國 立 交 通 大 學

應用數學系

博 士 論 文

生物應用衍生的群試問題

Combinatorial Nonadaptive Group Testing with Biological Applications

博 士 生：陳宏賓

指導老師：傅恆霖 教授

共同指導老師：黃光明 教授

中 華 民 國 九 十 五 年 十 月

生物應用衍生的群試問題

Combinatorial Nonadaptive Group Testing with Biological Applications

博 士 生：陳宏賓　　Student : Hong-Bin Chen

指導教授：傅恆霖　　Advisor : Hung-Lin Fu

共同指導教授：黃光明　　Co-advisor : Frank K. Hwang

國 立 交 通 大 學

應用數學系

博 士 論 文

A Thesis

Submitted to Department of Applied Mathematics

College of Science

National Chiao Tung University

In Partial Fulfillment of the Requirement

For the Degree of Doctor of Philosophy

In Applied Mathematics

October 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年十月

生物應用衍生的群試問題

博士生 ： 陳宏賓　　指導教授 ： 傅恆霖

共同指導教授 ： 黃光明

國 立 交 通 大 學

應 用 數 學 系

摘　　要

在計算分子生物學裡，群試是一個實驗設計的基本工具。例如，用來有效率地找出哪些克隆(clone)含有一個特定的因子。克隆如果含有這個特定因子則稱它為正的克隆；反之，稱為負的克隆。最傳統的群試模型就是每次都可以選擇任意的一些克隆去作實驗，實驗的反應若為陽性代表這群克隆裡面有至少一個是正的；若為陰性則代表這些克隆都是負的。目標就是找出所有正的克隆。而我們將藉由群試設計來減少實驗的次數以及節省所需的時間。

在應用上，除了正和負的克隆之外，還有一種稱為抑制酶的克隆，它的功能就是會抑制正克隆的反應，導致實驗的結果呈現陰性。此外，在某些應用上，實驗的陽性反應僅僅只會發生在某些特定的克

隆同時出現時，而這種會發生陽性反應的克隆集合被稱為正的複合物。我們的目標就是要找出所有的正複合物，而這樣的模型則被稱之為複合物模型。

群試的演算法大致上可以分為兩類：逐步演算法以及非調整型演算法。逐步演算法指的是實驗是一個接著一個進行的；也就是說可以利用之前的實驗結果來安排下一個實驗。而非調整型演算法指的是所有的實驗必須事先安排好，不能參考其他的實驗結果來作調整。因此，在理論上是可以視為所有實驗可以同時進行的同步演算法。在分子生物的應用裡，大多數的實驗都是很費時的，一個實驗快則幾個小時可以完成，慢則需要好幾天，甚至是幾個星期。因此，可以同步進行的非調整型演算法是比較能夠被接受且常用的方法。

在這篇論文裡，我們主要的工作就是去設計非調整型演算法，來解決生物應用所衍生出來的一些模型。首先，我們把非調整型演算法表示成矩陣的樣子，並且定義一些矩陣的性質。在第三章，我們討論了基本群試模型的矩陣性質間的關係。第四章，我們從解碼的角度出發，去探索目前分子生物學所衍生出來的群試模型之間的異同。第五章主要是討論複合物模型。值得一提的是，在這裡我們提出了兩種對應到非調整型演算法的矩陣的建構方法，而這種矩陣也適用於後面的章節。第六章的焦點是放在有抑制酶存在的群試模型。特別的是，除

了找出所有正的克隆之外，我們還去探討如何找出所有的抑制酶，在這篇論文裡提出了第一個非調整型的演算法來達到這個目標。最後，在第七章我們研究的是群試模型的一種推廣，稱為「門檻群試」。

Combinatorial Nonadaptive Group Testing with Biological Applications

Student: Hong-Bin Chen
Department of Applied Mathematics
National Chiao Tung University
Hsinchu, Taiwan 30050

Advisor: Hung-Lin Fu
Co-Advisor: Frank K. Hwang
Department of Applied Mathematics
National Chiao Tung University
Hsinchu, Taiwan 30050

Abstract

Combinatorial group testing is a basic tool in conducting experiments of tests which can be applied to computational molecular biology. For example, in screening clone library the goal is to determine which clones in the clone library hybridize with a given probe in an efficient fashion. A clone is said to be positive if it hybridizes with the probe, and negative otherwise. In practical applications, besides positive and negative clones, there is a third category of clones called inhibitors whose effect is to neutralize positive clones. Therefore, we shall have models of group testing with or without inhibitors. Also, in applications, we may face the situation that the property of being positive or negative is defined on subsets of items instead of on individual items. Such a model is known as a complex model. The study of complex models does have a significant impact in recent years.

Group testing algorithms can be generally divided into two types, sequential and nonadaptive. A sequential algorithm conducts the tests one by one and the outcomes of all previous tests can be used to set up the later test. A nonadaptive algorithm specifies all tests in advance so that they can be conducted simultaneously; thus forbidding using the information of previous tests to design later ones. Sequential algorithms require fewer number of tests in general, because extra information helps for more efficient test designs. Nonadaptive algorithms permit to conduct all tests simultaneously, thus saving time for testing. In most applications to molecular biology, an experiment can be time-consuming. Therefore, it is much preferable to have a nonadaptive algorithm where all tests are specified in advance; thus can be conducted simultaneously.

In this thesis, we first introduce a few types of matrices such as separable or disjunct matrices and then a connection between separability and disjunctness will

be provided in Chapter 3. Chapter 4 reviews various models in molecular biology by focusing on the angle of decoding. Chapter 5 studies the complex model, and provides two methods to construct generalized disjunct matrices. Chapter 6 focuses on group testing with inhibitors. In particular, we study a generalized problem that is to also identify the inhibitors besides the positive clones. Finally, in Chapter 7 we have made some contributions in threshold group testing.

Contents

Chapter 1

Introduction

1.1 The History of Group Testing

Group testing has been around for sixty years. The date may trace back to an event, World War II, in 1942 or early 1943. The origin of group testing is usually credited to a single person, Robert Dorfman [16]. It started as an idea to screen large number of blood tests for syphilis economically. When such needs subsided, group testing stayed dormant for many years until it was revived with needs for new industrial use. Sobel and Groll [47], two Bell Laboratories scientists, gave the phrase "group testing" new meaning by introducing many industrial applications for future study in their 74-page paper. Dorfman, as well as Sobel and Groll, studied group testing under the probabilistic models. Namely, a probability model is used to describe the distribution of defectives, and the goal is to minimize the expected number of tests required to identify the set of defectives.

Li [37] started to consider *combinatorial group testing* where the presumed knowledge on the set of defectives is that it must be a member, called a *sample*, of a given family called a *sample space*. For instance, the sample space could consist of all d-subsets of the n items when the presumed knowledge is that there are exactly d defectives among the n items. We will refer to this space as the $S(d, n)$ space

while the $S(\overline{d}, n)$ space specifies that d is an upper bound of defectives. In the classic combinatorial group testing problem, a deterministic model is used and the goal is usually to minimize the number of tests required under a worst-case scenario.

Since Li, combinatorial group testing has been studied alongside with the probabilistic group testing. Later, group testing is of interest in chemical and biological testing, DNA mapping, and also in several computer science applications. Many aspects of group testing have been studied in depth, such as experiment designs, coding theory, multiple access communication, among others. Here we refer to the book by Du and Hwang [17] for an overview of the vast literature. Many further developments can also be found in their new book *Pooling Designs and Nonadaptive Group Testing — Important Tools for DNA Sequencing* [18].

