2005

Graph algorithms for the haplotyping problem

Yunkai Liu
West Virginia University

Graph Algorithms for the Haplotyping Problem

Yunkai Liu

Dissertation submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Science

Elaine M. Eschen, Ph.D., Chair[1]
Cun-Quan Zhang, Ph.D., Co-chair[2]
Donald A. Adjeroh, Ph.D.[1]
E. James Harner, Ph.D.[3]
Frances L. Van Scoy, Ph.D.[1]
Keqiang Wu, Ph.D.[4]

Morgantown, West Virginia,
2005

Keywords: Haplotype Inference Problem, Perfect Phylogeny Haplotyping Problem, Single Nucleotide Polymorphism, Directed Graph, Poset.

[1] Lane Department of Computer Science and Electrical Engineering, West Virginia University.
[2] Department of Mathematics, West Virginia University.
[3] Department of Statistics, West Virginia University.
[4] Department of Biology, West Virginia University.

Abstract

Graph Algorithms for the Haplotyping Problem

Yunkai Liu

Evidence from investigations of genetic differences among human beings shows that genetic diseases are often the result of genetic mutations. The most common form of these mutations is single nucleotide polymorphism (SNP). A complete map of all SNPs in the human genome will be extremely valuable for studying the relationships between specific haplotypes and specific genetic diseases. Some recent discoveries [31] [12] [40] [17] show that the DNA sequence of human beings can be partitioned into long blocks where genetic recombination has been rare. Then, inferring both haplotypes from chromosome sequences is a biologically meaningful research topic, which has compounded mathematical and computational problems.

We are interested in the algorithmic implications to infer haplotypes from long blocks of DNA that have not undergone recombination in populations. The assumption justifies a model of haplotype evolution - haplotypes in a population evolves along a coalescent, based on the standard population-genetic assumption of infinite sites, which as a rooted tree is a perfect phylogeny. The Perfect Phylogeny Haplotyping (PPH) Problem was introduced by Daniel Gusfield in 2002. A nearly linear-time solution to the PPH problem ($O(nm\alpha(nm))$, where α is the extremely slowly growing inverse Ackerman function) is provided in [25]. However, it is very complex and difficult to implement. So far, even the best practical solution to the PPH problem [4] has the worst-case running time of $O(nm^2)$. D. Gusfield conjectured that a linear-time ($O(nm)$) solution to the PPH problem should be possible [25].

We solve the conjecture of Gusfield by introducing a linear-time algorithm for the PPH problem [35] [36]. Different kinds of posets for haplotype matrices and genotype matrices are designed and the relationships between them are studied. Since redundant calculations can be avoided by the transitivity of partial ordering in posets, we design a linear-time ($O(nm)$) algorithm for the PPH problem that provides all the possible solutions from an input. The algorithm is fully implemented and the simulation shows that it is much faster than previous methods.

Table of Contents

Chapter 1

Introduction

1.1 Biology Concepts

Genetics is an important research field in modern biology. Similar with other scientific disciplines, genetics has concluded a large amount of important dogmas on various of species in last two hundreds years. In 1860s, an Augustinian monk, Gregor Mendel, performed a series of experiments and discover the basic inheritance units, which is named as *genes* now. Since then, studies and researches in genetics have established a set of principles and experimental or analytical procedures, which have greatly contributed to different disciplines in modern biology.

A biology textbook, titled "Introduction to Genetic Analysis", introduced basic genetic concepts as follows. "A gene is a section of a threadlike double-helical molecule called **deoxyribonucleic acid**, abbreviated **DNA**. Genes dictate the inherent properties of a species. The products of most genes are specific **proteins**. Any one gene

[1] The purpose of this chapter is to introduce basic biological concepts and models that the inspired this research. In order to give readers a complete and accurate understanding of backgrounds of the whole research, parts of content (such as definitions and commonly agreed opinions) in this chapter were cited from classic textbooks and papers in case of any incorrect interpretations. Most of ideas in Chapter 1.1, 1.2 and 1.3 are not the author's original work. Readers can study further details in those publications cited.

can exist in several forms that differ from one another, generally in small ways. These forms of a gene are called **alleles**. Allelic variation causes hereditary variation within a species."[20]

If a gene is taken as an individual unit in genetics, the study of big picture of all units is another essential research topic. "An organism's basic complement of DNA is called its **genome**. The somatic cells of most plants and animals contain two copies of their genome; these organisms are **diploid**. The cells of most fungi, algae, and bacteria contain just one copy of the genome; these organisms are **haploid**. The genome itself is made up of one or more extremely long molecules of DNA that are organized into chromosomes. Genes are simply the regions of chromosomal DNA that are involved in the cell's production of proteins. Each chromosome in the genome carries a different array of genes. In diploid cells, each chromosome and its component genes are present twice. For example, human somatic cells contains two sets of 23 chromosomes, for total of 46 chromosomes. Two chromosomes with the same gene array are said to be **homologous**."[20]

The main point of genetics is to understand the interactions of the biology system at the gene level. "When cells divide, the chromosome must also make copies of themselves (replicate) to maintain the appropriate chromosome number in the descendant cells. In eukaryotes, the chromosomes replicate in two main types of nuclear divisions, called *mitosis* and *meiosis*. **Mitosis** is the nuclear division that results in two daughter nuclei whose genetic material is identical with that of the original nucleus. Mitosis can take place in diploid or haploid cells during asexual cell division. **Meiosis** is the general name given to *two* successive nuclear divisions called *meiosis I* and *meiosis II*. Meiosis takes place in special diploid cells called **meiocytes**. Because of the two successive divisions, each meiocyte cell gives rise to four cells: 1 cell → 2 cells → 4 cells. In animal and plants, the products of meiosis become the haploid **gametes**."[20]

Individuals in one specie may show different phenotypes. Discrete, discontinuous difference for one character are because of the alleles of one gene. What we mean is an allele maps to one phenotype and another one maps to the other phenotype. We can observe that those discontinuous phenotypes are in standard patterns of inheritance among generations. The "pattern" can be formulated by precise, specific ratios of individuals with each phenotype. That is also the "pattern" inspire the research of Gregor Mendel and cause the discovery of genes [20]. "Mendel's hypothesis contained not only the notion that genes account for discrete phenotypic difference, but also a mechanism of inheritance of these discrete differences. The essence of Mendel's thesis was that genes come in pairs; these segregate equally into the gametes, which come to contain one of each pair (**Mendel's first law**); and gene pairs on different chromosome pairs assort independently (a modern statement of **Mendel's second law**)."[20]

An American geneticist, Thomas Hunt Morgan, claimed that the two genes were located on the same pair of homologous chromosomes after a dihybrid testcross in *Drosophila*. That is called **linkage**. Based on Morgan's hypothesis, it is easy to understand that the reason for allele combinations from the parental generations remain together, is that they are physically attached by the segment of chromosome between them [20]. Morgan also guessed that " when homologous chromosomes pair in meiosis, the chromosomes occasionally break and exchange parts in a process called

crossing-over. The production of new allele combinations is formally called **recombination**. Crossovers are one mechanism for recombination, and so is independent assortment. Recombination is observed in a variety of biological situations, but it is related to meiosis in most of the cases. Some positions in DNA are occupied by a different nucleotide in different homologous chromosomes. These difference are called **single nucleotide polymorphisms**, or **SNPs** (pronounced "snips"). A SNP is a ubiquitous form of genetic variation in the nucleotide at a single position and SNPs are the most frequent forms of human genetic mutations."[20]

1.2 Introduction to the Haplotype Inferring Problem

A draft and complete map of human DNA sequence was finished in 2001 [1] [43]. Because of that, one of research topic in genetics, genomic research, is greatly boosted. One main objective of the fields is to a question"how similar among all mutations in the human population", because people have already assumed that some genetic diseases are sometimes mapping with the results of genetic mutations.

In diploid organisms, such as human, there are two copies for each chromosome. One copy of those two is called a *haplotype*. The general DNA sequence, which actually is the mixture of the two copies is named as a *genotype*. In medical study, complex diseases are generally affected by more than a single gene. So the knowledge of haplotype data is more important genotype data in drug discovery. The most important factor that decides the information of a haplotype is the SNPs in that region. "A SNP is a single nucleotide site where exactly two (of four) different nucleotides occur in a large percentage of the population "[38] [20]. People have already realized the importance of SNPs. National Institute of Health has already constructed a SNP map for different species to show the density of SNPs per thousand nucleotides. However, with the current technology, directly extracting haplotype information is still difficult or too expensive, though it is extremely valuable. Compare to haplotypes, genotype sequences are much easier to get.

In order to study haplotype more efficiently, a set of DNA sequences were generally considered as m sites (SNPs) in n individuals. Based on the property of SNP, each site have one of two possible states (alleles), which can be took as 0 or 1. For every two rows (one individual), the combinations of two states actually are the haplotypes.

More abstractly in mathematics, "input to the haplotyping problem consists of n genotype vectors, each of length m, where each value in the vector is either 0, 1, or 2. Each position in a vector is associated with a site of interest on the chromosome. The position in the genotype vector has a value of 0 or 1 if the associated chromosome site has that state on both copies (a *homozygous* site), and has a value of 2 otherwise (a *heterozygous* site)."[23]

As we stated above, simple screening technology only can extract the genotype ($2m$ states) of the individual, but can not get the two haplotypes of that individual. So it is important to uses computational methods to extract haplotype information from the given genotype information [25] [27]. Previous methods [10] [11] [18] [24] [37] [33] [34] are mostly statistical approaches. None of are presently fully satisfactory, Although some of them are impressively accurate, none satisfactory models were given to describe the process. Gusfield et. al [25] [4] [23] [9] [8] provides a deterministic and combinatorial approach on this question.

First of all, the biological problem need to be changed into a mathematical problem. “Given an input set of n genotype vectors, a solution to the **Haplotype Inference (HI) Problem** is a set of n pairs of binary vectors, one pair for each genotype vector. For any genotype vector g, the associated binary vectors v_1; v_2 must both have value 0 (or 1) at any position where g has value 0 (or 1); but for any position where g has value 2, exactly one of v_1; v_2 must have value 0, while the other has value 1. That is, v_1, v_2 must be a feasible “explanation” for the true (but unknown) haplotype pair that gave rise to the observed genotype g. Hence, for an individual with h heterozygous sites there are 2^{h-1} haplotype pairs that could appear in a solution to the HI problem. For example, if the observed genotype g is 0212, then the pair of vectors 0110, 0011 is one feasible explanation, out of two feasible explanations. Of course, we want to find the explanation that actually gave rise to g, and a solution for the HI problem for the genotype data of all the n individuals. However, without additional biological insight, one cannot know which of the exponential number of solutions is the “correct one”.”[25]

1.3 Introduction to the Perfect Phylogeny Haplotype Problem

Recently, some new discoveries on population genetics [31] [12] [40] [17] were made. Gusfield conclude the rules [25] as follows.

- “a human chromosome can be partitioned into long blocks where no (or few) recombination occurs”, and
- “the SNPs in each block induce a few common haplotypes in the majority of the population, even though the theoretical number of different haplotypes for

a block is exponential."

Based on the rules above, it is possible to transfer the biology problem, inferring haplotypes from chromosome sequence, into mathematical and computational problems. Also the HI problem will be biological meaningful based on the mathematical model.

A coalescent model, which is a rooted tree that matches with the tracks of a set of haplotypes from sampled individuals during evolution, was proposed [32] [41]. Furthermore, a assumption was taken based on the fact that " in the absence of recombination, each sequence has a single ancestor in the previous generation." [32]. "That is, if we follow backwards in time the history of a single haplotype H from a given individual I, when there is no recombination, that haplotype H is a copy of one of the haplotypes in one of the parents of individual I. It doesn't matter that I had two parents, or that each parent had two haplotypes. The backwards history of a single haplotype in a single individual is a simple path, if there is no recombination. That means that the history of a set of $2n$ individuals, if we look at one haplotype per individual, forms a tree. The histories of two sampled haplotypes (looking backwards in time) from two individuals merge at the most recent common ancestor of those two individuals." [25]

From the mathematics perspective, another important assumption in coalescent model is the infinite sites. "That is, the m sites in the sequence (SNP sites in our case) are so sparse relative to the mutation rate, that in the time frame of interest at most one mutation (change of state) will have occurred at any site. Hence the coalescent model of haplotype evolution says that without recombination, the true evolutionary history of $2n$ haplotypes, one from each of $2n$ individuals, can be displayed as a tree with $2n$ leaves, and where each site labels exactly one edge of the tree, i.e., at a point in history where a mutation occurred at that site. This is the underlying genetic model that we assume from here on." [25] Generally, we assume that we already know the ancestor and put that as a all zero array. It will make the question easier and the problem without knowing the ancestor can be transferred from the simple case. So the no-recombination and infinite sites model says that the $2n$ haplotype (binary) sequences can be explained by a perfect phylogeny [23] [22] which is defined as follows.

Definition: Let M be an $(n \times m)$ binary matrix. Without loss generality, we assume that M contains no repeat rows. Let $\vec{v} = \{v_1, \cdots, v_m\}$ be an m-length binary vector, called **the ancestor vector**. A directed tree T with root $\vec{v}$ is a **perfect phylogeny** for M with $\vec{v}$ as the ancestor if all the following properties are satisfied.

- Each leaf of T is labelled by one row of M and each of the n rows labels exactly one leaf or one internal node of T
- Each of the m columns labels exactly one edge of T.
- Every interior edge (one not incident on a leaf) of T is labelled by at least one column.
- For any row i, the value $M[i][j]$ is unequal to v_j if and only if j labels an edge on the unique path from the root to the leaf labelled i. Hence, that path is a compact representation of row i.

In this report, we only study the perfect phylogeny with the fixed ancestor vector $\vec{v} = \vec{0}$.

$$M = \begin{bmatrix} 1&0&0&0&0 \\ 1&0&1&0&0 \\ 0&0&0&1&0 \\ 0&1&0&1&0 \\ 0&1&0&1&1 \end{bmatrix}$$

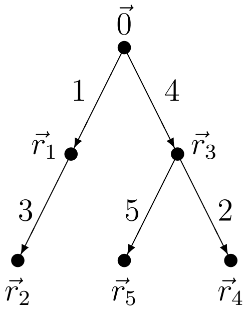

Figure 1.1: Example: M fits a perfect phylogeny. Note, $\vec{r}_i$ means the i-th row of M.

The biological interpretation is that an edge label j indicates the point in time where a mutation at site j occurred, and so the state of site j changes from its ancestral value to the opposite value.

Formally, the **Perfect Phylogeny Haplotype (PPH) Problem** is [25]:

Given a set of genotypes, infer a set of haplotypes that fits a perfect phylogeny, or tell that it is not possible.

Clark et. al. [33] first introduced the PPH problem and provided a solution that is based on the graph realization problem. The running time of the reduction part in the approach is $O(nm\alpha(nm))$. The α is inverse Ackerman function that increases very slow and is usually taken as a constant. Hence, the worst case time for the method is nearly linear. A simplified reduction was provided by Daniel Gusfield [25] later. In Gusfield's method, the time for the reduction is $O(nm)$, and the graph realization problem was solved by several published methods. One method in [6] runs in $O(nm\alpha(nm))$ time. but it was to be too complex to implement. Another method, GPPH [9], used a different solution to the graph realization problem with running time $O(nm^2)$. The other two solutions to the PPH problem, named DPPH [3]and HPPH [15], were also published with worst-case running time of $O(nm^2)$. In [25], D.

Gusfield conjectured that a linear-time ($O(nm)$) solution to the PPH problem should be possible.

1.4 Our Results

We solve the conjecture of D. Gusfield by introducing a linear-time algorithm for the PPH problem. We define several different posets for haplotype matrices and genotype matrices. After studying the relationship between them, we provide an alternative characterization of the PPH problem. Since redundant calculations can be avoided by the transitivity of partial ordering in posets, we design a linear-time ($O(nm)$) algorithm for the PPH problem that can provide all the possible solutions from an input. The algorithm is easy to program and fully implemented. Compared to some existing program, the test shows that our algorithm is much faster than previous methods in practice as well as in theory.

