?

- In fact, there is such a relationship!
- First, note that Y can always be decomposed into cliques of size c
- This means that each haplotype can be shared at least c times
- The only way that any clique in Y can be larger than c is if there was a clique in the original graph whose size was larger than c

Maximum Likelihood Haplotype Phasing is NP-Hard

Question: Are solutions with fewer haplotypes more likely?

- Answer: Yes, because of Property 1.
- First, we will show that any solution where a single genotype is explained by more than one pair of haplotypes cannot be the most likely solution
- Second, we will show that the likelihood of any particular genotype is inversely proportional to the total number of haplotypes
- The combination of these two facts means that the most likely solution must have the smallest possible number of haplotypes

Maximum Likelihood Haplotype Phasing is NP-Hard

- Suppose a genotype g is explained by two pairs of haplotypes: $h_1 + h_2$ and $h_3 + h_4$; the likelihood g is then $P(h_1)P(h_2) + P(h_3)P(h_4)$
- Now because of Property 1, we know that out of each pair of haplotypes, one must be unique to g. Let's say that the unique haplotypes are h_1 and h_3
- In this case, we can increase the likelihood of the overall solution by setting $P(h_3)$ to 0, and adding the leftover probability to $P(h_1)$
- This is guaranteed to increase the likelihood of g without affecting the likelihood of any other genotypes
- Therefore the maximum likelihood solution cannot have genotypes that are explained by more than one pair of haplotypes.

Maximum Likelihood Haplotype Phasing is NP-Hard

Since the maximum likelihood solution assigns only one pair of haplotypes to each genotype, the overall likelihood of the solution is

$$\prod P(h_i)^{n_i}$$

where n_i is the number of times h_i is used by any genotype

- If there are m haplotypes, then this product can be maximized by setting $P(h_i)$ to $\frac{n_i}{m}$
- Now consider what happens if we decrease m by 1
- First, $P(h_i)$ will increase because the divisor m has decreased
- Second, $P(h_i)$ will also increase because the numerator n_i has, on average, increased
- Thus the overall likelihood of all the genotypes will have increased

Maximum Likelihood Haplotype Phasing is NP-Hard

We have shown that given our technique for constructing genotypes from graphs, the maximum likelihood solution corresponds to the most parsimonious solution, which corresponds to the solution with the fewest number of cliques in Y

Now we use this fact to infer the size of the largest clique in Z.

Maximum Likelihood Haplotype Phasing is NP-Hard

- Start with $c = 1$, and compare the likelihood of the ML solution with the likelihood of the decomposition into cliques of size c
- If the ML solution has a higher likelihood, then the size of the max clique must be larger than c
- Keep increasing c until this is no longer the case; the size of the max clique is then equal to c
- Therefore, if we can determine the likelihood of the ML solution to the haplotype phasing problem in polynomial time, then we can determine the size of the max clique in polynomial time
- Since finding the size of the max clique of a graph is known to be NP-hard, then finding the ML solution must be NP-hard else P=NP.