1.2 Goals

First of all, we give a brief description of the basic model of combinatorial group testing. Consider a set N of n items consisting of at most d positive (used to be called defective) items with the others being negative (used to be called good) items. Typically d is much smaller than n. A group test, sometimes called a pool, can be applied to an arbitrary subset S of items with two possible outcomes; a negative outcome indicates all items in S are negative, while a positive outcome indicates otherwise, i.e., there exists at least one positive item in S, not knowing which or how many. Let P denote the set of all positive items. The problem is to identify all items in P.

Group testing algorithms can be roughly divided into two types, sequential and nonadaptive. A *sequential* algorithm conducts the tests one by one and the outcomes of all previous tests can be used to set up the later test. A *nonadaptive* algorithm

specifies all tests in advance so that all tests can be conducted simultaneously; thus forbidding using the information of previous tests to design later tests. Sequential algorithms require fewer number of tests in general, because extra information help for more efficient test designs. Nonadaptive algorithms permit to conduct all tests simultaneously, thus saving time for testing.

Typically, the main concern of group testing is to minimize the number of tests required to identify all positive items. Therefore, sequential algorithms have dominated the literature. But in the applications to molecular biology, it is another thing; while minimizing the number of tests is still important, two other goals emerge.

In the applications to molecular biology, an experiment corresponding to a group test could take several hours or even several days. Thus, it is impractical to perform the experiments sequentially and great importance is attached to *nonadaptive group testing algorithms*, also called *pooling designs* in the molecular biology literature, in which all experiments are performed simultaneously. Sometimes for a given set of parameters a pooling design cannot be found or it consumes too many tests, then one has to seek for 2-stage or k-stage designs for small k. Note that pooling designs lead to an attached decoding problem: How many computations are needed to identify the positive items from the outcomes of all tests? In general, there is a trade-off between the time complexity for decoding and the number of tests needed.

Another feature of biological experiments is that errors in the outcomes cannot be ignored. With experimental errors, test outcomes may consist of false negative outcomes and false positive outcomes. The former means that a test yields a negative outcome when a pool contains at least one positive clone. Likewise, the latter means that a test yields a positive outcome when a pool contains no positive clones. In practice, the decoding issue becomes even more difficult due to experimental errors.

So the second goal is to control the experimental errors, which has rarely been studied in the classical group testing literature, so that even though errors occur the positive items can still be identified. Thus for each model we will consider its error-tolerant version.

1.3 Applications to Molecular Biology

Recent advances in molecular biology, especially the success of the Human Genome Project, have made the study of gene functions more popular. The study of gene functions requires a high quality DNA library, which is a collection of the copies of DNA segments, called *clones*. In screening a clone library, the goal is to determine which clones in the clone library hybridize with a given *probe* in an efficient fashion. A clone is said to be positive if it hybridizes with the probe, and negative otherwise.

In some applications, beside positive and negative clones, there is a third category of clones called *inhibitors* whose effect is to neutralize positive clones. That means the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive clones in the pool. An example of inhibitors is an enzyme inhibitor, which is a molecule that binds to the active site of an enzyme during the reaction process and then prevents the success of this process. The inhibitor model was first introduced by Farach et al. [27], and studied further in [5, 6, 9, 32]. The usual concern in literature is to identify all positive clones. Another interesting problem here is to also identify the inhibitors, besides the positive ones.

Suppose the items are molecules. Then a biological function in some other applications may depend on the presence of a subset of molecules, called a *complex*, i.e., the property of being positive or negative is defined on subsets of molecules, instead of on individual molecules. Such a model is usually referred to as the complex model,

first introduced by Torney [51]. The problem is described as follows. Consider a set N of n molecules and an unknown family $P = \{P_i\}$ of subsets of N where the joint appearance of all molecules in such a subset causes a certain given biological phenomenon defined as a positive outcome. A set of molecules which is a candidate of a member of P is called a complex while members of P are called positive complexes. The goal is to identify P from a given set of complexes. Treating each molecule as a vertex and a complex as an edge, the complex model can be easily fitted into the framework of *graph testing*, learning the hidden subgraph P in a given graph. The complex model is also related to other problems such as superimposed codes and secure key distribution, among others [1, 10, 43, 20, 51].

1.4 Nonadaptive Group Testing

A nonadaptive group testing algorithm can be represented by an incidence matrix $M = [m_{ij}]$ where rows are labelled by pools, columns by items, and $m_{ij} = 1$ if and only if item j is in pool i. For convenience, we treat a column C_j as the characteristic vector of subset $\{i : m_{ij} = 1\}$, i.e., the set of row indices where C_j has 1-entries. Then we can talk about the union $\cup S$ and the intersection $\cap S$ of a set S of columns.

In the classic group testing problem, three types of binary matrices have been the major tools in understanding and constructing a pooling design.

Definition 1.4.1. A matrix is *d-separable* if $\cup D \neq \cup D'$ for any two distinct d-sets D, D' of columns, i.e., no two unions of d columns are same.

Definition 1.4.2. A matrix is $\overline{d}$*-separable* if $\cup D \neq \cup D'$ for any two distinct sets D, D' of columns with $|D|, |D'| \leq d$, i.e., no two unions of at most d columns are same.

Definition 1.4.3. A matrix is *d-disjunct* if for any $d + 1$ columns $C_0, C_1, \cdots, C_d$,

$$C_0 \not\subseteq \bigcup_{i=1}^{d} C_i,$$

i.e., no column is contained in the union of any other d columns.

These matrices have been studied elsewhere under other names. The d-separable matrix was first studied by Erdős and Moser [25] for $d = 2$. Frankl and Fűredi [28] called a d-separable matrix a *union-free hypergraph* by treating rows as vertices and columns as edges of a hypergraph (then the boolean sum of columns becomes the union of edges). The $\overline{d}$-separable matrix was first studied by Kautz and Singleton [35] as a special kind of code named UD_d (uniquely decipherable code of order d). The d-disjunct matrix was also first studied by them under the name of ZFD_d (zero-false-drop code of order d).

These properties are now explained in terms of pooling designs. Consider the sample space $S(d, n)$ where exact d positive items are present. The d-separability property implies that each sample in the sample space $S(d, n)$ induces a different outcome vector. By matching the outcome vector with the samples in $S(d, n)$, the d positive items can be identified. Moreover, the d-separability is also a necessary condition for a matrix M to be able to identify the d positive items. Similarly, the $\overline{d}$-separability implies that samples in $S(\overline{d}, n)$, where at most d positive items are present, are distinguishable. Although the $\overline{d}$-separability (d-separability) property is able to identify up-to-d (d) positive items respectively, the actual decoding algorithm can be messy. Thus, one can trade off a stronger requirement with an easier decoding. The d-disjunctness property offers such a trade-off. In a d-disjunct matrix, a negative item must appear in a negative pool, thus can be certainly identified as negative. Consequently, one does not have to build and look up a table mapping outcomes to samples in $S(d, n)$ or $S(\overline{d}, n)$. In fact, the d-disjunctness property substantially

reduces the time complexity of decoding from $O(tn^d)$ down to $O(tn)$, where t is the number of pools needed.

1.5 An Outline of the Thesis

As introduced, three types of binary matrices have been found to be major tools in understanding and constructing pooling designs: d-disjunct, $\overline{d}$-separable and d-separable. While there exists a simple decoding for d-disjunct matrices, only brute-force methods are known for the other two. In addition, the implications from the first two matrices to the last one are well documented. Chapter 3 gives an implication of the other direction for the first time. Moreover, we identify structures in the later two matrices which lead to significant improvements for decoding complexity.