Chapter 2

Notation and Terminology

2.1 About Matrices

Let M be an $(n \times m)$-matrix. We use vector $\vec{r_i}$ to denote the i-th row of M, and vector $\vec{c_j}$ to denote the j-th column of M.

Definition 2.1.1. *Let $\vec{v} = \{v_1, v_2, \cdots, v_n\}$ be a vector. The **support** of $\vec{v}$, denoted as supp($\vec{v}$), is the set $\{i : v_i \neq 0\}$.*

Let set $X \subseteq \{1, \cdots, n\}$ and set $Y \subseteq \{1, \cdots, m\}$. Then, the submatrix that consists of elements of $M[i][j]$, where $i \in X$ and $j \in Y$ is denoted as $M[X][Y]$. For example,

$$\vec{r_i} = M[i][*] = M[i][\{1, \cdots, m\}]$$

and

$$\vec{c_j} = M[*][j] = M[\{1, \cdots, n\}][j].$$

Definition 2.1.2. *Let M be a $\{0, 1, 2\}$-matrix. We define $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ as a 2-**tuple** in* *which x_i is the number of 1's of $\vec{c_i}$ and y_i is the number of 2's of $\vec{c_i}$.* $\begin{pmatrix} x_i \\ y_i \end{pmatrix} \geq \begin{pmatrix} x_j \\ y_j \end{pmatrix}$ *if $x_i > x_j$, or $x_i = x_j$ and $y_i \geq y_j$.*

Definition 2.1.3. *Let M be a $\{0, 1, 2\}$-matrix. M is* ***in column-descending structure*** *if* $\begin{pmatrix} x_i \\ y_i \end{pmatrix} \geq \begin{pmatrix} x_j \\ y_j \end{pmatrix}$ *whenever $1 \leq i \leq j \leq m$.*

Note that, *column-descending structure* also can be applied into binary matrices, in which y-values in those 2-tuples are 0.

2.2 Poset, Hasse Diagram and Antichain

Definition 2.2.1. *Let V be a set of vertices. An ordering "$\geq$" defined on $V \times V$ is a* ***partial ordering*** *if and only if it satisfies the properties as follows.*

- *$a \geq a$, for each $a \in V$ (**reflexivity**).*
- *$a \geq b$ and $b \geq a$ if and only if $a = b$, for each $a, b \in V$ (**antisymmetry**).*
- *if $a \geq b$ and $b \geq c$ then $a \geq c$, for each $a, b, c \in V$ (**transitivity**).*

*A set of vertices V associated with a partial ordering "$\geq$" is called a **poset**, denoted as $P = (V, \geq)$.*

Definition 2.2.2. *Let $P = (V, \geq)$ be a poset. Let a, b be two vertices in V. Then,*

- *a **dominates** b if $a \geq b$;*
- *$a > b$ if $a \geq b$ and $a \neq b$;*

- *and a **covers** b , if $a > b$ and there is no vertex $c \in V - \{a, b\}$ such that $a > c > b$.*

- *If $a \geq b$ or $b \geq a$, we say a and b are **comparable**.*

- *If $a \geq b$, a is an **ancestor** of b and b is a **descendant** of a.*

- *And a is a **parent** of b and b is a **child** of a, if a covers b.*

Definition 2.2.3. *Let $P_1 = (V_1, \geq_1)$ and $P_2 = (V_2, \geq_2)$ be two posets. $P_1 \subseteq P_2$ if $V_1 \subseteq V_2$ and for every pair of vertices $a, b \in V_1$, $a \geq_2 b$ whenever $a \geq_1 b$. It is also called that, P_1 is a **subposet** of P_2 and P_2 is a **supposet** of P_1.*

Definition 2.2.4. *Let $P = (V, \geq)$ be a poset. The **Hasse diagram** of P is a directed acyclic graph with vertex set V and arc set A such that $a \rightarrow b$ if and only if a covers b.*

Lemma 2.2.5. *Given a poset P, the Hasse diagram of P is unique.*

Definition 2.2.6. *Let $D = (V, A)$ be a directed graph. We call a and b as two **ends** of arc if $a \rightarrow b$ in D. For every vertex v in V, we define $N_D^-(v) = \{x : x \rightarrow v\}$ as the **set of in-neighbors** of v, and $N_D^+(v) = \{y : v \rightarrow y\}$ as the **set of out-neighbors** of v. And $d_D^-(v) = |N_D^-(v)|$ is the **indegree** of v and $d_D^+(v) = |N_D^+(v)|$ is the **outdegree** of v.*

By the transitivity property of partial ordering in posets, it is easy to prove the theorem as follows.

Theorem 2.2.7. *Let $P = (V, \geq)$ be a poset. Let $D = (V, A)$ be its Hasse diagram. Let v be a vertex in V such that $d_D^-(v) > 1$ (or $d_D^+(v) > 1$). Then every pair of vertices in $N_D^-(v)$ (or, $N_D^+(v)$ respectively) are not comparable.*

Definition 2.2.8. *Let $P = (V, \geq)$ be a poset. A subset U of V is called an **antichain** of P if a and b are not comparable for each pair of vertices $a, b \in U$. We define the **width** of P as the size of maximum antichain in P.*

Theorem 2.2.9. *Let $P = (V, \geq)$ be a poset and $D = (V, A)$ be its Hasse diagram. If the width of P is k, $d_D^-(v) \leq k$ and $d_D^+(v) \leq k$ for every vertex v in V.*

Definition 2.2.10. *Let $P = (V, \geq)$ be a poset. Let U be a vertex set such that $U \subseteq V$. The subposet of P induced by U is a poset $(U, \geq)$ such that, for any pair of vertices in U, e.g., a and b, $a \geq b$ in $(U, \geq)$ if $a \geq b$ in P.*

Definition 2.2.11. *Let $D = (V, E)$ be a directed graph. Let U be a vertex set such that $U \subseteq V$. The subgraph of D induced by U is a directed graph (U, E') such that, for any pair of vertices in U, e.g., a and b, $a \rightarrow b$ in E' if $a \rightarrow b$ in E.*

Chapter 3

Haplotype Posets

3.1 An Alternative Characterization of the PPH Problem

In this report, the input genotype sequences are generally represented by a $\{0,1,2\}$-matrix (called a **genotype matrix**), in which each row is a genotype sequence. And a binary matrix is a **haplotype matrix** if each row represents a haplotype sequence.

In the HI problem, an $(n \times m)$-genotype matrix M^G is inferred into a $(2n \times m)$ haplotype matrix M^H such that $(2i-1)$-th row and $2i$-th row of M^H generate the i-th row of M^G for every $i \in \{1, \cdots, n\}$. And we say, the haplotype matrix M^H is a **feasible expansion** of the genotype matrix M^G. If M^H is a feasible expansion of M^G and M^H fits a perfect phylogeny, we say M^G is **realizable** and M^H is a **legal expansion** of M^G. Then, the PPH problem is changed as follows.

"*Given a genotype matrix, find its legal expansion - a haplotype matrix that is a feasible expansion of the input and fits a perfect phylogeny.*"

The (3×2)-binary matrix as follows is called the **forbidden matrix** [25].

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The forbidden matrix is an important criteria that establish whether a haplotype matrix fits a perfect phylogeny.

Theorem 3.1.1. *[25] A haplotype matrix fits a perfect phylogeny if and only if it does not have any submatrix that is the forbidden matrix.*

3.2 Definitions of Posets for Haplotype Matrices

Definition 3.2.1. *Let M^H be an $(n \times m)$-haplotype matrix without repeat rows. The* ***haplotype poset constructed by rows*** *of M^H, denoted by $P^H_{row} = (V_{row}, \geq_{row})$, is defined as follows.*

1. *The vertex set V_{row} is a subset of Z_2^m. That is, each vertex of P^H_{row} is a $\{0,1\}$-vector of length m.*

2. *$\{\vec{r_i}$ in $M^H : 1 \leq i \leq n\} \subseteq V_{row}$, and $\vec{0}$ (all zero vector with length m) is a default member of V_{row}.*

3. *For each pair of vertices (vectors) $\vec{a}, \vec{b} \in V_{row}$, $\vec{a} \geq_{row} \vec{b}$ if and only if $supp(\vec{a}) \subseteq supp(\vec{b})$.*

4. *For each pair of vertices $\vec{a}, \vec{b} \in V_{row}$, $\vec{a}$ covers $\vec{b}$ if and only if $supp(\vec{a}) \subseteq supp(\vec{b})$ and $|supp(\vec{b})| - |supp(\vec{a})| = 1$.*

For a haplotype matrix M^H (without repeat rows), there are many different haplotype posets that can be constructed by the definition above. When M^H fits a perfect phylogeny T, all leaves and some internal nodes of T are labelled by rows of M^H. Other internal nodes also can be labelled by the definition of Perfect Phylogeny, so the dominating and covering relation in Definition 3.2.1 can be satisfied. Then the Hasse diagram of a haplotype poset constructed by rows of M^H is isomorphic with the perfect phylogeny T.

Definition 3.2.2. *Let M^H be an $(n \times m)$-haplotype matrix without repeat columns. The* ***haplotype poset constructed by columns*** *of M^H, denoted by $P^H_{col} = (V_{col}, \geq_{col}$), is defined as follows.*

1. *The vertex set V_{col} is a subset of Z_2^n. That is, each vertex of P^H_{col} is a $\{0,1\}$-vector with length n.*

2. *$\{\vec{c}_i$ in M^H: $1 \leq i \leq m\} \cup \{\vec{1}\} = V_{col}$.*

3. *For each pair of vertices (vectors) $\vec{a}, \vec{b} \in V_{col}$, $\vec{a} \geq_{col} \vec{b}$ if and only if $supp(\vec{a}) \supseteq supp(\vec{b})$.*

3.3 Properties of Haplotype Posets

Lemma 3.3.1. *Given a haplotype matrix without repeat columns, the haplotype poset constructed by columns is unique.*

Let M^H be a haplotype matrix without repeat rows and columns. Each column of M^H shows the mutation history of the site. If M^H fits a perfect phylogeny T, the number of columns in M^H should equal to the number of nodes on T. But since only leaves and some internal nodes are labelled by rows of M^H, the number of rows should be not more than the number of columns in M^H. And for every node itself and its descendants in T, those that are labelled by rows of M^H consist of the support of a column. So we can build an injection from a column in M^H to a node on T if M^H fits the perfect phylogeny T.

Note, an all zero column means there is no SNP on the site, therefore it is not in our consideration. And the ancestor sequence is always given as an all zero row in this report.

Theorem 3.3.2. *Let M^H be a haplotype matrix without repeat rows or columns. Assume there is no all zero column in M^H. If M^H fits a perfect phylogeny, then the haplotype poset constructed by columns of M^H is isomorphic with a haplotype poset constructed by rows of M^H.*

$$M = \begin{bmatrix} 1\,0\,0\,0\,0 \\ 1\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,1\,0\,1\,0 \\ 0\,1\,0\,1\,1 \end{bmatrix}$$

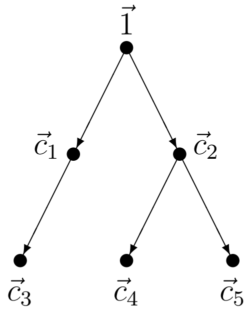

Figure 3.1: An example for Theorem 3.3.2 (which is isomorphic with the tree in Figure 1.1).

By the uniqueness of the haplotype poset constructed by columns and the lemma above, constructing the haplotype poset by columns is more convenient and efficient to the PPH problem than constructing haplotype poset by rows. In the following of this report, all the haplotype posets are constructed by columns of haplotype matrices.

Theorem 3.3.3. *Let M^H be a haplotype matrix and P^H be the haplotype poset for M^H. Let $D^H[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P^H induced by the* *support of μ-th row in M^H. M^H fits a perfect phylogeny if and only if $D^H[supp(\vec{r}_\mu)]$ is a directed path for every $\mu \in \{1, \cdots, n\}$.*

Corollary 3.3.4. *Let M^H be a haplotype matrix and D^H be the Hasse diagram of the haplotype poset for M^H. If M^H fits a perfect phylogeny, then the number of arcs in D^H is up to m.*

Chapter 4

Posets For Genotype Matrices

4.1 Orders Between Columns

Definition 4.1.1. *Let M^G be a genotype matrix and $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G.*

- *If there is a row $\vec{r}_{\mu_1}$ such that both $M^G[\mu_1][i]$ and $M^G[\mu_1][j]$ are nonzero and at least one of them is 1, we say $\vec{r}_{\mu_1}$ is* ***a (1, 1)-row between columns $\vec{c}_i$ and $\vec{c}_j$***.
- *If there is a row $\vec{r}_{\mu_2}$ such that $M^G[\mu_2][i] = M^G[\mu_2][j] = 2$, we say $\vec{r}_{\mu_2}$ is* ***a (2, 2)-row between columns $\vec{c}_i$ and $\vec{c}_j$***.
- *If there is a row $\vec{r}_{\mu_3}$ such that exactly one of $M^G[\mu_3][i]$ and $M^G[\mu_3][j]$ is 0 and the other is nonzero, we say $\vec{r}_{\mu_3}$ is* ***a (0, 1)-row between columns $\vec{c}_i$ and $\vec{c}_j$***.

Of course, there may be some $(0, 0)$-*rows between two columns.* However, since an all zero vector is always given as the ancestor vector, a $(0, 0)$-row exists in each pair of columns. That means $(0, 0)$-rows do not affect the realizability of the genotype matrix and we will not consider them in this report.

Definition 4.1.2. *Let M^G be a genotype matrix. Let $\vec{r}_\mu$ be a $(2,2)$-row between columns $\vec{c}_i$ and $\vec{c}_j$. If $M^G[\mu][i]$ and $M^G[\mu][j]$ are inferred in different ways in a feasible expansion of M^G, e.g., one is inferred as $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and the other is inferred as $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, we say that $M^G[\mu][i]$ and $M^G[\mu][j]$ are* ***in different order****. Else, they are* ***in the same order****.*

Definition 4.1.3. *Let M^G be a genotype matrix and $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G. Then,*

- *$\vec{c}_i$ and $\vec{c}_j$ are* ***in different order****, if $M^G[\mu][i]$ and $M^G[\mu][j]$ are in different order for every $(2,2)$-row $\vec{r}_u$ between $\vec{c}_i$ and $\vec{c}_j$. Note, if two columns are in different order, then there is at least one $(2,2)$-row between them.*

- *$\vec{c}_i$ and $\vec{c}_j$ are* ***in the same order****, if $M^G[\mu][i]$ and $M^G[\mu][j]$ are in the same order for every $(2,2)$-row $\vec{r}_u$ between $\vec{c}_i$ and $\vec{c}_j$. Note, two columns are in the same order even if there is no $(2,2)$-rows between them.*

Theorem 4.1.4. *Let M^G be a genotype matrix. If M^G is realizable, then $\vec{c}_i$ and $\vec{c}_j$ are in the same order or different order for every pair of columns $\vec{c}_i$ and $\vec{c}_j$ in M^G.*

4.2 Definitions of Posets for Genotype Matrices

Let M^G be a genotype matrix. For the elements in M^G, we define ones with value 1 dominate ones with value 2, while ones with value 2 dominate ones with value 0. That is, $1 > 2 > 0$.

We will introduce three different genotype posets as follows.