This thesis will focus on applications to molecular biology. Several biological models will be discussed. So far, there have been a number of related surveys in this area [2, 23, 17, 45]. To our best knowledge, however, none of which takes a look at group testing through the angle of decoding algorithm, namely, how P is identified from the outcomes of the pooling designs. Chapter 4 reviews several common models in molecular biology by focusing on decoding, namely, giving a comparative study of how the problem is solved in each of these models. From this angle, we see the simplicity and integrity of the pooling design theory in the sense that all models share the same basic structure in their decoding algorithms. We also see how the differences in the models are reflected in the modifications of the basic structure.

Chapter 5 studies the complex model. We propose two explicit methods to construct generalized disjunct matrices, which correspond to the pooling designs geared to the complex model. We also give a lower bound for the number of tests required by a combinatorial argument.

Chapter 6 studies the inhibitor model. We first strengthen the necessary condition by De Bonis and Vaccaro, and give a pooling design which is comparable to the best known results in the number of tests required, but improving significantly in decoding complexity. Furthermore, we study a generalized problem that is to also identify the inhibitors, besides the positive items.

Chapter 7 presents a natural generalization of group testing, called *threshold group testing*, which is first proposed by Damaschke [15]. In the threshold group testing, a group test gives a positive (negative) answer if the pool contains at least u (at most l) positive items, and an arbitrary answer if the number of positive items is between these fixed threshold l and u, $l < u$. Obviously, the classical group testing is a special case, $l = 0$ and $u = 1$, of this model. In this chapter, we first give a pooling design which improves a main result in Damaschke's paper [15] on the number of tests needed. Furthermore, we study a synthetic model in which the inhibitor model and the threshold model are combined together.

Chapter 2

Preliminaries

2.1 Group Testing

In this section, we adopt some notations and definitions of properties of matrices discussed in the rest of this thesis.

Consider an incidence matrix $M = [m_{ij}]$ where rows are labelled by pools, columns by items, and $m_{ij} = 1$ if and only if item j is in pool i. For convenience, we treat a column C_j as the characteristic vector of subset $\{i : m_{ij} = 1\}$, i.e., the set of row indices where C_j has 1-entries. Then we can say a column is contained in another, i.e., for two columns $C = (v_1, v_2, \cdots, v_t)$ and $C' = (v'_1, v'_2, \cdots, v'_t)$, we say $C \subseteq C'$ if and only if $v_i \leq v'_i$ for all $1 \leq i \leq t$. We say a row R_i intersects a column C_j if and only if $m_{ij} = 1$.

For any two columns C and C', we denote $C \cup C'$ as the union of C and C', which is nothing but the boolean sum. Similarly, $C \cap C'$ denotes the intersection of C and C'. Likewise, for a set X of columns, denote $\cup X$ as the union and $\cap X$ as the intersection of all columns in X.

A pool with a negative/positive outcome is called a negative/positive pool. For a set P of positive items, denote V as the outcome vector, i.e., $V = \cup P$ in case of error-free. Sometimes, we use $+$ (or $-$) as a superscript to denote a positive (or

negative) column, i.e., a column representing a positive (or negative) item.

For a column C, define $t_0^V(C) \equiv |C \setminus V|$ and $t_1^V(C) \equiv |C \cap V|$, i.e., the number of negative (positive) pools in which column C appears, respectively. For a subset X of columns, define $t_0^V(X) \equiv |(\cap X) \setminus V|$ and $t_1^V(X) \equiv |(\cap X) \cap V|$, i.e., the number of negative (positive) pools in which all columns in X appear, respectively.

Definition 2.1.1. A matrix is $(d; z)$-*disjunct* if for any $d+1$ columns $C_0, C_1, \cdots, C_d$,

$$\left| C_0 \setminus \bigcup_{i=1}^{d} C_i \right| \geq z.$$

That means for any $d+1$ columns $C_0, C_1, \cdots, C_d$, there exist at least z rows in which C_0 has 1-entries and the other d columns have 0-entries. When $z = 1$, $(d; z)$-disjunctness is equivalent to d-disjunctness.

Definition 2.1.2. A matrix is (d, c)-*disjunct* if for any $d+c$ columns $C_1, C_2, \cdots, C_{d+c}$,

$$\bigcup_{i=1}^{c} C_i \not\subseteq \bigcup_{i=c+1}^{d+c} C_i.$$

The interpretation is that no union of c columns is contained in the union of any other d columns. Note that $(d, 1)$-disjunctness is equivalent to d-disjunctness, and also (d, c)-disjunctness implies (d, t)-disjunctness for $t > c$.

Definition 2.1.3. A matrix is $(d, r]$-*disjunct* if for any $d+r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\bigcap_{i=1}^{r} C_i \not\subseteq \bigcup_{i=r+1}^{d+r} C_i.$$

That means for any $d + r$ columns there exists a row where each of the r designated columns has a 1-entry and each of the other d columns has a 0-entry.

Definition 2.1.4. A matrix is $(d, r; z]$-*disjunct* if for any $d+r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\left| \bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i \right| \geq z.$$

That means for any $d+r$ columns there exist z rows where each of the r designated columns has 1-entries and each of the other d columns has 0-entries.

Notice that $(d, r; 1]$-disjunctness is equivalent to $(d, r]$-disjunctness, and $(d, 1; z]$-disjunctness is equivalent to $(d; z)$-disjunctness. All previous definitions of disjunctness properties are special cases of the $(d, r; z]$-disjunctness property except the (d, c)-disjunctness property.

We present a few easy preliminary lemmas that will be used later.

Lemma 2.1.5. *$M = [m_{ij}]$ is a $(d, r; z]$-disjunct matrix if and only if $\overline{M} = [\overline{m}_{ij}]$ is a $(r, d; z]$-disjunct matrix, where $\overline{m}_{ij} = 1 - m_{ij}$.*

Proof. The proof follows by the fact that a $(d, r; z]$-disjunct matrix can be obtained from an $(r, d; z]$-disjunct matrix by interchanging all the 1's and 0's. ∎

Lemma 2.1.6. *Let M be a $(d, r; z]$-disjunct matrix, and M^1 be obtained from M by deleting a column and all the rows intersecting this column. Then M^1 is $(d-1, r; z]$-disjunct.*

Proof. Let C_{d+r} be the column deleted from M. Suppose to the contrary that M^1 is not $(d-1, r; z]$-disjunct. Then in M^1 there exist $d+r-1$ columns $C_1, C_2, \cdots, C_{d+r-1}$, such that $\left|\bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r-1} C_i\right| < z$. It is easy to see that tracing back to M,

$$\left|\bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i\right| < z,$$

violating the $(d, r; z]$-disjunctness. Hence, M^1 is $(d-1, r; z]$-disjunct. ∎

Lemma 2.1.7. *Let M be a $(d, r; z]$-disjunct matrix, and M^0 be obtained from M by deleting a column and all the rows not intersecting this column. Then M^0 is $(d, r-1; z]$-disjunct.*

Proof. The proof follows by a similar argument to that of Lemma 2.1.6. ∎

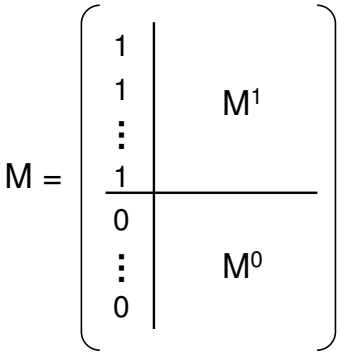

Figure 2.1: M is $(d, r; z]$-disjunct implies M^1 is $(d-1, r; z]$-disjunct and M^0 is $(d, r-1; z]$-disjunct.

2.2 Hypergraphs

Given a finite set X, a hypergraph $H = (X, \mathcal{F})$ is a family $\mathcal{F} = \{E_1, E_2, \cdots, E_m\}$ of subsets of X. The elements of X are called vertices, and the subsets E_i's are the edges of the hypergraph H.

The *rank* $r(H)$ of H is $\max|E_j|$. A hypergraph satisfying $\max|E_j| = \min|E_i|$ is said to be *uniform*. A hypergraph is said to be a *rank-r graph* if each edge contains at most r vertices, and a hypergraph is an *r-graph* if each edge contains exactly r vertices.

For $u \in X$, define the degree $d_H(u)$ of u to be the number of edges containing u. The maximum degree of H will be denoted by $\Delta(H) = \max_{v \in X} d_H(v)$. A hypergraph H in which all vertices have same degree is said to be *regular*, i.e., $d_H(u) = d_H(v)$ for all $u, v \in X$.

A vertex cover of H is a vertex subset intersecting all edges E_i's. Further, a z-cover of H is a subset $C \subseteq X$ such that $|C \bigcap E_i| \geq z$ for all i. Let $\tau_z(H)$ denote the minimum size among all z-covers of H.

For $z = 1$, Lovász [39] prove that

Lemma 2.2.1. $\tau_1(H) < \dfrac{|X|}{\min_{E\in\mathcal{F}}|E|}(1+\ln\Delta)$.

2.3 Codes

First, we give a brief description of finite fields. The *order* of a finite field is the number of elements in it. It is well-known that n must be a prime power. Every finite field F of order n is denoted by $GF(n)$.

A *code* consists of a set of vectors, called *codewords*. The *length* of a codeword $\mathbf{u} = (u_1, u_2, \cdots, u_t)$ is t. The *size* of a code is the number of codewords in it. A code of length t and size n can be represented by a matrix of size $t \times n$.

Definition 2.3.1. A binary $t \times n$ matrix $M = [m_{ij}]$ is called a *superimposed* (d, r)-*code* if for any two sets $\mathscr{X}, \mathscr{Y}$ of columns such that $|\mathscr{X}| = d$, $|\mathscr{Y}| = r$ and $\mathscr{X} \cap \mathscr{Y} = \emptyset$, there exists a row i for which

$$m_{ij} = 1 \qquad \text{for all } j \in \mathscr{Y}, \qquad m_{ij} = 0 \qquad \text{for all } j \in \mathscr{X}.$$

The *Hamming distance* between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as the number of positions in which they differ.

Definition 2.3.2. *The Maximum Distance Separable Code (MDS-code)* with parameters (q, k, t) is a q-ary code of size $n = q^k$, length t and the Hamming distance $t - k + 1$.

For any integer $k \geq 2$ and a prime power $q \geq k-1$ there exists an MDS-code with parameters $(q, k, q+1)$, which is a Reed-Solomon code [40]. Note that finite fields of order q play the key role of constructing Reed-Solomon codes.

Chapter 3

Separable Matrices

As mentioned in Chapter 1, three types of binary matrices have become the major tools in constructing a pooling design:

1. M is *d-separable* if no two unions of d columns are same.

2. M is *$\overline{d}$-separable* if no two unions of at most d columns are same.

3. M is *d-disjunct* if no column is contained in the union of any other d columns.

Let n denote the number of columns in the given matrix. It is easily seen from definitions that d-separability ($\overline{d}$-separability, d-disjunctness) implies k-separability ($\overline{k}$-separability, k-disjunctness) for $1 \leq k < d < n$, respectively. Kautz and Singleton [35] proved the following relations.

$$\overline{d+1}\textbf{-separability} \Rightarrow d\textbf{-disjunctness} \Rightarrow \overline{d}\textbf{-separability} \Rightarrow d\textbf{-separability}$$

In particular, d-disjunctness $\Rightarrow$ $\overline{d}$-separability with the option of dropping an arbitrary row. Note that the relations between the three types of matrices miss a link from d-separability to k-disjunctness or $\overline{k}$-separability for some $k < d$. In the following section, we (Chen and Hwang [13]) give such a link for the first time.

3.1 From $2d$-Separability to d-Disjunctness

Notice that $\cup S$ denote the union (or the boolean sum) of a set S of columns. Chen and Hwang [13] show the following equivalence.

Lemma 3.1.1. *Let M be a d-separable matrix. Then M is $\overline{k+1}$-separable, $1 \leq k \leq d-1$, if and only if M is k-disjunct.*

Proof. Sufficiency.

Suppose to the contrary that there exist two distinct sets S and S' of columns in M, $|S| \leq k+1$, $|S'| \leq k+1$, such that $\cup S = \cup S'$. By the d-separability property of M, we may assume $|S| < |S'| \leq k+1$. Then there exists a column $C \in S' \setminus S$. Since $\cup S = \cup S'$, we obtain $C \subseteq B(S)$, which violates the k-disjunctness property of M.

Necessity.

Suppose M is not k-disjunct, i.e., there exist a column C and a set S of k other columns such that $C \subseteq \cup S$. Then $\cup S = \cup S'$ where $S' = S \cup \{C\}$ and $|S|, |S'| \leq k+1$. Hence M is not $\overline{k+1}$-separable. ∎

According to Lemma 3.1.1, we give a construction showing how to convert a separable matrix to a disjunct matrix by adding tests and reducing d.

Theorem 3.1.2. *Let M be $2d$-separable. Then there exists a d-disjunct matrix obtained by adding at most one row to M.*

Proof. If M is d-disjunct, then we are done. Suppose it is not. Then there exist a column C and a set S of d other columns such that $C \subseteq \cup S$. Add a row R which has a 1-entry at C and a 0-entry at each column of S to break up the containment $C \subseteq \cup S$ in M. Of course, there may exist C' and S', also with d columns, such that $C' \subseteq \cup S'$ in M. Then we break it up by using R in the same fashion. However, what we need

to show is that this procedure of setting the entries in R is not self-conflicting, i.e., there does not exist a column C such that $C \subseteq \cup S$, yet on the other hand $C \in S'$ while $\cup S' \supseteq C' \neq C$ (since then C must have a 1-entry from $C \subseteq \cup S$ and a 0-entry from $\cup S' \supseteq C'$).

Suppose to the contrary that there exist C, C', S and S' as described above with $|S| \leq d$, $|S'| \leq d$ in M. Define

$$S_0 = \{C'\} \cup S \cup S',$$
$$S_1 = S_0 \setminus \{C\},$$
$$S_2 = S_0 \setminus \{C'\}.$$

Then

$$|S_0| \equiv s \leq 2d + 1,$$
$$|S_1| = s - 1 \leq 2d,$$
$$|S_2| = s - 1 \leq 2d.$$

The fact $|S_1| = s - 1$ follows from $C \in S_0$, since $C \in S'$. Note that $S_1 \neq S_2$, but they have the same cardinality which is at most $2d$. We now show $\cup S_1 = \cup S_2$, thus violating the assumption of $2d$-separability (which implies $(s-1)$-separability).

Since the only column in S_1 but not in S_2 is C', whose 1-entries are covered by S' which is in S_2, we have $\cup S_1 \subseteq \cup S_2$.