Definition 4.2.1. *Let M^G be an $(n \times m)$-genotype matrix. A* ***genotype poset*** *$P^G = (V, \geq)$ for M^G is defined as follows.*

- *$V = \{\vec{c}_k : k = 1, \cdots, m\}$. That is, every vertex in V is a column of M^G.*
- *Let $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G. $\vec{c}_i \geq \vec{c}_j$ if $M^G[\mu][i] \geq M^G[\mu][j]$ for every $\mu \in \{1, \cdots, n\}$.*

Definition 4.2.2. *Let M^G be an $(n \times m)$-genotype matrix. A* ***left-prior genotype poset*** *$P_l^G = (V, \geq_l)$ for M^G is defined as follows.*

- *$V = \{\vec{c}_k : k = 1, \cdots, m\}$.*
- *Let $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G. $\vec{c}_i \geq_l \vec{c}_j$ if $i \leq j$ and $M^G[\mu][i] \geq M^G[\mu][j]$ for every $\mu \in \{1, \cdots, n\}$.*

Definition 4.2.3. *Let M^G be an $(n \times m)$-genotype matrix that is inferred to a feasible expansion. An* ***ordered genotype poset*** *$P_o^G = (V, \geq_o)$ for M^G is defined as follows.*

- *$V = \{\vec{c}_k : k = 1, \cdots, m\}$.*
- *Let $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G. $\vec{c}_i \geq_o \vec{c}_j$ if $\vec{c}_i$ and $\vec{c}_j$ are in the same order and $M^G[\mu][i] \geq M^G[\mu][j]$ for every $\mu \in \{1, \cdots, n\}$.*

For the three genotype posets, we use same terms (e.g., “dominate”, “comparable” and “cover”) as we defined in poset in Section 2.2. Since vertices in all of those posets are columns of the genotype matrix, we will use v instead of $\vec{c}$ in the following of this paper and those vertices will have the same index with columns, e.g., v_i means the i-th column in M^G.

4.3 Properties of Genotype Posets

For a genotype matrix M^G, the genotype poset and the left-prior genotype poset are both unique. However, there are many ordered genotype posets possible, since the orders between columns are not unique. In this section, some properties of those posets are introduced as follows.

Lemma 4.3.1. *Let M^G be a genotype matrix and P^G be the genotype poset for M^G. Let v_i and v_j be two vertices in P^G. If v_i and v_j are not comparable in P^G, then there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][i] > M^G[\mu_1][j]$ and $M^G[\mu_2][i] < M^G[\mu_2][j]$.*

Theorem 4.3.2. *Let M^G be a genotype matrix and M^H be a feasible expansion of M^G. If M^H fits a perfect phylogeny, then the haplotype poset for M^H is isomorphic with an ordered genotype poset for M^G.*

Lemma 4.3.3. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns. Let P^G be the genotype poset for M^G. If $v_i \geq v_j$ in P^G, then $i \leq j$, for any $i, j \in \{1, \cdots, m\}$.*

We denote the left-prior genotype poset for $M^G[\{1, \cdots, \mu\}][*]$ as P_l^{μ}. So $P_l^n = P_l^G$. Let P_l^0 be the "*complete*" left-prior genotype poset, in which each column dominates all the columns on its right side. Then, we have an important theorem as follows.

Theorem 4.3.4. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure without repeat rows or columns. Then for M^G,*

$$P_l^0 \supseteq P_l^1 \supseteq \cdots \supseteq P_l^n = P_l^G = P^G \supseteq P_o^G$$

.

4.4 Properties of Posets for Realizable Genotype Matrices

Lemma 4.4.1. *Let M^G be an $(n \times m)$-genotype matrix. Let $\vec{c}_i$ and $\vec{c}_j$ be two columns such that there is a row $\vec{r}_\mu$ such that $M^G[\mu][i] = 1$ and $M^G[\mu][j] \neq 0$. If M^G is realizable, then $v_i \geq_o v_j$ in any ordered genotype poset for M^G.*

Lemma 4.4.2. *Let M^G be an $(n \times m)$-genotype matrix. Let v_i, v_j and v_k be three vertices. Suppose there is a row $\vec{r}_\mu$ such that $M^G[\mu][i] = M^G[\mu][j] = M^G[\mu][k] = 2$. If M^G is realizable, then at least one pair of v_i, v_j and v_k are comparable in an ordered genotype poset for M^G.*

Theorem 4.4.3. *Let M^G be an $(n \times m)$-genotype matrix. Let P_o^G be an ordered genotype poset for M^G. Let $P_o^G[supp(\vec{r}_\mu)]$ be the subposet of P_o^G induced by the support of row $\vec{r}_\mu$. If M^G is realizable, then the width of $P_o^G[supp(\vec{r}_\mu)]$ is at most 2, for each $\mu \in \{1, \cdots, n\}$.*

Corollary 4.4.4. *Let M^G be a genotype matrix. Let P^G be the genotype poset for M^G. Let $P^G[supp(\vec{r}_\mu)]$ be the subposet of P^G induced by the support of row $\vec{r}_\mu$. If M^G is realizable, then the width of $P^G[supp(\vec{r}_\mu)]$ is at most 2, for each $\mu \in \{1, \cdots, n\}$.*

Theorem 4.4.5. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure without all zero columns. Let D_l^G be the Hasse diagram of the left-prior genotype poset for M^G. If M^G is realizable, then no vertex has indegree greater than 2 in D_l^G.*

Note, if M^G is realizable, then the vertex in the Hasse diagram of the left-prior genotype poset may have outdegree greater than 2. That means, the width of the left-prior genotype poset for M^G may be greater than 2.

Lemma 4.4.6. *Let M^G be an $(n \times m)$-genotype matrix. Let P_o^G be an ordered genotype poset for M^G. Let $D_o^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P_o^G induced by the support of row $\vec{r}_\mu$. If M^G is realizable, then $d^+(v) \leq 2$ and $d^-(v) \leq 1$ for any vertex v in $D_o^G[supp(\vec{r}_\mu)]$.*

Lemma 4.4.7. *Let M^G be an $(n \times m)$-genotype matrix. Let P_o^G be an ordered genotype poset for M^G. Let $D_o^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P_o^G induced by the support of row $\vec{r}_\mu$. Assume M^G is realizable. Let v_i be a vertex in $D_o^G[supp(\vec{r}_\mu)]$.*

- *If $d^+(v_i) = 2$ in $D_o^G[supp(\vec{r}_\mu)]$, then $M^G[\mu][i] = 1$, all descendants of v_i in $D_o^G[supp(\vec{r}_\mu)]$ are 2 in row $\vec{r}_\mu$, and all ancestors of v_i in $D_o^G[supp(\vec{r}_\mu)]$ are 1 in row $\vec{r}_\mu$.*
- *And if vertex v_i is 2 in row $\vec{r}_\mu$, both $d^+(v_i)$ and $d^-(v_i)$ are less than or equal to 1.*

Theorem 4.4.8. *Let M^G be an $(n \times m)$-genotype matrix. Let P_o^G be an ordered genotype poset for M^G. Let $D_o^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P_o^G induced by the support of row $\vec{r}_\mu$. M^G is realizable if and only if for every $\mu \in \{1, \cdots, n\}$,*

- *$D_o^G[supp(\vec{r}_\mu)]$ is a rooted directed tree with two leaves, which satisfies that in row $\vec{r}_\mu$,*

  - *the only vertex v with outdegree 2 is 1 in row $\vec{r}_\mu$;*
  - *all ancestors of v are 1 in row $\vec{r}_\mu$;*
  - *and all descendants of v are 2 in row $\vec{r}_\mu$;*

- *or, two vertex-disjointed directed paths, in which all the vertices are 2 in row $\vec{r}_\mu$;*
- *or, one directed path, in which no vertex with value "2" is an ancestor of vertices with value "1" in row $\vec{r}_\mu$.*

Chapter 5

Our Linear Solution to the PPH Problem

5.1 Why Linear Solution to The PPH Problem

Since Gusfield introduced the the Perfect Phylogeny Haplotyping (PPH) Problem [25], a lot of studies have been done to find efficient solutions. The problem was first transferred in the graph realization problem, which has been proved a NP-complete problem. However, with further study, researchers realized that the possibility of a linear method [9], since what the Perfect Phylogeny Haplotyping model constructed is a special tree. In this chapter, we solve the open problem, and give a practical, deterministic linear-time algorithm based on graph theories about posets and Hasse diagrams. The method has been fully implemented and simulations show it is much faster in practice than prior methods.

Our solution to the open problem not only provided a smart data-structure in graph algorithm, but also greatly affected the related biology study. Most of applications so far are only able to handle hundreds of SNPs. The genomic compositions of haplotypes in the populations of human and other species are still unknown. Some recent discovery shows that some genes with high linkage disequilibrium could extend into even hundreds kilobases. So we believe our method make the genomic study on haplotypes to be possible, because of the dramatic improvement on the computational efficiency.

5.2 Brief Description of Main Algorithm

The old solutions to the PPH problem consider the relation between each pair of columns in the input genotype matrix. That is the reason their time-complexities are at least $O(nm^2)$. We notice that, redundant calculations can be avoided by the transitivity property of partial ordering in posets. After permuting the input genotype matrix into a column-descending structure, the genotype poset can be constructed by building the left-prior posets for submatrices of the input (see Theorem 4.3.4). Then, we remove all the arcs in the Hasse diagram of the genotype poset whose two ends are in different order, and construct an ordered genotype poset that is isomorphic with the haplotype poset for a legal expansion of the input. The algorithm is briefly described as follows.

Algorithm 5.2.1.

Input: *a genotype matrix* M^G_{input}.

Procedure:

Step 1: *Repeat rows and columns are removed, and the input matrix is checked and permuted into column-descending structure. The output is denoted as* M^G_{desc}. *For details, see Section 5.3.*

Step 2: *The Hasse diagram of the left-prior genotype poset for* $M^G_{desc}[\{1, \cdots, \mu\}][*]$ $(\mu \in \{1, \cdots, n\})$ *is built (updated) by induction. The output is the Hasse diagram of the genotype poset for* M^G_{desc}*. For details, see Section 5.4.*

Step 3: *Those arcs in the Hasse diagram of the genotype poset, whose ends are in different order, are removed. For details, see Section 5.5*

Step 4: *An ordered genotype poset is constructed. For details, see Section 5.6.*

Step 5: *A haplotype matrix that fits a perfect phylogeny is inferred from* M^G_{desc}*; and columns are permuted back. For details, see Section 5.7.*

Output: *a haplotype matrix that is a legal expansion of* M^G_{input}*, or that* M^G_{input} *is not realizable.*

5.3 To Pre-scan the Input

As we explained before, repeat rows and columns do not affect the realizability of the input. Neither does the all zero column. The all zero rows is default the ancestor vector. So, we need to remove them in the first step of main algorithm to reduce calculations. Column-descending structure does not affect the realizability of the input also, but it will be very helpful for us to use the transitivity property of partial ordering in posets.

Algorithm 5.3.1.

Input: *a genotype matrix* M^G_{input}.

Procedure:

Step 1: *All repeat rows and columns of M^G_{input} are removed; and all zero row and all zero column are removed;*

Step 2: *2-tuples of columns are calculated and sorted, and columns are permuted. M^G_{input} is changed into column-descending structure.*

Step 3: *Each row of the new matrix is checked. If there is a 2 left of a 1 in a row, then M^G_{input} is not realizable (see Lemma 5.3.1).*

Output: *a genotype matrix M^G_{desc}, which has no repeat rows or columns, has no all zero row or all zero column and is in column-descending structure, or that M^G_{input} is not realizable.*

Lemma 5.3.1. *Let M^G be a genotype matrix in column-descending structure. If there is a row $\vec{r}_\mu$ such that $M^G[\mu][i] = 2$, $M^G[\mu][j] = 1$ and $i < j$, then M^G is not realizable.*

5.4 To Construct the Genotype Poset

Suppose we already have M^G_{desc} after pre-scanning the input (see Algorithm 5.3.1). For simplification, we denote M^G_{desc} by M^G in follows.

5.4.1 Brief Idea

We will briefly introduce how to construct the Hasse diagram of the genotype poset for M^G_{desc} in this section.

First, we initialize a "complete" left-prior genotype poset P^0_l (defined in Section 4.2). In D^0_l, which is the Hasse diagram of P^0_l, $v_i \rightarrow v_{i+1}$ for every $i \in \{1, \cdots, m-1\}$.

Suppose we already have the Hasse diagram of $P_l^{\mu-1}$ (denoted as $D_l^{\mu-1}$), $\mu \in \{2, \cdots, n\}$. Let v_i and v_j be two vertices. If $M^G[\mu][i] = 0$, $M^G[\mu][j] \neq 0$ and $v_i \rightarrow v_j$ in $D_l^{\mu-1}$, then

- we add arc(s) $v_{i'} \rightarrow v_j$, when $M^G[\mu][i'] \neq 0$, $v_{i'}$ is an ancestor of v_i, and vertices on any path between $v_{i'}$ and v_j in $D_l^{\mu-1}$ are 0 in row $\vec{r}_\mu$;
- we add arc(s) $v_i \rightarrow v_{j'}$, when $M^G[\mu][j'] = 0$, $v_{j'}$ is a descendant of v_j, and vertices on any path between v_i and $v_{j'}$ in $D_l^{\mu-1}$ are non-zero in row $\vec{r}_\mu$;
- we delete arc $v_i \rightarrow v_j$.

The output is D_l^μ, which is the Hasse diagram of P_l^μ. By induction, we can finally get D_l^n that is also the Hasse diagram of the genotype poset for M_{desc}^G. Note, if M_{desc}^G is realizable, then by Theorem 4.4.5, for every v_i and v_j, there are at most two $v_{i'}$s but many (up to $m-3$) $v_{j'}$s.

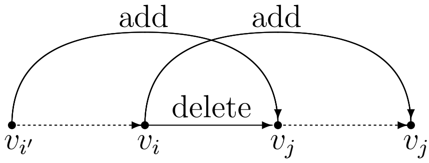

Figure 5.1: Brief idea of updating the left-prior genotype posets. Note, the dash lines mean dominating relation, and solid lines mean covering relation.

By the description above, $v_i \not>_l v_j$ in P_l^μ, since $M_{desc}^G[\mu][i] < M_{desc}^G[\mu][j]$. But $M_{desc}^G[\mu][i'] \geq M_{desc}^G[\mu][j]$, because $M_{desc}^G[\mu][i']$ and $M_{desc}^G[mu][j]$ are non-zero and there is no 2 left of 1 in any row of M_{desc}^G. Then $v_{i'} >_l v_j$ in P_l^μ. And $v_i >_l v_{j'}$ in P_l^μ.

Since all internal vertices on paths between $v_{i'}$ and v_j in $D_l^{\mu-1}$ are 0 in row $\vec{r}_\mu$, they do not dominate v_j in P_l^μ. Then, there is no vertex v between $v_{i'}$ and v_j such that $v_{i'} > v > v_j$ in P_l^μ. So $v_{i'}$ covers v_j in P_l^μ. And $v_{i'} \rightarrow v_j$ in D_l^μ. With the same reason, v_i does not dominate any vertex on paths between v_i and $v_{j'}$ in $D_l^{\mu-1}$. Then v_i covers $v_{j'}$ in P_l^{μ} and $v_i \rightarrow v_{j'}$ in D_l^{μ}.

However, finding out all the $v_{i'}$ and $v_{j'}$ for every $v_i \rightarrow v_j$ can be very complicated. So, more detailed analysis is to be presented as follows.

5.4.2 λ Function

Definition 5.4.1. *Let M^G be an $(n \times m)$-genotype matrix. For each column $\vec{c}_i$ ($i \in \{1, \cdots, m\}$) in M^G, we define $\lambda(\vec{c}_i)$ recursively as follows.*

- $\lambda(\vec{c}_i) = 1$ *if there is at least one $\mu_1 \in \{1, \cdots, n\}$ such that $M^G[\mu_1][i] = 1$.*
- $\lambda(\vec{c}_i) = 2$ *if $\lambda(\vec{c}_i) \neq 1$ and there is at least one $\mu_2 \in \{1, \cdots, n\}$ such that $M^G[\mu_2][i] = 2$.*
- *Otherwise $\lambda(\vec{c}_i) = 0$. That means i-th column of M^G is an all zero column.*

Since vertex v_i in any genotype poset is the i-th column of the genotype matrix, $\lambda(v_i) = \lambda(\vec{c}_i)$ and it is called as *λ-value of vertex v_i*. Note, M^G_{desc} has no all zero column (see Algorithm 5.3.1). But all zero columns may exist in $M^G_{desc}[1, \cdots, \mu][*]$ when $\mu < n$.