On the other hand, the only column in S_2 but not in S_1 is C, whose 1-entries are covered by S which is in S_1. Hence $\cup S_2 \subseteq \cup S_1$. ∎

Theorem 3.1.3. *Let M be $2d$-separable. Then there exists a $\overline{d+1}$-separable matrix obtained by adding at most one row to M.*

Proof. Theorem 3.1.3 follows from Theorem 3.1.2 and Lemma 3.1.1. ∎

3.2 New Bounds

Let $t(n,d)$ denote the minimum number of tests for a d-disjunct matrix with n items, and let $t_s(n,d)$ and $t_s(n,\overline{d})$ denote the counterparts for d-separable and $\overline{d}$-separable matrices, respectively. It is well known [17] that

1. $t(n,d) = \Omega(d^2 \log n / \log d)$;

2. $t(n,d) = O(d^2 \log n)$.

Since the d-disjunctness is stronger than d or $\overline{d}$-separability, the upper bound of $t(n,d)$ remains an upper bound of $t_s(n,d)$ and $t_s(n,\overline{d})$, respectively. However, the lower bound is not preserved. Previously, there was no good argument for lower bounds of $t_s(n,d)$ and $t_s(n,\overline{d})$ except

$$t_s(n,d) = \Omega(d \log n)$$

from the simple-minded argument that the number of distinct d-subsets, $\binom{n}{d}$, cannot exceed the number of distinct outcomes, 2^t. Theorem 3.1.3 shows immediately that $t_s(n,d)$, $t_s(n,\overline{d})$ and $t(n,d)$ have the same lower bound in the order of magnitude.

Theorem 3.2.1. $t_s(n,\overline{d}) \geq t_s(n,d) = \Omega(d^2 \log n / \log d)$.

In a sequential group-testing algorithm, the tests are done sequentially which means we can use outcomes from previous tests to determine what to test next. Let $t'(n,\overline{d})$ or $t'(n,d)$ denote the minimum number of tests required to identify at most d or exact d positive columns among n columns, respectively. Hwang, Song and Du [33] proved

Theorem 3.2.2. $t'(n,\overline{d}) - t'(n,d) \leq 1$.

It is easy to see that $t_s(n, \overline{1}) - t_s(n, 1) \leq 1$. Setting $d = 1$ in Theorem 3.1.3, Chen and Hwang [13] prove the following result.

Theorem 3.2.3. $t_s(n, \overline{2}) - t_s(n, 2) \leq 1$.

Note that, for $d > 2$, the difference between $t_s(n, \overline{d})$ and $t_s(n, d)$ remains unknown.

3.3 Bounding the Number of Items Appearing Only in Positive Pools

As introduced in Chapter 1, a d-separable matrix or a $\overline{d}$-separable matrix has no simple decoding. The only known decoding is the brute-force method [52] by computing the output vectors of all candidate sets of positive items (this can be done in advance) and check which matches the actual outcome vector. Notice that items appearing in a negative pool are not positive; thus they are not considered in a candidate set. In this section, we will bound the number of columns not appearing in any negative pool to reduce the number of candidate sets.

Let M be a $t \times n$ d-separable (or $\overline{d}$-separable or d-disjunct) matrix, $P = \{P_1, \cdots, P_d\}$ the set of positive items, and V the outcome vector corresponding to $\{P_1, \cdots, P_d\}$, i.e., $V = \bigcup_{i=1}^{d} P_i$. Let T_0 and T_1 denote the sets of negative pools and positive pools, respectively, with $|T_0| = t_0$ and $|T_1| = t_1$ where $t_0 + t_1 = t$. Let $M_1(M_0)$ be the $t_1 \times n_1 (t_0 \times n_0)$ submatrix of M such that the rows are $T_1(T_0)$ and the columns are those which have no 1-entries in $T_0(T_1)$, respectively. Note that given a matrix M, our aim is to bound n_1.

Chen, Li and Hwang [14] observed the following relations between M and M_1.

Lemma 3.3.1. *If M is d-separable (or $\overline{d}$-separable or d-disjunct), then M_1 is d-separable (or $\overline{d}$-separable or d-disjunct), respectively.*

Proof. Since a column in M_1 preserves all the 1-entries in M, M_1 inherits the property of M. ∎

An immediate consequence of Lemma 3.3.1 is that bounds of n for a d-separable (or $\overline{d}$-separable or d-disjunct) matrix can be used to bound n_0 and n_1. For a d-disjunct matrix, Füredi [29] proved $n \leq d \cdot 2^{4t/d^2}$ by a combinatorial argument. D′yachkov and Rykov [22] gave the asymptotic bound $n \leq d \cdot 2^{2t/d^2}(1 + o(1))$.

Bounds of n_1 for d-separable and $\overline{d}$-separable matrices can be obtained through their relation with d-disjunct matrices. Kautz and Singleton [35] proved that a $\overline{d}$-separable matrix is a $(d-1)$-disjunct matrix. Thus

Theorem 3.3.2. *Suppose M is $\overline{d}$-separable. Then $n_1 \leq (d-1) \cdot 2^{4t_1/(d-1)^2}$. Also $n_1 \leq (d-1) \cdot 2^{2t_1/(d-1)^2}(1 + o(1))$ asymptotically.*

From Theorem 3.1.2, we have proved that a d-separable matrix can be converted to a $\lfloor d/2 \rfloor$-disjunct matrix by adding a row. Thus we obtain

Theorem 3.3.3. *Suppose M is d-separable. Then $n_1 \leq \lfloor d/2 \rfloor \cdot 2^{4(t_1+1)/\lfloor d/2 \rfloor^2}$. Also $n_1 \leq \lfloor d/2 \rfloor \cdot 2^{2(t_1+1)/\lfloor d/2 \rfloor^2}(1 + o(1))$ asymptotically.*

Ironically, the case that M being d-disjunct does not have any analogous result. This is because that a much stronger result is well known. Suppose the actual number of positive clones is $p \leq d$. Then $n_1 = p$ [35].

Chen, Li and Hwang [14] observed that M_1 actually satisfies an additional constraint that there exists a set D of d columns in M_1 such that the union of D intersects all rows in M_1(any set of d columns containing all positive clones will do). We make use of this constraint to derive a new bound for the d-separable case.

Let N_1 denote the set of columns in M_1 and let $D = \{D_1, \cdots, D_d\}$. Define $D_i^* = D_i \setminus \bigcup_{j \neq i} D_j$ for $1 \leq i \leq d$.

We prove the following results.

Lemma 3.3.4. *$C \cap D_i^* \neq C' \cap D_i^*$ for all C, $C' \in N_1 \setminus D$ and $1 \leq i \leq d$.*

Proof. Suppose to the contrary that there exist C, C' and i such that

$$C \cap D_i^* = C' \cap D_i^*.$$

Since the union of D intersects all rows in M_1,

$$C \setminus D_i^* \subseteq \bigcup_{j \neq i} D_j \text{ and } C' \setminus D_i^* \subseteq \bigcup_{j \neq i} D_j.$$

Thus we have $C \cup (\bigcup_{j \neq i} D_j) = C' \cup (\bigcup_{j \neq i} D_j)$, violating the assumption of d-separability.∎

Theorem 3.3.5. *For a d-separable matrix M, $n_1 \leq d + 2^{\lfloor t_1/d \rfloor} - 1$.*

Proof. Clearly,

$$\min_{1 \leq j \leq d} |D_j^*| \leq \lfloor t_1/d \rfloor.$$

Without loss of generality, assume D_i^* achieves the minimum. By Lemma 3.3.4, all columns in $N_1 \setminus D$ have distinct intersections with D_i^*, hence there are at most $2^{|D_i^*|} \leq 2^{\lfloor t_1/d \rfloor}$ of them. But we have to subtract one since the intersection number cannot be $|D_i^*|$. For otherwise, the union of that column with $\bigcup_{j \neq i} D_j$ equals D, violating the assumption of d-separability . ∎

Compare the two bounds in Theorem 3.3.3 and Theorem 3.3.5, the bound in Theorem 3.3.5 is better for $d \leq 16$, which is usually the case in biological applications.