By the definition of λ-function, it is easy to prove the following lemma.

Lemma 5.4.2. *Let M^G be an $(n \times m)$-genotype matrix and D_l^G be the Hasse diagram of the left-prior genotype poset for M^G. If v_i is a vertex such that $\lambda(v_i) \neq 0$, then every ancestor of v_i in D_l^G has non-zero λ-value in M^G.*

Theorem 5.4.3. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and D_l^G be the Hasse diagram of the left-prior genotype poset for M^G. Assume M^G is realizable.*

- $d^-(v_i) \leq 1$ *for every vertex* v_i *in* D_l^G *such that* $\lambda(v_i) = 1$ *in* M^G.
- $d^-(v_i) \leq 2$ *for every vertex* v_i *in* D_l^G *such that* $\lambda(v_i) = 2$ *in* M^G.
- *If* v_i *is a vertex such that* $\lambda(v_i) = 2$ *in* M^G *and* $d^-(v_i) = 2$ *in* D_l^G *(suppose* $N^-(v_i) = \{v_j, v_k\}$*), then* $M^G[\mu][j] = M^G[\mu][k] = 2$ *for each row* $\vec{r}_\mu$ *such that* $M^G[\mu][i] = 2$.

Corollary 5.4.4. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure. Let D_l^μ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu\}][*]$, $\mu \in \{1, \cdots, n\}$. Let v_i be a vertex in D_l^μ. If M^G is realizable, and*

- $\lambda(v_i) = 1$ *in* $M^G[\{1, \cdots, \mu\}][*]$*, then* $d^-(v_i) \leq 1$ *in* D_l^μ*, for every* $\mu \in \{1, \cdots, n\}$;
- $\lambda(v_i) = 2$ *in* $M^G[\{1, \cdots, \mu\}][*]$*, then* $d^-(v_i) \leq 2$ *in* D_l^μ*, for every* $\mu \in \{1, \cdots, n\}$;

5.4.3 Bad-zeros and Bad-ones

Definition 5.4.5. *Let M^G be an $(n \times m)$-genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu - 1\}][*]$, $\mu \in \{2, \cdots, n\}$. We define vertex v_i as a **bad-zero** in row $\vec{r}_\mu$ if $\lambda(v_i) \neq 0$ in $M^G[\{1, \cdots, \mu - 1\}][*]$, $M^G[\mu][i] = 0$ and at least one descendant of v_i in $D_l^{\mu-1}$ is non-zero in row $\vec{\mu}$. And a vertex v_j is called a **bad-one** in row $\vec{r}_\mu$ if $\lambda(v_j) \neq 0$ in $M^G[\{1, \cdots, \mu - 1\}][*]$, $M^G[\mu][j] \neq 0$ and at least one ancestor of v_j in $D_l^{\mu-1}$ is 0 in row $\vec{r}_\mu$.*

Definition 5.4.6. *We define those vertices that are* 0 *but not bad-zeros in row* $\vec{r}_\mu$ *as* ***good zeros*** *in row* $\vec{r}_\mu$*, and those that are non-zero but not bad-ones in row* $\vec{r}_\mu$ *as* ***good ones*** *in row* $\vec{r}_\mu$*.*

Note, since there is no 2 left of 1 in any row of M^G_{desc}, if $M^G[\mu][i] < M^G[\mu][j]$ then $M^G[\mu][i] = 0$ and $M^G[\mu][j] \neq 0$.

Lemma 5.4.7. *Let* M^G *be an* $(n \times m)$*-genotype matrix in column-descending structure. Let* P_l^μ *be the left-prior genotype poset for* $M^G[\{1, \cdots, \mu{-}1\}][*]$ $(\mu \in \{1, \cdots, n\})$*, and* P^G *be the genotype poset for* M^G*. If there is a* $\mu \in \{1, \cdots, n\}$ *such that* v_i *and* v_j *are not comparable in* P_l^μ*, then* v_i *and* v_j *are not comparable in* P^G*.*

Lemma 5.4.8. *Let* M^G *be an* $(n{\times}m)$*-genotype matrix in column-descending structure and* $D_l^{\mu-1}$ *be the Hasse diagram of the left-prior genotype poset for* $M^G[\{1, \cdots, \mu - 1\}][*]$*,* $\mu \in \{2, \cdots, n\}$*. Let* v_i*,* v_j *and* v_k *be three vertices, such that* $v_j \rightarrow v_i$ *and* $v_k \rightarrow v_i$ *in* $D_l^{\mu-1}$ *and* $\lambda(v_i) \neq 0$ *in* $M^G[\{1, \cdots, \mu{-}1\}][*]$*. If* $M^G[\mu][j] = M^G[\mu][k] = 0$ *and* $M^G[\mu][i] \neq 0$*, then* M^G *is not realizable.*

Theorem 5.4.9. *Let* M^G *be an* $(n{\times}m)$*-genotype matrix in column-descending structure and* $D_l^{\mu-1}$ *be the Hasse diagram of the left-prior genotype poset for* $M^G[\{1, \cdots, \mu{-}1\}][*]$*,* $\mu \in \{2, \cdots, n\}$*. Let* v_i *be a bad-one in row* $\vec{r}_\mu$*. If* v_i *has two ancestors in* $D_l^{\mu-1}$ *such that they are bad-zeros in row* $\vec{r}_\mu$ *and are not comparable, then* M^G *is not realizable.*

The proof for Theorem 5.4.9 is similar to that for Lemma 5.4.8. Thus we can easily get the following corollary.

Corollary 5.4.10. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v_i be a bad-one in row $\vec{r}_\mu$. If M^G is realizable, then all the ancestors of v_i that are bad-zeros in row $\vec{r}_\mu$ is on a direct path.*

Lemma 5.4.11. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v_h be a vertex such that $\lambda(v_h) \neq 0$ in $M^G[\{1, \cdots, \mu-1\}][*]$. Let v_i, v_j and v_k be three vertices such that $v_i \longrightarrow v_j$, $v_i \longrightarrow v_k$, $v_j \longrightarrow v_h$ and $v_k \longrightarrow v_h$ in $D_l^{\mu-1}$ and $\lambda(v_i) \neq 0$. If $M^G[\mu][j]$ and $M^G[\mu][k]$ are non-zero and $M^G[\mu][i] = 0$, then M^G is not realizable.*

Theorem 5.4.12. *Let M^G be an $(n\times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v_h be a vertex whose λ-value in $M^G[\{1, \cdots, \mu-1\}][*]$ is non-zero. Suppose v_h has two ancestors (e.g., v_j and v_k) in $D_l^{\mu-1}$ that are both bad-ones in row $\vec{r}_\mu$ and not comparable with each other. If v_j and v_k have a common ancestor that is a bad-zero in $\vec{r}_\mu$, then M^G is not realizable.*

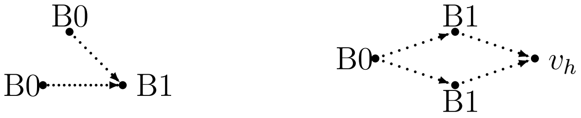

Figure 5.2: M^G is not realizable in both cases above. Note, dash-lines mean dominating relation; "B1" means a vertex that is bad-one in row $\vec{r}_\mu$; and "B0" means a vertex that is bad-zero in row $\vec{r}_\mu$. $\lambda(v_h) \neq 0$ in $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. For detail, see Theorems 5.4.9 and 5.4.12.

Corollary 5.4.13. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v_i be a bad-one in row $\vec{r}_\mu$. If M^G is realizable, then all ancestors of v_i in $D_l^{\mu-1}$ that are bad-ones are on one directed path.*

Theorem 5.4.14. *Let M^G be an $(n\times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let $D_l^{\mu-1}(B0(\vec{r}_\mu))$ be the Hasse diagram of the subposet of $P_l^{\mu-1}$ induced by bad-zeros in row $\vec{r}_\mu$. Let $D_l^{\mu-1}(B1(\vec{r}_\mu))$ be the Hasse diagram of the subposet of $P_l^{\mu-1}$ induced by bad-ones in row $\vec{r}_\mu$. Then $D_l^{\mu-1}(B0(\vec{r}_\mu))$ and $D_l^{\mu-1}(B1(\vec{r}_\mu))$ are in one of the structures as follows.*

- *A rooted tree with only two leaves, on which only one vertex has outdegree 2 and indegree at most 1, and other vertices have indegree and outdegree less or equal to* 1*;*
- *two vertex-disjoined paths;*
- *or one path.*

Definition 5.4.15. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. For every vertex v_i, we define its* ***branch index*** *as follows.*

- *When $j \in index_0(v_i)$, v_i has a descendant v_j in $D_l^{\mu-1}$ that is a bad-zero in row $\vec{r}_\mu$.*

- *When $k \in index_1(v_i)$, v_i has a descendant v_k in $D_l^{\mu-1}$ that is a bad-one in row $\vec{r}_\mu$.*

For any vertex v in $D_l^{\mu-1}$, $index_0(v) = \bigcup_{v_i \in N^+(v)}(index_0(v_i))$ and $index_1(v) = \bigcup_{v_i \in N^+(v)}$ $(index_1(v_i))$.

*For any two vertices v_i and v_j, we say they are **on the same branch** if $index_0(v_i) \supseteq index_0(v_j)$ or $index_0(v_i) \subseteq index_0(v_j)$ or $index_1(v_i) \supseteq index_1(v_j)$ or $index_1(v_i) \subseteq index_1(v_j)$.*

Note, for a vertex v, we generally use the location of its rightmost descendant in $D_l^{\mu-1}$ that is also a bad-zero (or bad-one) as its branch index. If there is a vertex v such that $|index_0(v)| > 2$ or $|index_1(v)| > 2$, then by Theorem 5.4.14, M^G is not realizable.

Corollary 5.4.16. *Let M^G be an $(n \times m)$-genotype matrix in column-descending structure and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v_i be a vertex. Suppose every bad-zeros (or bad-ones) in row $\vec{r}_\mu$ satisfies Theorem 5.4.14. If v_i is a bad-zero (or a bad-one) in row $\vec{r}_\mu$ and v_j is the closest ancestor of v_i that is also a bad-zero (respectively a bad-one), then there is an arc $v_j \rightarrow v_i$ in D_l^μ.*

Algorithm 5.4.1.

Step 1: *All the bad zeros and bad-ones in row $\vec{r}_\mu$ are found by searching $D_l^{\mu-1}$ from right to left.*

Step 2: *The branch index of each vertex is marked by searching $\vec{r}_\mu$ from right to left (see Definition 5.4.15). Note, generally we use the location of the rightmost bad-zero (or bad-one) that has not been marked during searching as the marker.*

Step 3: *The $index_0$ and $index_1$ of each site are checked to make sure that their size is less than or equal to 2.*

Step 3: *Searching $D_l^{\mu-1}$ to check if all the bad-zeros and bad-ones in row $\vec{r}_\mu$ satisfy Theorem 5.4.14 and add new arcs (see Corollary 5.4.16 and Section 5.4.1).*

1. *Bad-zeros (bad-ones) that are in one path are picked out by comparing their indices (values of $index_0$ function or $index_1$ function) and searching from left to right.*
2. *Among those vertices that are picked out, we check if they are in one path. For example, let v_i and v_j $(i < j)$ be two bad-zeros that are picked out by last step. If there is no v_k that is also picked out and $i < k < j$, we will check if v_i connects to v_j.*
3. *Arcs are added between those vertices that are picked out. For example, if v_i and v_j $(i < j)$ are picked out and there is no k such that $i < k < j$ and v_k is also picked out, then add arc $v_i \longrightarrow v_j$.*

5.4.4 Parent Function and Descendant Function

Definition 5.4.17. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1,\cdots,\mu-1\}][*]$, $\mu \in \{2,\cdots,n\}$. For a vertex v_i such that $\lambda(v_i) \neq 0$ in $M^G[\{1,\cdots,\mu-1\}][*]$, we define the **parent function** of v_i in row $\vec{r}_\mu$ as $parent(v_i) = \{v_j : v_j$ is a bad-zero in row $\vec{r}_\mu$, v_j is an ancestor of v_i in $D_l^{\mu-1}$ and all vertices on the paths between v_j and v_i are non-zero in row $\vec{r}_\mu.\}$.*

Definition 5.4.18. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1,\cdots,\mu-1\}][*]$, $\mu \in \{2,\cdots,n\}$. For a vertex*

v_i such that $\lambda(v_i) \neq 0$ in $M^G[\{1, \cdots, \mu-1\}][*]$, we define the ***descendant function*** *of v_i in row $\vec{r}_\mu$ as $desc(v_i) = \{v_j : v_j$ is a bad-one in row $\vec{r}_\mu$, v_j is a descendant of v_i in $D_l^{\mu-1}$ and all vertices on paths between v_i and v_j are 0 in row $\vec{r}_\mu.\}$.*

By Definitions 5.4.17 and 5.4.18, it is easy to get the lemma as follows.

Lemma 5.4.19. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v be a vertex. If $\{v_i, v_j\} \in parent(v)$ (or $\{v_i, v_j\} \in desc(v)$), then v_i and v_j are not comparable in $P_l^{\mu-1}$.*

By Lemma 5.4.19, we can easily get two corollaries as follows.

Corollary 5.4.20. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. If M^G is realizable, then*

- *$|parent(v)| \leq 1$ for every vertex v in $D_l^{\mu-1}$;*
- *$|desc(v)| \leq 2$ for every vertex v in $D_l^{\mu-1}$.*

Corollary 5.4.21. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu-1\}][*]$, $\mu \in \{2, \cdots, n\}$. Let v be a vertex.*

- *If $parent(v) = \{v_i, v_j\}$ $(i < j)$ and $index_0(v_i) = index_0(v_j)$, then $v_i \not\to v$ in D_l^{μ}.*
- *If $desc(v) = \{v_i, v_j\}$ $(i < j)$ and $index_1(v_i) = index_1(v_j)$, then $v \not\to v_j$ in D_l^{μ}.*

We can prove Corollary 5.4.21 by the transitivity property of partial ordering in posets.

Definition 5.4.22. *For two sets A and B,*

- $A \bigcup B$ *is the set including all different elements from sets A and B. $|A|$ is the size of A.*
- *We define $A \bigcup_0^* B$ as the subsets of $A \bigcup B$. For any pair of vertices v_i and v_j in $A \bigcup B$, if $i \leq j$ and $index_0(v_i) \supseteq index_0(v_j)$, then $v_i \notin A \bigcup_0^* B$.*
- *We define $A \bigcup_1^* B$ as the subsets of $A \bigcup B$. For any pair of vertices v_i and v_j in $A \bigcup B$, if $i \leq j$ and $index_1(v_i) \supseteq index_1(v_j)$, then $v_j \notin A \bigcup_1^* B$.*

Note, if $i = j$, then v_i (or v_j) are not in $A \bigcup_0^* B$ or $A \bigcup_1^* B$.

Theorem 5.4.23. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu - 1\}][*]$, $\mu \in \{2, \cdots, n\}$. For every vertex v_i that is not a bad-zero in row $\vec{r}_\mu$,*

1. *if v_i has only one parent v_j, then $parent(v_i) = parent(v_j)$;*
2. *if $N^-(v_i) = \{v_j, v_k\}$ in $D_l^{\mu-1}$, then $parent(v_i) = parent(v_j) \bigcup_0^* parent(v_k)$;*
3. *if $N^-(v_i) = \{v_j, v_k\}$ in $D_l^{\mu-1}$ such that $M^G[\mu][j]$ and $M^G[\mu][k]$ are both non-zero and $parent(v_j) \bigcap parent(v_k) \neq \emptyset$, then M^G is not realizable;*

Theorem 5.4.24. *Let M^G be a genotype matrix and $D_l^{\mu-1}$ be the Hasse diagram of the left-prior genotype poset for $M^G[\{1, \cdots, \mu - 1\}][*]$, $\mu \in \{2, \cdots, n\}$. For every vertex v_i that is not a bad-one in row $\vec{r}_\mu$,*

1. *if v_i has only one descendant v_j in $D_l^{\mu-1}$, then $desc(v_i) = desc(v_j)$;*
2. *if $N^+(v_i) \subseteq \{v_j, v_k\}$ in $D_l^{\mu-1}$, then $desc(v_i) = desc(v_j) \bigcup_1^* desc(v_k)$.*

3. if $N^+(v_i) \subseteq \{v_j, v_k\}$ in $D_l^{\mu-1}$ such that $M^G[\mu][j]$ and $M^G[\mu][k]$ are both zero and $descv_j \bigcap desc(v_k) \neq \emptyset$, then M^G is not realizable.

The proof is similar with the proof of Theorem 5.4.23. Case 3 can be proved by Theorem 5.4.9.

Algorithm 5.4.2.

Step 1: *for all bad-zeros v_i in row $\vec{r}_\mu$, set $parent(v_i) = \{i\}$ and $desc(v_i) = \emptyset$; for all bad-ones v_j in row $\vec{r}_\mu$, set $parent(v_j) = \emptyset$ and $desc(v_j) = \{j\}$; for all the other vertex v_k (good-zeros or good-ones in row $\vec{r}_\mu$), set $parent(v_k) = desc(v_k) = \emptyset$.*

Step 2: *for all the good-zeros in $\vec{r}_\mu$, update their parent function from right to left by Theorem 5.4.23.*

Step 3: *for all the good-ones in $\vec{r}_\mu$, update their descendant function from left to right by Theorem 5.4.24.*

Step 4: *check the size of parent (descendant) functions for each vertex (see Corollary 5.4.20).*

Step 5: *for each good-zero v_i such that at least one in-neighbor of v_i is non-zero in row $\vec{r}_\mu$, add arc $v_k \rightarrow v_i$ for each $v_k \in parent(v_i)$.*

Step 6: *for each good-one v_j such that at least one out-neighbor of v_i is zero in row $\vec{r}_\mu$, add arc $v_j \rightarrow v_k$ for each $v_k \in desc(v_j)$.*

5.4.5 New Vertices are Added

So far, we only study vertices whose λ-value in $M_{desc}^G[\{1, \cdots, \mu - 1\}][*]$ are non-zero for $\mu \in \{2, \cdots, n\}$. Since there is no all zero column in M_{desc}^G, every vertex will have λ-value non-zero in M^G_{desc}. But a special kind of vertices will also be important during updating. Those vertices are non-zero in row $\vec{r}_\mu$, but have zero λ-value in $M^G_{desc}[\{1, \cdots, \mu - 1\}][*]$. It is easy to prove the theorems as follows.

Theorem 5.4.25. *Let M^G be a genotype matrix in column-descending structure. Let v_i and v_j be two vertices that have zero λ-value in $M^G_{desc}[\{1, \cdots, \mu - 1\}][*]$ ($\mu \in \{2, \cdots, n\}$) and $M^G[\mu][i]$ and $M^G[\mu][j]$ are non-zero. Suppose $i < j$. Then $v_i \longrightarrow v_j$ in D^μ_l.*

Theorem 5.4.26. *Let M^G be a genotype matrix in column-descending structure. Let $v_i \longrightarrow v_j$ be an arc in $D^{\mu-1}_l$ such that $\lambda(v_i)$ and $\lambda(v_j)$ are non-zero in $M^G[\{1, \cdots, \mu - 1\}][*]$ ($\mu \in \{2, \cdots, n\}$). Let v_k be a vertex such that $\lambda(v_k) = 0$ in $M^G_{desc}[\{1, \cdots, \mu - 1\}][*]$ and $M^G[\mu][k] \neq 0$. If $i < k < j$, then $v_i \longrightarrow v_k$ in D^μ_l and v_k and v_j are not comparable in $M^G[\{1, \cdots, \mu - 1\}][*]$.*

Algorithm 5.4.3.

Step 1: *find those vertices that have zero λ-value in $M^G_{desc}[\{1, \cdots, \mu - 1\}][*]$ but are non-zero in row $\vec{r}_\mu$ of M^G_{desc}. We call them "new vertices" here.*

Step 2: *Search from left to right by the output of Algorithm 5.4.2. Let v_i has non-zero lambda-value in $M^G_{desc}[\{1, \cdots, \mu - 1\}]$ and v_k be a "new vertex" such that there is no other "new vertex" between v_i and v_k. If $v_j \in N^+(v_i)$ and $j > k$, then add arc $v_i \longrightarrow v_k$.*

Step 3: *Connect those "new vertices". Let v_{h_1} and v_{h_2} ($h_1 < h_2$) be two "new vertices". If there is no other "new vertex" between v_{h_1} and v_{h_2}, then add arc $v_{h_1} \longrightarrow v_{h_2}$.*

5.4.6 Algorithm to Construct the Hasse Diagram of the Genotype Poset

To construct the Hasse diagram of the genotype poset for M^G_{desc}, we build the Hasse diagram of the subposet of the left-prior genotype poset induced by vertices whose λ-value are non-zero in $M^G_{desc}[\{1, \cdots, \mu\}][*]$ for every $\mu \in \{1, \cdots, n\}$. During induction, we use parent functions to transfer information and find new arcs. Details of the algorithm is as follows.

Algorithm 5.4.4.

Input: *the Hasse diagram $D^{\mu-1}_{\lambda}$ of the subposet of the left-prior genotype poset for $M^G_{desc}[\{1, \cdots, \mu-1\}][*]$ that is induced by vertices whose λ-values are non-zero in $M^G_{desc}[\{1, \cdots, \mu-1\}][*]$, and μ-th row of M^G_{desc}.*

Procedure:

Step 1: *All the bad-zeros and bad-ones in row $\vec{r}_\mu$ are found, checked and connected. For details, see Algorithm 5.4.1.*

Step 2: *Parent and descendant functions for every vertex in $D^{\mu-1}_{\lambda}$ are updated. And good-zeros (good-ones) are connected to bad-zeros (bad-ones). For detail, see Algorithm 5.4.2.*

Step 3: *Vertices that have zero λ-value in $M^G_{desc}[\{1, \cdots, \mu-1\}][*]$ but are non-zero in row $\vec{r}_\mu$ are added into $D^{\mu-1}_{l}$. And related arcs are added by theorems 5.4.25 and 5.4.26.*

Step 4: *The indegree and outdegree of every vertex in D^{μ}_{l} are checked. If any of them are greater than 2, then M^G_{desc} is not realizable.*

Step 5: *The λ function for $M^G_{desc}[\{1, \cdots, \mu\}][*]$ is updated.*

Output: *D^{μ}_{λ}, which is the Hasse diagram of subposet of the left-prior genotype poset for $M^G_{desc}[\{1, \cdots, \mu\}][*]$ that is induced by vertices whose λ-values are non-zero in $M^G_{desc}[\{1, \cdots, \mu\}][*]$, or that M^G_{desc} is not realizable.*

5.5 To Simplify the Hasse Diagram of the Genotype Poset

After constructing the Hasse diagram D^G of the genotype poset for M^G_{desc}, we need to simply D^G into the Hasse diagram of the ordered genotype poset for M^G_{desc}.

Lemma 5.5.1. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the genotype poset for M^G. Let $D^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r}_\mu$. Let v_i, v_j and v_k be three vertices in such that $N^+(v_i) = \{v_j, v_k\}$ in $D^G[supp(\vec{r}_\mu)]$. If $|N^+(v_j) \bigcup N^+(v_k)| > 2$, then M^G is not realizable.*

Theorem 5.5.2. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the Hasse diagram of the genotype poset for M^G. Let $D^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r}_\mu$. Let v_i, v_j and v_k be three vertices such that $N^+(v_i) = \{v_j, v_k\}$ in $D^G[supp(\vec{r}_\mu)]$. If M^G is realizable and $d^-(v_j) = d^-(v_k) = 2$ in $D^G[supp(\vec{r}_\mu)]$, then v_j and v_k have the same set of in-neighbors in $D^G[supp(\vec{r}_\mu)]$.*

Theorem 5.5.2 shows that $K_{2,2}$, which is a complete bipartite graph with two vertices on each sides , will not be changed (simplified) in the Hasse diagram of the genotype poset.

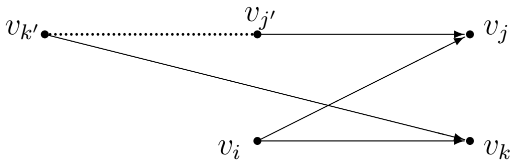

Figure 5.3: Description of Theorem 5.5.2. Note, dot line means dominating relation and solid lines mean covering relation.

Lemma 5.5.3. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the Hasse diagram of the genotype poset for M^G. Let $D^G[supp(\vec{r_\mu})]$ be the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r_\mu}$. Let v_i, v_j and v_k be three vertices such that $v_i \rightarrow v_j$, $d^-(v_j) = 1$ in $D^G[supp(\vec{r_\mu})]$ and v_i and v_k are not comparable. If M^G is realizable, then v_j and v_k are not comparable too.*

Theorem 5.5.4. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the Hasse diagram of the genotype poset for M^G. Let $D^G[supp(\vec{r_\mu})]$ be the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r_\mu}$. Let v_i, v_j and v_k be three vertices such that $v_i \rightarrow v_j$, $v_i \rightarrow v_k$, $d^-(v_j) = 2$ and $d^-(v_k) = 1$ in $D^G[supp(\vec{r_\mu})]$. If M^G is realizable, then v_i and v_j are in different order.*

Corollary 5.5.5. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the Hasse diagram of the genotype poset for M^G. Let $D^G[supp(\vec{r_\mu})]$ be the Hasse diagram of the subposet of*

P^G induced by the support of row $\vec{r}_\mu$. Let v_i, v_j and v_k be three vertices such that v_i dominates v_j, v_i dominates v_k, $d^-(v_j) = 2$ and $d^-(v_k) = 1$ in $D^G[supp(\vec{r}_\mu)]$. If M^G is realizable, then v_i and v_j are in different order.

Corollary 5.5.6. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the Hasse diagram of the genotype poset for M^G. Let $D^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r}_\mu$. Let v_i, v_j and v_k be three vertices such that v_j dominates v_i, v_k dominates v_i, $d^+(v_j) = 2$ and $d^-(v_k) = 1$ in $D^G[supp(\vec{r}_\mu)]$. If M^G is realizable, then v_i and v_j are in different order.*

The proofs of Corollaries 5.5.5 and 5.5.6 are similar with the proof of Theorem 5.5.4.

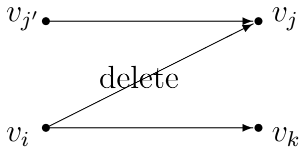

Figure 5.4: Description of Theorem 5.5.4

To construct the Hasse diagram D_o^G of an ordered genotype poset P_o^H for M_{desc}^G, we need to delete those arcs in the Hasse diagram D^G of the genotype poset for M_{desc}^G whose both ends are in different order. By Theorem 5.5.4 and Corollary 5.5.5, we should delete arc $v_i \to v_j$. After this operation, we will get a simplified direct graph D_{sim}^G.

Lemma 5.5.7. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let D^P be the Hasse diagram of the genotype poset P^G for M^G. Let $D^P[supp(\vec{(r)}_\mu)]$ be the Hasse diagram of the subposet of P^G*

induced by the support of row $\vec{r}_\mu$, $\mu \in \{1, \cdots, n\}$. Then every arc in $D^P[supp(\vec{(r)}_\mu)]$ is also in D^G, for every $\mu \in \{1, \cdots, n\}$. And for every arc in D^P, there exists a μ such that the arc is also in $D^P[supp(\vec{(r)}_\mu)]$.

By Lemma 5.5.7, we can easily get another lemma as follows.

Lemma 5.5.8. *Let M^G be a genotype matrix in column-descending structure without repeat rows or columns or all zero column. Let P^G be the genotype poset for M^G and D^G be the Hasse diagram of P^G. Then the Hasse diagram of the subposet of P^G induced by the support of row $\vec{r}_\mu$ is same with the subgraph of D^G induced by by the support of row $\vec{r}_\mu$, for every $\mu \in \{1, \cdots, n\}$.*

Theorem 5.5.9. *Let $D^G_{sim}[supp(\vec{r}_\mu)]$ be the subgraph of D^G_{sim} induced by the support of row $\vec{r}_\mu$. Let v be a vertex in $D^G_{sim}[supp(\vec{r}_\mu)]$. Let v_i and v_j be two descendants of v in $D^G_{sim}[supp(\vec{r}_\mu)]$ that are not comparable. If M^G is realizable, then both indegree and outdegree (if exist out-neighbor) of v_i and v_j in $D^G_{sim}[supp(\vec{r}_\mu)]$ are 1, except $K_{2,2}$ situation.*

Corollary 5.5.10. *Let $D^G_{sim}[supp(\vec{r}_\mu)]$ be the subgraph of D^G_{sim} induced by the support of row $\vec{r}_\mu$. Let v be a vertex in $D^G_{sim}[supp(\vec{r}_\mu)]$. Let v_i and v_j be two ancestors of v in $D^G_{sim}[supp(\vec{r}_\mu)]$ that are not comparable. If M^G is realizable, then both indegree and outdegree (if exist out-neighbor) of v_i and v_j in $D^G_{sim}[supp(\vec{r}_\mu)]$ are 1, except $K_{2,2}$ situation.*

Corollary 5.5.11. *Let $D^G_{sim}[supp(\vec{r}_\mu)]$ be the subgraph of D^G_{sim} induced by the support of row $\vec{r}_\mu$. Let v be a vertex and v' be a descendant of v in $D^G_{sim}[supp(\vec{r}_\mu)]$. Let*

$N^-(v') = \{v_i, v_j\}$. If M^G is realizable, then both indegree and outdegree (if exist out-neighbor) of v_i and v_j in $D^G_{sim}[supp(\vec{r_\mu})]$ are 1, except $K_{2,2}$ situation.

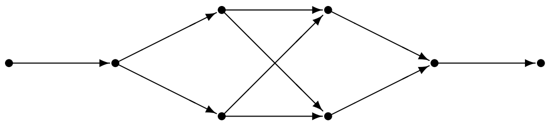

Figure 5.5: One possible output after simplifying.

By Theorems 5.5.2, 5.5.9 and Corollaries 5.5.10, 5.5.11, we can see that in $D^G_{sim}[supp(\vec{r_\mu})]$, for a vertex v, there are only several graph structure as follows. (Note, we do not consider the case that v has no in-neighbor or out-neighbor.)

- v has both indegree and outdegree 1.
- v has indegree 1 and outdegree 2 and both its out-neighbors have indegree 1. Its out-neighbors have outdegree 1 or are inside a $K_{2,2}$.
- v has indegree 2 and outdegree 1 and both its in-neighbors have outdegree 1. Its in-neighbors have indegree 1 or are inside a $K_{2,2}$.
- All neighbors of v are inside $K_{2,2}$s.

Algorithm 5.5.1.

Input: *the Hasse diagram D^G of the genotype poset for M^G_{desc}.*

Procedure: *For each $D^G[supp(\vec{r_\mu})]$, $\mu \in \{1, \cdots, n\}$,*

Step 1: *indegree and outdegree of every vertex in $D^G[supp(\vec{r_\mu})]$ are checked. If any one is greater than 2, then M^G_{desc} is not realizable.*

Step 2: *every vertex that has outdegree 2 and both out-neighbors have indegree 2 is checked by Theorem 5.5.2.*

Step 3: *the arcs whose ends are in different order are removed by Theorem 5.5.4.*

Output: D^G_{sim} *which is the simplified Hasse diagram of the genotype poset for* M^G_{desc}*, or that* M^G_{desc} *is not realizable.*

5.6 To Construct the Hasse Diagram for the Ordered Genotype Poset

In this section, we build the Hasse diagram for the ordered genotype poset by coloring arcs. We define that, the red color means both ends (columns) must be in the same order and blue color means its ends must be in different order. Suppose we already got the output from the last section D^G_{sim}, the Hasse diagram of the genotype poset whose "bad chords" have been deleted.