Theorem 3.3.6. *For a $\overline{d}$-separable matrix M,*

$$n_1 \begin{cases} = p & \text{if } p \leq d-1, \\ \leq d + 2^{\lfloor t_1/d \rfloor} - 2 & \text{if } p = d. \end{cases}$$

Proof. Since M_1 is $\overline{d}$-separable, hence $(d-1)$-disjunct, the only columns in M_1 are the positive items if $p \leq d-1$.

If $p = d$, then $C \cap D_i^*$ can be neither D_i^* nor $\emptyset$ (leading to $C \cup (\bigcup_{j \neq i} D_j) = \bigcup_{j \neq i} D_j$). Hence 2 is subtracted from $2^{\lfloor t_1/d \rfloor}$. ∎

3.4 Concluding Remarks

In this chapter, we provide a link from d-separability to k-disjunctness or $\overline{k}$-separability, but not as strong as we expect, i.e., k is not large enough. Therefore the value of our link is not in its practicality in constructing efficient k-disjunct or $\overline{k}$-separable matrices from known d-separable matrices, but rather in calling awareness to the existence of such a link, so that further research can improve on it.

Establishing such a link leads to two principal consequences. The obvious one is in constructing k-disjunct or $\overline{k}$-separable matrices from known d-separable matrices, even though it seems not so efficient. The less obvious one is as described in section 3.2, an improvement of the lower bound for the number of rows in the d-separable matrices. This implies that the three types of matrices share the same lower bound in the order of magnitude.

As mentioned in Chapter 1, d-disjunct matrices have a simple decoding algorithm, namely, a column is positive if and only if it does not appear in a negative row. On the other hand, d-separable matrices or $\overline{d}$-separable matrices need fewer tests but have no simple decoding. The only known decoding was the brute-force method [52] by computing the output vectors of all candidate sets of positive items and checking which matches the actual outcome vector. Let S and $\overline{S}$ denote the sizes of the candidate sets for d-separable and $\overline{d}$-separable, respectively. Then essentially,

$$S = \binom{n}{d} \text{ and } \overline{S} = \sum_{j=0}^{d} \binom{n}{j}.$$

We show that n can be replaced by the bounds of n_1 in Theorem 3.3.3 or 3.3.5 for S, and by the bounds in Theorem 3.3.2 or 3.3.6 for $\overline{S}$ for large savings.

Chapter 4

An Overview through Decoding Algorithms

This Chapter gives an overview of several models discussed in the group testing literature. Its coverage includes the error-tolerance model, the inhibitor model and the complex model. It is worth pointing out that the angle we use to cut through these models is the decoding algorithm. From this angle, we see the simplicity and integrity of the pooling design theory in the sense that all models share the same basic structure in their decoding algorithms. We also see how the differences in the models are reflected in the modifications of the basic structure. Note that the content of this chapter was taken from Chen and Hwang [12].

Suppose V is the outcome vector. Review that, for a column C, $t_0^V(C) = |C \setminus V|$ and $t_1^V(C) = |C \cap V|$, i.e., the number of negative (positive) pools in which column C appears, respectively.

4.1 Various Models of Group Testing

4.1.1 The Basic Model

In the classic group testing problem, we consider a set N of n items consisting of at most d positive items with the others being negative items. Recall that a matrix

is said to be d-disjunct if for any $d+1$ columns $C_0, C_1, \cdots, C_d$,

$$\left| C_0 \setminus \bigcup_{i=1}^{d} C_i \right| \geq 1.$$

ITEM SELECTION(N, V, D, e)

1 **for** each item $C \in N$

2 **compute** $t_0^V(C)$

3 **if** $t_0^V(C) \leq e$

4 **then** $D \leftarrow D \cup \{C\}$

5 **return** D

Remark. The **ITEM SELECTION**(N, V, D, e) algorithm is a common decoding tool to help determine which individual item is what we need via the function $t_0^V(C)$. This algorithm returns the set D consisting of all items that appear in at most e negative pools under the outcome vector V. In this chapter all decoding algorithms, except that of complex models, have the **ITEM SELECTION** algorithm as a sub-algorithm in common, but the parameters should be geared to the need of each individual model.

d-BASIC ALGORITHM

0 **use** a d-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 **ITEM SELECTION**$(N, V, D, 0)$

Theorem 4.1.1. *The* **d-BASIC ALGORITHM** *can identify the up-to-d positive items.*

Proof. For a positive item C^+, obviously, $t_0^V(C^+) = 0$. Assume there are at most d positive items. Consider a negative item C^-, by the d-disjunctness property, there is a row intersecting C^- but none of P; thus $t_0^V(C^-) \geq 1$. Therefore we can separate all positive items from negative ones by using this algorithm. ∎

4.1.2 The Error-Tolerant Model

In this subsection, the problem is to identify the up-to-d items in P with at most e erroneous outcomes.

Review that a matrix is said to be $(d; z)$-disjunct if for any d+1 columns $C_0, C_1, \cdots, C_d$,

$$\left| C_0 \setminus \bigcup_{i=1}^{d} C_i \right| \geq z.$$

That means there exist at least z rows in each of which C_0 has a 1-entry and every C_i, $1 \leq i \leq d$, has a 0-entry.

(d, e)-E ALGORITHM

0 **use** a $(d; 2e + 1)$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 **ITEM SELECTION**(N, V, D, e)

Theorem 4.1.2. *The* (d, e)**-E ALGORITHM** *can identify the up-to-d positive items with at most e errors.*

Proof. Assume the number of errors is at most e. For each negative item C^-, by the $(d; 2e + 1)$-disjunctness property, there exist at least $2e + 1$ rows intersecting C^- but none of P. Therefore the pools corresponding to these rows must have negative

outcomes. Even for the worst case that e outcomes are erroneous, C^- still appears in at least $e+1$ negative pools, i.e., $t_0^V(C^-) \geq e+1$. On the other hand, the outcomes of the pools containing a positive item C^+ should be positive except for the occurrence of errors. Hence $t_0^V(C^+) \leq e$. Thus, we can determine, via the function $t_0^V(C)$, whether C is positive or not. ∎

4.2 Group Testing with Inhibitors

4.2.1 The Error-Tolerant Inhibitor (EI) Model

Denote I as the set of all inhibitors. In the **EI**-model, an additional assumption we make is $|I| \leq h$. Notice that the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive items in the pool.

(d, h, e)-EI ALGORITHM

```
0    use a (d + h; 2e + 1)-disjunct matrix
1    V ← the outcome vector
2    D ← ∅
3    O ← ∅
4    for every item C ∈ N
5        compute t_1^V(C)
6        if t_1^V(C) ≤ e
7            then O ← O ∪ {C}
8    for all h-subsets S ⊆ O
9        V ← V ∪ (∪S)
10       ITEM SELECTION(N \ O, V, D, e)
```

Remark. D'yachkov et al. [21] first gave a nonadaptive algorithm for the inhibitor model without erroneous outcomes. The basic idea is to restore all positive outcomes neutralized by inhibitors and their method exhaustively searches all $\binom{n}{h'}$, $h' \leq h$, h'-subsets of the n items. Hwang and Liu [32] gave a more efficient decoding algorithm with error-tolerance that can substantially reduces the number of searching operations down to $\binom{|O|}{h}$, where O is a set containing all inhibitors but no positives. Computing $t_0^V(C)$ and $t_1^V(C)$ for each item C as a prior operation, they partition the n items into four sets so that all inhibitors are separated from all positives. Here we give a simplified version. The idea of this algorithm is to first collect all inhibitors into the set O, and then identify a column C as positive if there exists one S for which $t_0^V(C) \leq e$ under the outcome vector V adjusted by S.