Lemma 5.6.1. *Let* M^G *be a genotype matrix and* P^G *be its genotype poset. Let* $v_i \to v_j$ *be an arc in the Hasse diagram (denoted as* D^G*) of* P^G*. If* M^G *is realizable and there is a row* $\vec{r}_\mu$ *such that* $M^G[\mu][i] = 1$*, then* v_i *and* v_j *are in the same order.*

Lemma 5.6.2. *Let* M^G *be a realizable genotype matrix and* P^G *be the genotype poset for* M^G*. In* D^G_{sim}*, let* $v_i \to v_j$ *be an arc. If there is a row* $\vec{r}_\mu$*, such that in* $D^G_{sim}[supp(\vec{r}_\mu)]$ *there is a vertex* v_k *with outdegree 2, every vertex on the path between* v_k *and* v_j *has both indegree and outdegree 1 and the indegree of* v_j *is 1, then* v_i *and* v_j *have the same order.*

Definition 5.6.3. *Let M^G be a genotype matrix and P^G be its genotype poset. Let v be a vertex in the Hasse diagram (denoted as D^G) of P^G. We define $L_{max}(v) = max_{\mu \in \{1,\cdots,n\}}\{i : i$ is the rightmost site in row $\vec{r}_\mu$ such that $M^G[\mu][i] = 1\}$, and $L_{min}(v) = min_{\mu \in \{1,\cdots,n\}}\{i : i$ is the rightmost site in row $\vec{r}_\mu$ such that $M^G[\mu][i] = 1\}$.*

Lemma 5.6.4. *Let M^G be a realizable genotype matrix. In D^G_{sim}, if there is a vertex v_j such that $L_{max}(v_j) \neq L_{min}(v_j)$, then every pair of vertices on the path(s) between $v_{L_{max}(v_j)}$ and v_j has the same order.*

Algorithm 5.6.1.

Input: *D^G_{sim}, which is the simplified Hasse diagram for the genotype poset for M^G_{desc}.*

Procedure:

Step 1: *Arc $v_i \to v_j$ in D^G_{sim} is colored red,*

- *if there exists $\mu \in \{1,\cdots,n\}$ such that $M[\mu][i] = 1$ and $M[\mu][j] \neq 0$; (See Lemma 5.6.1).*
- *if there exists $\mu \in \{1,\cdots,n\}$ and k ($k \leq i < j$), such that in $D^G_{sim}[supp(\vec{r}_\mu)]$, v_k has outdegree 2, all the other vertices between v_k and v_j are of indegree and outdegree 1 and the indegree of v_j is also 1. (See Lemma 5.6.2)*
- *calculate the L_{max} and L_{min} for every vertex. Color arc red by Lemma 5.6.4.*

Step 2: *Arc $v_i \to v_j$ in D^G_{sim} is colored blue,*

- *if there exists $\mu \in \{1,\cdots,n\}$ such that $v_k \to v_j$ is already colored red, $k \neq i$;*

- *if there exists $\mu \in \{1, \cdots, n\}$ such that $v_i \rightarrow v_k$ is already colored red, $k \neq j$.*

Step 3: *Each arc in $D^G_{sim}[supp(\vec{r}_\mu)]$ ($\mu \in \{1, \cdots, n\}$), e.g., $v_i \rightarrow v_j$, that has not been colored, is colored as follows (see Theorem 5.6.5).*

1. *each arc that has not been colored in step 1 and step 2 of Algorithm 5.6.1 is assigned an index.*
2. *each arc is assigned some neighbor by searching $D^G_{sim}[supp(\vec{r}_\mu)]$, $\mu \in \{1, \cdots, n\}$. An example is given as follows.*

   Suppose the index of arc $v_i \rightarrow v_j$ is k_1, the index of arc $v_{i'} \rightarrow v_j$ is k_2 and the index of arc $v_i \rightarrow v_{j'}$ is k_3.

   - *If there is a μ_1 such that $v_i \rightarrow v_j$ and $v_{i'} \rightarrow v_j$ in $D^G_{sim}[supp(\vec{r}_{\mu_1})]$, then arc k_2 is a neighbor of arc k_1.*
   - *If there is a μ_2 such that $v_i \rightarrow v_j$ and $v_i \rightarrow v_{j'}$ in $D^G_{sim}[supp(\vec{r}_{\mu_2})]$, then arc k_3 is a neighbor of arc k_1.*
3. *Randomly pick one arc and color it red (or blue). Its neighbors are colored by different order. Doing depth first search. If there are still arcs not colored, then repeatedly do step 3.*

Step 4: *if there is any confliction in D^G_{sim} during the coloring, then the input matrix is not realizable.*

Output: *the Hasse diagram of an ordered genotype poset for M^G_{desc}, or that M^G_{desc} is not realizable.*

Theorem 5.6.5. *After step 1 and step 2 of Algorithm 5.6.1, for those arcs that have not been colored,*

- *if there is a row $\vec{r}_\mu$ such that the outdegree of v_i is 2 in $D^G_{sim}[supp(\vec{r}_\mu)]$, and the two arcs from v_i are colored same, then M^G_{desc} is not realizable;*
- *If there is a row $\vec{r}_\mu$ such that the indegree of v_i is 2 in $D^G_{sim}[supp(\vec{r}_\mu)]$, and the two arcs to v_j are colored same, then M^G_{desc} is not realizable.*
- *If $v_i \rightarrow v_j$ and the outdegree of v_i and the indegree of v_j are both 1 in any $D^G_{sim}[supp(\vec{r}_\mu)]$, $\mu \in \{1, \cdots, n\}$, then the arc could be colored randomly.*

Let $v_i \rightarrow v_j$ be an arc in D^G_{sim}. If $v_i \rightarrow v_j$ is colored red and blue in Algorithm 5.6.1, then there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G_{desc}[\mu_1][i]$ and $M^G_{desc}[\mu_1][j]$ are in the same order and $M^G_{desc}[\mu_2][i]$ and $M^G_{desc}[\mu_2][j]$ are in different order. Since $M^G_{desc}[\mu_1][i]$ and $M^G_{desc}[\mu_1][j]$ are non-zero, $M^G[\{\mu_1, \mu_2\}][\{i, j\}]$ is not realizable. That is the reason we have step 4 of Algorithm 5.6.1.

5.7 To Build a Legal Expansion

Algorithm 5.7.1.

Input: *the Hasse diagram of an ordered genotype poset for M^G_{desc}.*

Procedure:

Step 1: *a haplotype matrix is built row by row. If $M[\mu, i] = M[\mu, j] = 2$ and there is an arc $v_i \rightarrow v_j$ colored red, then these 2's should be in the same order in the haplotype matrix; else if 2's are in different order;*

Step 2: *the original order for columns is recovered.*

Output: *a haplotype matrix that is a legal expansion of M^G_{input}.*

5.8 Complexity

Theorem 5.8.1. *Let M^H be an $(n \times m)$-haplotype matrix without repeat rows or columns. Assume there is no all zero columns in M^H. If M^H fits a perfect phylogeny, then $\lceil \frac{2}{3}m \rceil \leq n \leq m$.*

Theorem 5.8.2. *Let M^G be an $(n \times m)$-genotype matrix without repeat rows or columns. If M^G is realizable, then $n \geq \frac{1}{9}m^2$.*

In step 1 of Algorithm 5.2.1, we can use linear time (O(nm)-time)to remove repeat rows and columns and do the checking. Only the sorting need $O(m\lg(m))$-time. By Theorem 5.8.2, it is obviously that $O(m\lg(m))$ is less than $O(nm)$.

In step 2 of Algorithm 5.2.1, since there is no all zero column in M^G_{desc}, by Theorem 4.4.5, it is easy to see that the upper limit for the number of arcs in the Hasse diagram of the left-prior genotype poset for M^G_{desc} is $2m$. And the sizes of $parent_0$ and $parent_1$ for each vertex are not more than 2. Then the Hasse diagram of the genotype poset can be build in $O(nm)$-time. With the same reason, all the other steps in Algorithm 5.2.1 can be done in $O(nm)$-time.

5.9 Test Results

We implemented our algorithm by Matlab, and compared it with existing programs for the PPH problem. During three solutions given by D. Gusfield, program DPPH [4] is the fastest [8]. It is about two times faster than HPPH [15] and three times faster than GPPH [9]. Some representative examples are shown in the table below.

				Ave Running Time (sec)
Sites (m)	Individuals (n)	# of Test Cases	DPPH	Our Algorithm
50	1000	20	0.20	0.08
100	1000	20	1.06	0.15
300	150	30	1.07	0.06
500	250	30	5.72	0.18
1000	500	30	45.85	0.65
1000	1000	10	92.20	1.24
2000	1000	10	467.18	2.43

Table 5.1: Test results

In the case of $m = 2000$, $n = 1000$, our program is about 200 times faster than DPPH, and linear behavior of its running time is clear. The result is an average of 10 test cases. Out test data is generated by the program in [32]. That program is the widely-used standard for generating sequences the reflect the coalescent model of SNP sequence evolution. The cases of 50 and 100 sites and 1000 individual are included because they reflect the sizes of the subproblems that of current interest in larger genomic scans. In those applications, there may be a huge number of such subproblems that will be examined. Our program can be downloaded at http://www.csee.wvu.edu/∼yliu/lpph.

5.10 All Solutions to the PPH Problem

To find all the legal expansions from an input is also an important question. Our algorithm can provide a solution to the problem.

After simplifying the Hasse diagram of the genotype matrix for M^G_{desc} (step 4 of Algorithm 5.2.1) we will color the arcs of D^G_{sim} to show the orders between vertices. In Algorithm 5.6.1, sometime an arc can be colored “randomly”. After coloring those arc whose both ends are in the same order, we can pick a not-colored arc and find the "component" contains the arc. Those "components" are defined as follow.

Definition 5.10.1. *Let $v_i \to v_j$ be an arc in D^G_{sim} that has not been colored after step 1 of Algorithm 5.6.1. If there is a μ such that*

- *$v_i \to v_j$ and $v_{k_1} \to v_j$ in $D^G_{sim}[supp(\vec{r}_\mu)]$, then we say arc $v_{k_1} \to v_j$ is in same **component** with arc $v_i \to v_j$.*
- *Or $v_i \to v_j$ and $v_i \to v_{k_2}$ in $D^G_{sim}[supp(\vec{r}_\mu)]$, then we say arc $v_j \to v_{k_2}$ is in same **component** with arc $v_i \to v_j$.*

Those components are also "connected components" in the graph, which takes not colored arcs of D^G_{sim} as vertices in step 3 of Algorithm 5.6.1.

Lemma 5.10.2. *If a component has a subgraph which is a cycle with odd length, then M^G is not realizable.*

Theorem 5.10.3. *Let the number of components be k. The number of all solutions from M^G_{desc} is 2^k.*

Those components are already found out by step 3 of Algorithm 5.6.1. And if we change the colors in those component, then all the solutions are easily to be found.

Appendix A

Proofs

Proof of Theorem 2.2.9 is as follows.

Proof. If $d_D^-(v) > k$, then by Theorem 2.2.7, it is easy to see that there is an antichain in P whose size is greater than k. It is symmetric for $d_D^+(v) > k$. □

Proof of Theorem 3.3.2 is as follows.

Proof. Let M^H be an $(n \times m)$-haplotype matrix. If M^H fits a perfect phylogeny T, every node on T is labelled by a vertex in V_{row}. Then $|V_{row}| = |V_{col}|$. For every two vertices v_{c_1} and v_{c_2} in P^H_{col}, if $v_{c_1} \geq_{col} v_{c_2}$, then there are two nodes on T such that one is the ancestor of the other, and their row labels are comparable in P^H_{row}. If two vertices are comparable in P^H_{row}, then they are on the same path from the ancestor sequence. The related vertices in P^H_{col} are comparable too. □

Proof of Theorem 3.3.3 is as follows.

Proof. If there is a row $\vec{r_\mu}$ such that $D^H[supp(\vec{r_\mu})]$ is not a directed path, then there are two vertices (columns) v_i and v_j such that they are not comparable in P^H and

$M^H[\mu][i] = M^H[\mu][j] = 1$. Then $M^H[*][\{i,j\}]$ does not fit a perfect phylogeny. Neither does M^H.

If $D^H[supp(\vec{r}_\mu)]$ is a directed path for every $\mu \in \{1, \cdots, n\}$, then for every $\mu' \in \{1, \cdots, n\}$ and $i, j \in \{1, \cdots, m\}$ such that $M^H[\mu'][i] = M^H[\mu'][j] = 1$, $\vec{c}_i$ and $\vec{c}_j$ are comparable. Then there is no forbidden matrix in M^H and M^H fits a perfect phylogeny. □

Proof of Corollary 3.3.4 is as follows.

Proof. Every arc in $D^H[supp(\vec{r}_\mu)]$ ($\mu \in \{1, \cdots, n\}$) is also in D^H. By Theorem 3.3.3, the number of arcs in D^H is up to m, since every vertex has indegree 1 at most in each $D^H[supp(\vec{r}_\mu)]$. □

Proof of Theorem 4.1.4 is as follows.

Proof. Let $\vec{c}_i$ and $\vec{c}_j$ be two columns in M^G. If $\vec{c}_i$ and $\vec{c}_j$ are neither in the same order nor in different order, then there are two $(2,2)$-rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ between columns $\vec{c}_i$ and $\vec{c}_j$, such that $M^G[\mu_1][i]$ and $M^G[\mu_1][j]$ are in the same order and $M^G[\mu_2][i]$ and $M^G[\mu_2][j]$ are in different order. Then $M^G[\{\mu_1, \mu_2\}][\{i,j\}]$ is not realizable by Theorem 3.1.1. It conflicts with the assumption that M^G is realizable. □

Proof of Lemma 4.3.1 is as follows.

Proof. Without loss generality, suppose there is no row $\vec{r}_{\mu_2}$ such that $M^G[\mu_2][i] < M^G[\mu_2][j]$. Then, by Definition 4.2.1, $v_i \geq v_j$. It conflicts with the assumption that v_i and v_j are not comparable. □

Proof of Theorem 4.3.2 is as follows.

Proof. If two columns are different in M^G then the related columns in M^H are different too. Since we assume there is no repeat columns in M^G, the haplotype poset P^H_{col} for M^H has the same vertex set with any ordered genotype posets P^G_o for M^G. Since M^G is realizable, by Theorem 4.1.4, every pair of columns in M^G are in the same order or in different order. By Definitions 3.2.2 and 4.2.3, we can set up the order between each pair of columns in M^G by the dominating relation in P^H_{col} and build an ordered genotype poset. It is obvious that they are isomorphic with each other. □

Proof of Lemma 4.3.3 is as follows.

Proof. Let v_i and v_j be two vertices in P^G. If $i > j$, then by Definition 2.1.3, $\begin{pmatrix} x_i \\ y_i \end{pmatrix} \leq \begin{pmatrix} x_j \\ y_j \end{pmatrix}$. Since M^G has no repeat column, there is at least one row $\vec{r}_\mu$ such that $M^G[\mu][i] < M^G[\mu][j]$. It conflicts with the assumption that $v_i \geq v_j$ in P^G. □

Proof of Theorem 4.3.4 is as follows.

Proof. All of $P^0_l, P^1_l, \cdots, P^n_l, P^G_l, P^G$ and P^G_o take columns of the genotype matrix as vertices, so they have the same vertex set. By Definitions 4.2.1, 4.2.2 and 4.2.3, it is easy to prove that for any two vertices v_i and v_j, if $v_i \geq_o v_j$ then $v_i \geq v_j$. And if $v_i \geq_l v_j$ in $M^G[\{1, \cdots, \mu_2\}][*]$ then $v_i \geq_l v_j$ in $M^G[\{1, \cdots, \mu_1\}][*]$ when $0 \leq \mu_1 \leq \mu_2 \leq n$. Then $P^G \supseteq P^G_o$, and $P^{\mu_1}_l \supseteq P^{\mu_2}_l$ when $1 \leq \mu_1 \leq \mu_2 \leq n$. Since M^G is in column-descending structure, if $v_i \geq v_j$, then by Lemma 4.3.3, $i < j$. That is, if $v_i \geq v_j$, then $v_i \geq_l v_j$. Then $P^n_l = P^G_l = P^G$. □

Proof of Lemma 4.4.1 is as follows.

Proof. If v_i and v_j are not comparable in any ordered genotype poset, then v_i and v_j are in different order, or (by Lemma 4.3.1) there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][i] > M^G[\mu_1][j]$ and $M^G[\mu_2][i] < M^G[\mu_2][j]$. In the first case, there is a $(2,2)$-row $\vec{r}_{\mu'}$ such that $M^G[\mu'][i]$ and $M^G[\mu'][j]$ are in different order. Then, $M^G[\{\mu,\mu'\}][\{i,j\}]$ is not realizable. In the second case, $M^G[\{\mu,\mu_1,\mu_2\}][\{i,j\}]$ is not realizable. Both cases conflict with the assumption that M^G is realizable. □

Proof of Lemma 4.4.2 is as follows.

Proof. By Theorem 4.1.4, v_i, v_j and v_k are in the same order or different order with each other; else M^G is not realizable. Since $M^G[\mu][i] = M^G[\mu][j] = M^G[\mu][k] = 2$, at least one pair of v_i, v_j and v_k are in the same order. Without loss generality, suppose v_i and v_j are in the same order but not comparable in P^G, then by Lemma 4.3.1, there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][i] > M^G[\mu_1][j]$ and $M^G[\mu_2][i] < M^G[\mu_2][j]$. Then, $M^G[\{\mu,\mu_1,\mu_2\}][\{i,j\}]$ is not realizable. Neither is M^G. □

Proof of Theorem 4.4.3 is as follows.

Proof. Let $D^G_o[supp(\vec{r}_\mu)]$ be the Hasse diagram of $P^G_o[supp(\vec{r}_\mu)]$. Suppose there is a row $\vec{r}_\mu$ such that $D^G_o[supp(\vec{r}_\mu)] = 3$. Let v_i, v_j and v_k be the three vertices in $D^G_o[supp(\vec{r}_\mu)]$ that are not comparable with each other. By the definition of the support of a vector, $M^G[\mu][i]$, $M^G[\mu][j]$ and $M^G[\mu][k]$ are all non-zero. By Lemmas 4.4.1 and 4.4.2, M^G is not realizable. It conflicts with the assumption that M^G is realizable. □

Proof of Corollary 4.4.4 is as follows.

Proof. Let $D^G[supp(\vec{r}_\mu)]$ be the Hasse diagram of $P^G[supp(\vec{r}_\mu)]$. Let v_i, v_j and v_k be three vertices in $D^G[supp(\vec{r}_\mu)]$ that are not comparable with each other. They are also not comparable in any ordered genotype posets. Then, M^G is not realizable. It causes the contradiction. □

Proof of Theorem 4.4.5 is as follows.

Proof. Let v_i be a vertex in D_l^G. Suppose $N^-(v_i) = \{v_h, v_k, v_j\}$, $1 \leq h < j < k < i \leq m$. Then v_h, v_k, v_j are not comparable with each other (by Lemma 2.2.7). Since M^G has no all zero column, by the definition of the left-prior genotype poset, there is one row $\vec{r}_\mu$ such that $M^G[\mu][h], M^G[\mu][j], M^G[\mu][k], M^G[\mu][i] \neq 0$. However, the left-prior Hasse diagram for a genotype matrix in column-descending structure is same the genotype poset for the genotype poset. Then, the width of $P^G[supp(\vec{r}_\mu)]$ is greater than 2. By Corollary 4.4.4, M^G is not realizable. It causes the contradiction. □

Proof of Lemma 4.4.6 is as follows.

Proof. By Theorems 2.2.9 and 4.4.3, it is easy to prove that $d^+(v) \leq 2$ and $d^-(v) \leq 2$ for any vertex v in $D_o^G[supp(\vec{r}_\mu)]$. Let v be a vertex in $D_o^G[supp(\vec{r}_\mu)]$ such that $N^-(v) = \{v_i, v_j\}$. By Theorem 2.2.7, v_i and v_j are not comparable. Since both v_i and v_j are in the same order with v, v_i and v_j are in the same order. Then, there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][i] > M^G[\mu_1][j]$ and $M^G[\mu_2][i] < M^G[\mu_2][j]$. $M^G[\{\mu, \mu_1, \mu_2\}][\{i, j\}]$ is not realizable. It conflicts with the assumption that M^G is realizable. □

Proof of Lemma 4.4.7 is as follows.

Proof. If $d^+(v_i) = 2$ in $D^G_o[supp(\vec{r}_\mu)]$, suppose $N^+(v) = \{v_j, v_k\}$, then v_j and v_k are not comparable. If $M^G[\mu][i] = 2$, then v_i, v_j and v_i, v_k are in the same order. Then v_j and v_k are in the same order too. By Lemma 4.3.1, there are rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][j] > M^G[\mu_1][k]$ and $M^G[\mu_2][j] < M^G[\mu_2][k]$. $M^G[\{\mu, \mu_1, \mu_2\}][\{j, k\}]$ is not realizable. It causes the contradiction with the assumption. If any descendant of v_i, e.g., $v_{j'}$, is 1 in row $\vec{r}_\mu$, then $v_{j'}$ is also a descendant of v_j or v_k. Without loss generality, if $v_{j'}$ is a descendant of v_j, then $v_{j'}$ and v_k are not comparable, by Lemma 4.4.6. By Lemma 4.4.1, M^G is not realizable. It causes the contradiction. If any ancestor of v_i, e.g., $v_{k'}$, is 2 in row $\vec{r}_\mu$, then $M^G[\mu][k']$ does not dominate $M^G[\mu][i]$. It conflicts with the Definitions 4.2.3. By the proof above and Lemma 4.4.6, we can easily prove the second part of this lemma. □

Proof of Theorem 4.4.8 is as follows.

Proof. If M^G is realizable, then by Theorem 4.4.3, the width of $P^G_o[supp(\vec{r}_\mu)]$ is less than 3 for every $\mu \in \{1, \cdots, n\}$. By Lemma 4.4.6, $d^+(v) \leq 2$ and $d^-(v) \leq 1$ for every vertex v in $D^G_o[supp(\vec{r}_\mu)]$. If there are two vertices have outdegree 2, then the union of their out-neighbors has size of 3 and every pair of vertices inside are not comparable with each other. Then only one vertex in $D^G_o[supp(\vec{r}_\mu)]$ may have outdegree 2. By Lemma 4.4.7, it is easy to get the first case of this theorem. Suppose the width of $P^G_o[supp(\vec{r}_\mu)]$ is 2 and every vertex has both indegree and outdegree of 1. If there is at least one vertex with value 1 in row $\vec{r}_\mu$, then the vertex is not comparable with another vertex in $D^G_o[supp(\vec{r}_\mu)]$, by Lemma 4.4.1, M^G is not realizable. Then there is no 1 in row $\vec{r}_\mu$. That is same with the second case of this theorem. If the width of $P^G_o[supp(\vec{r}_\mu)]$ is 1, then every vertex has indegree and outdegree 1 and every pair of vertices are comparable with each other. By the definition of ordered genotype posets, no 2 is an ancestor of 1 in row $\vec{r}_\mu$. This matches our third case.

Let M^H be a haplotype matrix inferred from M^G following the orders between columns. Then $P_o^G = P_{col}^H$ and $D_o^G = D_{col}^H$. For any pair of vertices v_i, v_j in $D_o^G[supp(\vec{r}_\mu)]$ ($\mu \in \{1, \cdots, n\}$), if $v_i \rightarrow v_j$, then $v_i \rightarrow v_j$ in $D_{col}^H[supp(\vec{r}_{2\mu-1})]$ or $D_{col}^H[supp(\vec{r}_{2\mu})]$ or both. If $v_i \not\rightarrow v_j$, then v_i and v_j are not on both $D_{col}^H[supp(\vec{r}_{2\mu-1})]$ and $D_{col}^H[supp(\vec{r}_{2\mu})]$. Then the subgraph of D_{col}^H induced by the support of each row is a directed path. By Theorem 3.3.3, M^H fits a perfect phylogeny. Then M^G is realizable. □

Proof of Lemma 5.3.1 is as follows.

Proof. By Lemma 4.3.3, if $i < j$, then $v_i \not\leq v_j$ in any genotype poset for M^G. Then, there is a row $\vec{r}_{\mu'}$ such that $M^G[\mu'][i] > M^G[\mu'][j]$. And $M^G[\{\mu, \mu'\}][\{i, j\}]$ is not realizable. So M^G is not realizable. □

Proof of Theorem 5.4.3 is as follows.

Proof. By Theorem 4.4.5, $d^-(v_i) \leq 2$ in D_l^G for every vertex v_i such that $\lambda(v_i) \neq 0$. If $d^-(v_i) = 2$, suppose $N^-(v_i) = \{v_j, v_k\}$, then v_j and v_k are not comparable (by Lemma 2.2.7). If $\lambda(v_i) = 1$, then there is a row $\vec{r}_\mu$ such that $M^G[\mu][i] = M^G[\mu][j] = M^G[\mu][k] = 1$. Since M^G is in column-descending structure, there are two rows $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$ such that $M^G[\mu_1][j] > M^G[\mu_1][k]$ and $M^G[\mu_2][j] < M^G[\mu_2][k]$. Then $M^G[\{\mu, \mu_1, \mu_2\}][\{j, k\}]$ is not realizable. And M^G is not realizable. If $\lambda(v_i) = 2$, then $M^G[\mu][j]$ and $M^G[\mu][k]$ are non-zero, for each row $\vec{r}_\mu$ such that $M^G[\mu][i] \neq 0$. If one of them or both of them are 1, then the submatrix of M^G induced from $\vec{c}_j$ and $\vec{c}_k$ is not realizable. So M^G is not realizable. □

Proof of Corollary 5.4.4 is as follows.

Proof. If $d^-(v_i) = 3$ in D_l^μ, suppose $N^-(v_i) = \{v_{i_1}, v_{i_2}, v_{i_3}\}$, then v_{i_1}, v_{i_2} and v_{i_3} are not comparable with each other in D_l^μ. They are also not comparable with each other in D_l^G. If $\lambda(v_i) = 2$ in $M^G[\{1, \cdots, \mu\}][*]$, then there is a row $\vec{r}_{\mu 1}$ such that $M^G[\mu_1][i_1]$, $M^G[\mu_1][i_2]$, $M^G[\mu_1][i_3]$ and $M^G[\mu_1][i]$ are all non-zero. Let D^G be the Hasse diagram of the genotype poset for M^G. $D^G = D_l^n$, since M^G is in column-descending structure. Then the width of $P^G[supp(\vec{r}_{\mu_1})]$ is 3. By Corollary 4.4.4, M^G is not realizable. Similar proof for the case that $\lambda(v_i) = 1$ and $d^-(v_i) \leq 1$ in D_l^μ. □

Proof of Lemma 5.4.7 is as follows.

Proof. Without loss generality, suppose $i < j$. If v_i and v_j are not comparable in P_l^μ, then there is a row $\vec{r}_{\mu_1}$ such that $M^G[\mu_1][i] < M^G[\mu_1][j]$ $(1 \leq \mu_1 \leq \mu)$. Since $i < j$ and M^G is in column-descending structure, there is a row $\vec{r}_{\mu_2}$ such that $M^G[\mu_2][i] > M^G[\mu_2][j]$. Then v_i and v_j are not comparable in P^G. □

Proof of Lemma 5.4.8 is as follows.

Proof. v_j and v_k are not comparable in M^G, since $v_j \to v_i$ and $v_k \to v_i$ in $D_l^{\mu-1}$. Because $M^G[\mu][j] = M^G[\mu][k] = 0$ and $M^G[\mu][i] \neq 0$, $v_j \not>_l v_i$ and $v_k \not>_l v_i$ in P_l^μ. Since $\lambda(\vec{c}_i) \neq 0$, both $\lambda(\vec{c}_j)$ and$\lambda(\vec{c}_k)$ are non-zero. So there is a row $\vec{r}_{\mu_1}$ such that $M^G[\mu_1][i]$, $M^G[\mu_1][j]$ and $M^G[\mu_1][k]$ are non-zero. By Corollary 4.4.4, M^G is not realizable. □

Proof of Lemma 5.4.11 is as follows.

Proof. It is easy to see that $v_i \not>_l v_j$ and $v_i \not>_l v_k$ in P_l^μ. And v_j and v_k are not comparable. Since $\lambda(v_h) \neq 0$ in $M^G[\{1, \cdots, \mu - 1\}][*]$, there is a row $\vec{r}_{\mu_1}$ such that

$M^G[\mu_1][i]$, $M^G[\mu_1][j]$, $M^G[\mu_1][k]$ and $M^G[\mu_1][h]$ are all non-zero. Then M^{desc} is not realizable, by Corollary 4.4.4. □

Proof of Corollary 5.4.13 is as follows.

Proof. Let v_j and v_k be two ancestors of v_i in $D_l^{\mu-1}$ such that they are bad-ones in $\vec{r}_\mu$ and are not comparable. Then there are two bad-zeros $v_{j'}$ and $v_{k'}$ such that $v_{j'}$ is an ancestor of v_j and $v_{k'}$ is an ancestor of v_k in $D_l^{\mu-1}$. Without loss generality, if $v_{j'}$ is also an ancestor of v_k, then by Theorem 5.4.12, M^G is not realizable. Then $v_{j'}$ and $v_{k'}$ are not comparable. By Theorem 5.4.9, M^G is not realizable. □

Proof of Theorem 5.4.14 is as follows.

Proof. Let v_i be a vertex in $D_l^{\mu-1}(B0(\vec{r}_\mu))$. Suppose v_i has three descendants v_{i_1}, v_{i_2} and v_{i_3} in $D_l^{\mu-1}(B0(\vec{r}_\mu))$ that are not comparable with each other. Then they have three descendants $v_{i'_1}$, $v_{i'_2}$ and $v_{i'_3}$ in $D_l^{\mu-1}$ such that $M^G[\mu][i'_1]$, $M^G[\mu][i'_2]$ and $M^G[\mu][i'_3]$ are non-zero. Without loss generality, if $v_{i'_1}$ is also a descendant of v_{i_2} in $D_l^{\mu-1}$, then by Theorem 5.4.9, M^G is not realizable. Then $v_{i'_1}$, $v_{i'_2}$ and $v_{i'_3}$ are not comparable with each other. By Corollary 4.4.4, M^G is not realizable. So there is at most one vertex in $D_l^{\mu-1}(B0(\vec{r}_\mu))$ that has outdegree 2. With the similar prove, we can get the width of $P_l^{\mu-1}(B0(\vec{r}_\mu))$ is at most 2. By Corollary 5.4.10, we can get every vertex in $D_l^{\mu-1}(B0(\vec{r}_\mu))$ has indegree at most 1.

Let v_j be a vertex in $D_l^{\mu-1}(B1(\vec{r}_\mu))$. By Corollary 4.4.4, the width of $P_l^{\mu-1}(B1(\vec{r}_\mu))$ is at most 2. And at most one vertex in $D_l^{\mu-1}(B1(\vec{r}_\mu))$ has outdegree 2. By Corollary 5.4.13, every vertex in $D_l^{\mu-1}(B1(\vec{r}_\mu))$ has indegree at most 1. □

Proof of Corollary 5.4.16 is as follows.