Theorem 4.2.1. *The* (d, h, e)**-EI ALGORITHM** *can identify all positive items under the* (d, h, e)-**EI** *model.*

Proof. To prove that (d, h, e)**-EI ALGORITHM** works for the (d, h, e)-**EI** model, what we need to show first is that O contains all inhibitors but no positives. Observe that an item which appears in at most e positive pools cannot be positive due to the $(d+h; 2e+1)$-disjunctness property. Further, even for the worst case that e outcomes are erroneous, every inhibitor appears in at most e positive pools. Hence the set O contains all inhibitors but no positives.

Consider a negative item $C^- \in N \setminus O$ and a set P of at most d positive items. For each h-subset $S \subseteq O$, by the $(d + h; 2e + 1)$-disjunctness property, there exists a $(d + h)$-set R of columns containing all positive items and S such that there are at least $2e + 1$ rows each intersecting C^- but none of R. The outcomes of the pools corresponding to these rows should be negative except for the occurrence of errors. Therefore, we can conclude that $t_0^V(C^-) \geq (2e + 1) - e = e + 1$. Hence no negative

item is selected into D.

Consider an h-subset $S \subseteq O$ containing all up-to-h inhibitors with the others being negative items. For a positive item $C^+ \in N \setminus O$, C^+ appears only in the pools in the new outcome vector V adjusted by S. For the worst case that e outcomes are erroneous, C^+ still appears in at most e negative pools, i.e., $t_0^V(C^+) \leq e$. Hence every positive item will be selected into D.

From the above discussion, the output of the (d, h, e)-**EI ALGORITHM** is the set of all positive items. ∎

4.2.2 The General Error-Tolerant Inhibitor (GEI) Model

In the simplest inhibitor model, the model discussed in Section 4.2.1, the mere existence of a single inhibitor dictates the outcome to be negative regardless of the presence of positive items. This notion has been extended to the k-inhibitor model [7] which requires the existence of k inhibitors to dictate a negative outcome. We could make even more complicate assumption that each set of k inhibitors cancel the effect of a set of g positive items, but practically, accurate information of k and g is usually not available. Thus in the general inhibitor model, we only assume the existence of some kind of cancelling effect between the inhibitors and the positive items, but no further quantifiable information. Surprisingly, a decoding algorithm exists even under such ambiguity.

A result by Chang, Chang and Hwang [9] implies that a $(d+h; 2e+1)$-disjunct matrix identifies all positive items under the (d, h, e)-**GEI** model as well as the (d, h, e)-**EI** model. The main idea is similar to that in the (d, h, e)-**EI** model, that is, restoring all possible positive outcomes neutralized by inhibitors. Unfortunately, the same method on separating all inhibitors from all positives in advance does not work in this model.

So, instead of searching all h-sets in O, we have to search all h-sets in N.

(d, h, e)-GEI ALGORITHM

0 **use** a $(d + h; 2e + 1)$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 **for** all h-subsets $S \subset N$

4 $V \leftarrow V \cup (\cup S)$

5 **ITEM SELECTION**$(N \setminus S, V, D, e)$

Theorem 4.2.2. *The* (d, h, e)**-GEI ALGORITHM** *can identify all positive items under the* (d, h, e)-**GEI** *model.*

Proof. The proof is similar to that of Theorem 4.2.1. Note that a positive item C is identified when S, $C \notin S$, is a set containing all inhibitors. ∎

4.3 Group Testing on Complexes

The pooling design we have discussed so far has a set of positive items each inducing a positive effect. In some applications the property of being positive or negative is defined on subsets of items, instead of on individual items. Such a model is usually referred to as the complex model, first introduced by Torney [51].

In the complex model, we consider a set N of n items and an unknown family $P = \{P_i\}$ of subsets of N where the joint appearance of all items in such a subset causes a certain given biological phenomenon defined as a positive outcome. A set of items which is a candidate of a member of P is called a complex while members of P are called positive complexes. The problem is to identify P from a given set of

complexes through a few experiments. An experiment can be applied to an arbitrary subset $S \subseteq N$ with two possible outcomes; a negative outcome implies S does not contain any $P_i \in P$, and a positive outcome implies otherwise.

Of particular note in the complex model is the basic assumption that no two complexes X and X' satisfy $X' \subseteq X$. The reason is as follows. Observe that in case that a complex X contains a positive complex X^+ as a proper subset, then X^+ appears in all pools where X appears. Therefore X can only appear in positive pools no matter it is positive or negative, i.e., X cannot be identified. Since we do not know which complexes are positive, we make the more sweeping assumption of no containment between any pair of complexes to cover all possible cases.

Let H denote the given set of complexes, then H can be viewed as a hypergraph with items as vertices and complexes as edges. Accordingly, the group testing problem on complexes is related to the graph testing problem on searching a hidden subgraph P in a given graph H, which consists of the set of positive edges. Also, a graph testing algorithm can be represented by an incidence matrix $M = [m_{ij}]$ where rows are labelled by pools, columns by vertices, and $m_{ij} = 1$ if and only if vertex j is in pool i.

Recall that $\cap S$ is the characteristic vector of the set of pools in which every item in S appears. Suppose H is a rank-r graph (each edge consists of at most r vertices) and our only knowledge of P is $|P| \leq d$. A binary matrix is said to be $(H_{\overline{r}} : d)$-*disjunct* if for any $d+1$ edges $e_0, e_1, \cdots, e_d$,

$$\cap e_0 \not\subseteq \bigcup_{i=1}^{d} (\cap e_i).$$

It is easy to see that an $(H_{\overline{r}} : d)$-disjunct matrix can identify P since every edge not in P appears in a test not covering any hidden edge, thus the outcome is negative and the edge is identified.

A binary matrix is said to be $(d, r]$-disjunct (different from (d, r)-disjunct), first studied by Mitchell and Piper [44], if for any $d + r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\left| \bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i \right| \geq 1.$$

That means for any $d+r$ columns there exists a row in which each of the r designated columns has an 1-entry and each of the other d columns has a 0-entry. Such a property was further studied in [11, 36, 49, 50], sometimes under the name of generalized cover-free families or generalized superimposed codes with application to the secure key distribution problem.

Chen, Du and Hwang [10] established a connection between the complex model and the secure key distribution problem, and showed the relation that

$$(d, r]\text{-disjunctness} \Rightarrow (H_{\overline{r}} : d)\text{-disjunctness for all } H_{\overline{r}}.$$

Establishing such a connection leads to the consequence that the $(d, r]$-disjunct matrices can be used to solve the complex model problem.

4.3.1 The Error-Tolerant Complex (EC) Model

In the **EC** model, we consider the problem on the complex model with at most e erroneous outcomes. Review that for a subset X of items, $t_0^V(X) \equiv |\cap X \setminus V|$ and $t_1^V(X) \equiv |(\cap X) \cap V|$, i.e., the number of negative (positive) pools in which every item in X appears, respectively.