Proof. If v_i is a bad-zero, then by Theorem 5.4.14, every vertices on the path between v_i and v_j are non-zero. Then v_j covers v_i in P_l^μ. Same when v_i is a bad-one. □

Proof of Corollary 5.4.20 is as follows.

Proof. If there is a vertex v such that $|parent(v)| = 2$, then by Lemma 5.4.19, two vertices in $parent(v)$ are not comparable with each other. By Theorem 5.4.9, M^G is not realizable. If there is a vertex v such that $|desc(v)| = 3$, then by Lemma 5.4.19 and Theorem 5.4.14, M^G is not realizable. □

Proof of Theorem 5.4.23 is as follows.

Proof. By Definition 5.4.17, it is easy to prove case 1 above. For case 2, it is already proved in Corollary 5.4.21. If $M^G[\mu][j]$ and $M^G[\mu][k]$ are non-zero and $parent(v_i) \bigcap parent(v_j) \neq \emptyset$, then by Theorem 5.4.12, M^G is not realizable. □

Proof of Lemma 5.5.1 is as follows.

Proof. Suppose $N^+(v_j) \bigcup N^+(v_k) = \{v_{h_1}, v_{h_2}, v_{h_3}\}$. Then they are not comparable with each other. By Corollary 4.4.4, M^G is not realizable. □

Proof of Theorem 5.5.2 is as follows.

Proof. Suppose $N^-(v_j) = \{v_i, v_{j'}\}$ and $N^-(v_k) = \{v_i, v_{k'}\}$. If $v_{j'}$ and $v_{k'}$ are not comparable with each other, then v_i, $v_{j'}$ and $v_{k'}$ are not comparable with each other. By Corollary 4.4.4, M^G is not realizable. Suppose $v_{j'}$ dominates $v_{k'}$. Then v_i are not comparable with $v_{j'}$ and $v_{k'}$. Also $M^G[\mu][i] = M^G[\mu][j'] = M^G[\mu][k'] = M^G[\mu][j] = M^G[\mu][k] = 2$. If v_i and v_j are in the same order, then v_j and $v_{j'}$ are in different order, v_k and $v_{k'}$ are in different order and $v_{j'}$ and $v_{k'}$ are in different order too. By

Theorem 4.4.3, M^G is not realizable. Same when v_i and v_k are in the same order. So $v_{j'} = v_{k'}$. □

Proof of Lemma 5.5.3 is as follows.

Proof. Suppose v_j and v_k are comparable. If v_j dominates v_k, then v_i dominates v_k too. It causes a contradiction with the assumption. If v_k dominates v_j, then v_k dominates v_i, since v_i covers v_j and $d^-(v_j) = 1$. It causes the contradiction too. □

Proof of Theorem 5.5.4 is as follows.

Proof. Suppose $N^-(v_j) = \{v_i, v_{j'}\}$. Then v_i and $v_{j'}$ are not comparable. And $M^G[\mu][i] = M^G[\mu][j'] = M^G[\mu][j] = M^G[\mu][k] = 2$. If v_i and v_j are in the same order, then v_i and v_k are in different order and v_j and $v_{j'}$ are in different order. By Lemma 5.5.3, $v_{j'}$ and v_k are not comparable. Then by Theorem 4.4.3, M^G is not realizable. □

Proof of Lemma 5.5.7 is as follows.

Proof. If $v_i \to v_j$ in $D^P[supp(\vec{(r)}_\mu)]$, then v_i dominates v_j in P^G. And $i < j$ because M^G is in column-descending structure. If v_i does not covers v_j in M^G, then by the definition of covering relation in posets, there is at least a vertex v_k such that $v_i > v_k > v_j$ and $i < k < j$. But obviously $M^G[\mu][k] = 0$. Since $M^G[\mu][j] \neq 0$, $v_k \ngeq v_j$.

For an arc $v_i \to v_j$ in D^G, since M^G has no all zero column, there is a μ such that $M^G[\mu][j] \neq 0$. Then $D^P[supp(\vec{(r)}_\mu)]$ has arc $v_i \to v_j$ too. □

Proof of Theorem 5.5.9 is as follows.

Proof. Let $D^G[supp(\vec{r}_\mu)]$ be the subgraph of the Hasse diagram of the genotype poset for M^G. By Corollary 5.5.5, if only one of v_i and v_j has indegree 2 in $D^G[supp(\vec{r}_\mu)]$, then one arc is deleted and v_i and v_j have indegree 1 in $D^G_{sim}[supp(\vec{r}_\mu)]$. If both of them have indegree 2, then only $K_{2,2}$ is allowed.

Suppose v_i has outdegree 2 in $D^G_{sim}[supp(\vec{r}_\mu)]$. Assume $N^+(v_i) = \{v_{i_1}, v_{i_2}\}$. If both v_{i_1} and v_{i_2} are not comparable with v_j, then by Corollary 4.4.4, M^G is not realizable. If one of v_{i_1} and v_{i_2} is comparable with v_j, then by Corollary 5.5.5, one arc should be removed and both of them should have outdegree 1 in $D^G_{sim}[supp(\vec{r}_\mu)]$. □

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

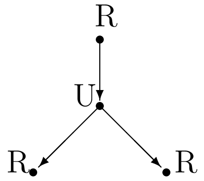

Figure A.1: "R" means the node is labelled by a row in M^H, and "U" means that the node is not labelled by any row of M^H. Note, in this example the first "R" is the ancestor vector.

Proof of Lemma 5.6.2 is as follows.

Proof. Since in $D^G_{sim}[supp(\vec{r}_\mu)]$ v_k has outdegree 2 and every vertex on the path between v_k and v_j has both indegree and outdegree 1, there is at least one vertex v_h such that v_h is not comparable with both v_i and v_j in P^G. Then v_h is also not comparable with v_i and v_j in any ordered genotype poset for M^G. If v_i and v_j have different order, then v_i and v_j are not comparable in any ordered genotype poset too. So the width of antichain in $P^G_{sim}[supp(\vec{r}_\mu)]$ is at least 3. By Theorem 4.4.3, M^G is not realizable. It conflicts with the assumption. □

Proof of Lemma 5.6.4 is as follows.

Proof. Since $L_{max}(v_j) \neq L_{min}(v_j)$, there is a row $\vec{r}_{\mu_1}$ such that $M^G[\mu_1][L_{max}(v_j)] = 2$ and $M^G[\mu_1][j] = 2$, and a row $\vec{r}_{\mu_2}$ such that $M^G[\mu_2][L_{max}(v_j)] = 1$ and $M^G[\mu_2][j] = 2$. Because $M^G[\mu_1][j] = M^G[\mu_2][j] = 2$, vertices on the path(s) between $v_{L_{max}(v_j)}$ and v_j are 2 in both $\vec{r}_{\mu_1}$ and $\vec{r}_{\mu_2}$. If any one of those vertices, e.g., v_k, has different order with $v_{L_{max}(v_j)}$, then $M^G[\{\mu_1, \mu_2\}][\{L_{max}(v_j), k\}]$ is not realizable. So M^G is also not realizable. Then every pair of vertices on the path(s) between $v_{L_{max}(v_j)}$ and v_j has the same order. □

Proof of Theorem 5.8.1 is as follows.

Proof. Suppose M^H fits a perfect phylogeny T. On T, we call those nodes that are labelled by rows of M^H as labelled nodes, and those nodes that are not labelled by rows of M^H as unlabelled nodes.

As we explained in Section 3.3, a node (if it is labelled) and its descendants that are labelled consist of the support of a column. For those labelled nodes, they are related with distinct columns of M^H. But there may exist some other unlabelled nodes related with different columns too. See Figure A.1. Then $n \leq m$.

Let v be an unlabelled node on T. Then v is an internal node on T. If there is only one child of v that is labelled, denoted as v_c, then the columns related with v and v_c are same. Of course, if v has a child labelled, the columns related with v and the child are also same. So in M^H, the number of distinct columns that are related with those unlabelled nodes is less than half of the number of columns that are related with labelled nodes. See Figure A.1. It also means that, the number of distinct columns related with unlabelled nodes is less than half of the number of distinct rows in M^H. So $\lceil \frac{2}{3} \rceil m \leq n$. □

Proof of Theorem 5.8.2 is as follows.

Proof. Let M^H be a feasible expansion of M^G. By Theorem 5.8.1, the number of different rows is less or equal to 2/3 of the number of different columns in M^H. M^H and M^G have the same number of different columns. By the HI problem, the number of rows in M^G is at least $\frac{1}{9}m^2$. □

Proof of Lemma 5.10.2 is as follows.

Proof. In that case, there should have coloring conflict (one arc must be colored in two different colors) anyway. □

Proof of Theorem 5.10.3 is as follows.

Proof. For each component, we have two choice to color. That is the reason the number of solutions is 2^k. □

Bibliography

[1] International Human Genome Sequence Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860-921, February 2001.

[2] Ian Anderson. Combinatorics of Finite Sets. Oxford, England: Oxford University Press, p. 38, 1987.

[3] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. Technical report, UC Davis, Department of Computer Science, 2002.

[4] V. Bafna and D. Gusfield and G. Lancia and S. Yooseph. Haplotyping as Perfect Phylogeny: A Direct Approach. *Journal of Computational Biology*, Vol. 10, No. 3 (2003), pp. 323-340.

[5] Garrett Birkhoff. Lattice Theory, 3rd ed. Providence, RI: Amer. Math. Soc., 1967.

[6] R. E. Bixby and D. K. Wagner. An almost linear-time algorithm for graph realization. *Mathematics of Operations Research*, 13:99-123, 1988.

[7] Gary Chartrand. Directed Graphs as Mathematical Models. New York: Dover, 1985.

[8] R.H. Chung and D. Gusfield. Empirical Evaluation of Perfect Phylogeny Haplotypers and Haplotyping. *Proceedings of the 2003 Cocoon Conference*, published by Springer in the LNCS series.

[9] R.H. Chung and D. Gusfield. Perfect phylogeny haplotyper: Haplotype inferral using a tree model. *Bioinformatics*, 19(6):780-781, 2003.

[10] A. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol*, 7:111-122, 1990.

[11] A. Clark, K. Weiss, and D. Nickerson et. al. Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. *Am. J. Human Genetics*, 63:595-612, 1998.

[12] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics,* 29:229-232, 2001.

[13] P. Damaschke. Fast Perfect Phylogeny Haplotype Inference. *14th Symp. on Fundamentals of Comp. Theory FCT'2003, LNCS 2751*, 183-194, 2003.

[14] P. Damaschke. Incremental haplotype inference, phylogeny and almost bipartite graphs. *2nd RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotypes*, pre-proceedings, 1-11, 2004.

[15] E. Eskin, E. Halperin, and R. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. Technical report, UC Berkeley, Computer Science Division (EECS), 2002.

[16] Peter C. Fishburn. Interval Orders and Interval Sets: A Study of Partially Ordered Sets. New York: Wiley, 1985.

[17] L. Friss, R. Hudson, A. Bartoszewics, J. Wall, T. Donfalk, and A. Di Rienzo. Gene conversion and differential population histories may explain the contrast between polymorphism and linkage disequilibrium levels. *Am.J.of Human Genetics,* 69:831-843, 2001.

[18] M. Fullerton, A. Clark, Charles Sing, and et. al. Apolipoprotein E variation at the sequence haplotype level: implications for the origin and maintenance of a major human polymorphism. *Am. J. of Human Genetics*, pages 881-900, 2000.

[19] F. Gavril and R. Tamari. An algorithm for constructing edge-trees from hypergraphs. *Networks*, 13:377-388, 1983.

[20] Anthony J.F. Griffiths, and et. al. Introduction to Genetic Analysis, Eight Edition, W.H. Freeman And Company. New York., 2005.

[21] D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19-28, 1991.

[22] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

[23] D. Gusfield. A practical algorithm for deducing haplotypes in diploid populations. In Proceedings of *8'th International Confernece on Intelligent Systems in Molecular Biology*, pages 183-189. AAAI Press, 2000.

[24] D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *Journal of computational biology*, 8(3), 2001.

[25] D. Gusfield. Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions (Extended Abstract). In *Proceedings of RECOMB 2002: The Sixth Annual International Conference on Computational Biology,* pages 166-175, 2002.

[26] D. Gusfield, S. Eddhu, C. Langley. Optimal, Efficient Reconstruction of Phylogenetic Networks with Constrained Recombination *Journal of Bioinformatics and Computational Biology*, Vol. 2 no. 1 (2004) p. 173-213

[27] D. Gusfield. An Overview of Combinatoric Methods for Haplotype Inference. *Lecture Notes in Computer Science*, vol. 2983, Spirnger, p. 9-25, 2004.

[28] B. V. Halldorsson et al. Combinatorial problems arising in SNP and haplotype analysis. In C. Calude, M. Dinneen, and V. Vajnovski, editors, *Discrete Mathematics and Theoritical Computer Science. Proceedings of DMTCS '03 Conference*, volumn 2731 of *Springer Lecture Notes in Computer Science.*

[29] E. Halperin and E. Eskin. Haplotye reconstruction from genotype data using Imperfect Phylogeny. *Bioinformatics*, 20:1842-1849, 2004.

[30] E. Halperin and R. M. Karp. Perfect Phylogeny and Haplotype Assignment. In *proceedings of RECOMB 2004: The Eighth Annual International Conference on Research in Computational Molecular Biology*, page 10-19, 2004.

[31] L. Helmuth. Genome research: Map of the human genome 3.0. *Science,* 293(5530): 583-585, 2001.

[32] R. Hudson. Gene genealogies and the coalescent process. *Oxford Survey of Evolutionary Biology*, 7:1-44, 1990.

[33] S. Lin, D. Cutler, M. Zwick, and A. Cahkravarti. Haplotype inference in random population samples. *Am. J. of Hum. Genet.*, 71:1129-1137, 2003.

[34] T. Niu, Z. Qin, X. Xu, and J.S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am. J. Hum. Genet*, 70:157-169, 2002.

[35] Yunkai Liu and Cun-Quan Zhang. A linear solution for haplotype perfect phylogeny problem (Poster). *Proceeding of 7th Annual Conference on Computational Genomics*, October, 2004.

[36] Yunkai Liu and Cun-Quan Zhang. A linear solution for haplotype perfect phylogeny problem (Extended Abstract). *Advances in Bioinformatics and its Applications*, spring 2005.

[37] S. Orzack, D. Gusfield, and V. Stanton. The absolute and relative accuracy of haplotype inferral methods and a consensus approach to haplotype inferral. Abstract Nr 115 in Am. Society of Human Genetics, Supplement 2001.

[38] N. Patil, A. J. Berno, and *et. al.* Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(5547):1669-1670, 2001.

[39] Steven Skiena. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Reading, MA: Addison-Wesley, p. 241, 1990.

[40] J. C. Stephens and et al. Haplotype variation and linkage disequilibrium in 313 human genes. *Science,* 293:489-493, 2001.

[41] S. Tavare. Calibrating the clock: Using stochastic processes to measure the rate of evolution. In E. Lander and M. Waterman, editors, *Calculating the Secretes of Life.* National Academy Press, 1995.

[42] William T. Trotter. Combinatorics and Partially Ordered Sets: Dimension Theory. Baltimore, MD: Johns Hopkins University Press, 1992.

[43] J. C. Venter, M. D. Adams, E. W. Mayers and *et al.* The sequence of the human genome. *Science*, 291(5507):1304-1351, 2001.

[44] L. Wang, K. Zhang, and L. Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology, 8:6978, 2001.*

[45] C. Wiuf. Inference on Recombination and Block Structure Using Unphased Data. *Genetics*, 166(1):537-545, January 2004.

[46] Cun-Quan Zhang. Integer Flows and Cycle Covers of Graphs, Marcel Dekker Inc. New York., 1997.