Stinson and Wei [49] first gave an error-tolerant version of a $(d, r]$-disjunct matrix. A binary matrix is said to be $(d, r; z]$-disjunct if for any $d+r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\left| \bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i \right| \geq z.$$

COMPLEX SELECTION(H, V, D, e)

1 **for** each complex $X \in H$

2 **compute** $t_0^V(X)$

3 **if** $t_0^V(X) \leq e$

4 **then** $D \leftarrow D \cup \{X\}$

5 **return** D

(d, r, e)**-EC ALGORITHM**

0 **use** a $(d, r; 2e + 1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 **COMPLEX SELECTION**(H, V, D, e)

Theorem 4.3.1. *The* (d, r, e)**-EC ALGORITHM** *can identify all positive complexes under the* (d, r, e)-**EC** *model.*

Proof. It is easy to see that even for the worst case that e outcomes are erroneous, a positive complex X^+ appears in at most e negative pools, i.e., $t_0^V(X^+) \leq e$.

Consider a set P of positive complexes and a negative complex X^-. By the $(d, r; 2e + 1]$-disjunctness property, there exist an r-set R containing X^- and a d-set T, $T \cap R = \emptyset$, intersecting each positive complex such that there are at least $2e + 1$ rows each containing R but none of T. The pools corresponding to these rows must test negative since they do not contain any positive complex. Even for the worst case that e outcomes are erroneous, X^- still appears in at least $e + 1$ negative pools, i.e., $t_0^V(X^-) > e$. Therefore, one can separate all positive complexes from negative ones by using this algorithm. ∎

4.3.2 The Error-Tolerant Inhibitor Complex (EIC) Model

In this subsection, we will introduce a synthetic model on complexes with the presence of inhibitors and erroneous outcomes. Chang et al. [9] are the first to consider such an environment allowing the coexistence of inhibitors and complexes. We use the parameters (d, h, r, e) to denote the assumption that among the complexes which are subsets of n molecules, there are at most d positive complexes each consisting of at most r items, and there are at most h inhibitors and at most e erroneous outcomes.

Consider the simplest inhibitor model ((d, h, r, e)-**EIC** model). The mere existence of a single inhibitor dictates the outcome to be negative, regardless of the presence of positive complexes. The first decoding algorithm we provide here is similar to that in Section 4.2.1 except replacing items by complexes.

(d, h, r, e)-EIC ALGORITHM I

0 **use** a $(d + h, r; 2e + 1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 $O \leftarrow \emptyset$

4 **for** every complex $X \in H$

5 **compute** $t_1^V(X)$

6 **if** $t_1^V(X) \leq e$

7 **then** $O \leftarrow O \cup \{X\}$

8 **for** all h-subsets $S \subseteq O$

9 $V \leftarrow V \cup \left(\bigcup_{X \in S} (\cap X) \right)$

10 **COMPLEX SELECTION**$(H \setminus O, V, D, e)$

Theorem 4.3.2. *The* (d, h, r, e)-**EIC ALGORITHM I** *can identify all positive complexes under the* (d, h, r, e)-**EIC** *model.*

Proof. Similar to the proof of Theorem 4.2.1 except replacing items by complexes. ∎

Notice that O is a set of complexes, thus $|O|$ can be much larger than n. Therefore, the **for loop** in the line 8 requires to go through $\binom{|O|}{h}$ times, which could be a very large number. We now provide an alternative algorithm that only needs to go through $\binom{n}{h}$ times in the worst case.

(d, h, r, e)-**EIC ALGORITHM II**

0 **use** a $(d + h, r; 2e + 1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 **for** all h-subsets $S \subseteq N$

4 $V \leftarrow V \cup (\cup S)$

5 **COMPLEX SELECTION**$(H \setminus S, V, D, e)$

Theorem 4.3.3. *The* (d, h, r, e)-**EIC ALGORITHM II** *can identify all positive complexes under the* (d, h, r, e)-**EIC** *model.*

Proof. The proof is similar to that of Theorem 4.3.2 except that the restoring operation runs through all h-subsets $S \subseteq N$. ∎

4.3.3 The General Error-Tolerant Inhibitor Complex (GEIC) Model

Consider the (d, h, r, e)-**GEIC** model which only assumes the existence of some kind of cancelling effect between the inhibitors and the positive complexes, but no further quantifiable information.

Theorem 4.3.4. *The* (d, h, r, e)**-EIC ALGORITHM II** *identifies all positive complexes under the* (d, h, r, e)-**GEIC** *model as well.*

Proof. Note that the proof of Theorem 4.3.2 does not depend on quantifiable information about the cancelling effect. With a slight modification of the proof of Theorem 4.3.2, we can conclude that the (d, h, r, e)**-EIC ALGORITHM II** also identifies all positive complexes under the (d, h, r, e)-**GEIC** model. ∎

Chapter 5

Group Testing on Complexes

For convenience of investigation, we start with a review. In the complex model, we consider a set N of n items and an unknown family $P = \{P_i\}$ of subsets of N where each such subset is a cause of a certain given biological phenomenon. A set of items which is a candidate of a member of P is called a complex while members of P are called positive complexes. The problem is to identify P from a given set of complexes. An experiment can be applied to an arbitrary subset $S \subseteq N$ with two possible outcomes; a negative outcome implies S does not contain any $P_i \in P$, and a positive outcome implies otherwise.

Of particular note in the complex model is the basic assumption that no two complexes X and X' satisfy $X' \subseteq X$. The reason is as follows. Observe that in case that a complex X contains a positive complex X^+ as a proper subset, X^+ appears in those pools where X also appears. Therefore X can only appear in positive pools no matter it is positive or negative, i.e., X cannot be identified. Since we do not know which complexes are positive, we make the more sweeping assumption of no containment between any pair of complexes to cover all possible cases.

The classic group testing problem has been extended to graph testing (see Chapter 10 of [17] for reference) where a hypergraph $H(V, E)$ is given. The problem is to identify a hidden subgraph P with a small number of graph tests. A graph test can be

applied to an arbitrary subset $S \subseteq V$ with two possible outcomes; a negative outcome implies that all edges in the subgraph H_S induced by S are not in P, while a positive outcome implies otherwise, i.e., H_S contains at least one edge in P, not knowing which or how many. We could have different graph testing problems according as prior knowledge of P; the usual assumption is P has at most d edges, but it can also be P is a matching [1, 4] or a hamiltonian circuit [30]. It is easily seen that the classic group testing problem is a special case of the graph testing problem where H is a 1-graph, i.e., each edge is a vertex.

Let H denote the given set of complexes, then H can be viewed as a hypergraph with items as vertices and complexes as edges. Accordingly, the group testing problem on complexes is related to the graph testing problem on searching a hidden subgraph P in a given graph H.

Establishing such a relation leads to two consequences. The obvious one is all results on graph testing are now available to solve the complex model problem. The less obvious one is a change of emphasis in graph testing research due to the influence of the complex model application. An experiment in the complex model can be time-consuming. Hence it is much preferable to have a nonadaptive algorithm where all subsets for testing are specified at once (and hence can be tested at once theoretically), or at least by a k-round algorithm for some small k. The literature on nonadaptive or k-round algorithms can be found in [1, 30, 31, 19, 38].

In the secure key distribution problem, n persons want to communicate securely in groups of r persons. For this purpose, one takes a number of keys which are distributed within the n persons. When a group of r persons decide to communicate with each other, they take all the keys which are owned by all of them to generate a common key for the whole group. The security requirement is no other d persons

can generate the same key by using the union of all keys they owned.

D'yachkov, Villenkin, Macula and Torney [20] proposed the binary superimposed (d, r)-code which satisfies the property that for any $d+r$ codewords $C_1, C_2, \cdots, C_{d+r}$, there exists an alphabet which is in every code $C_1, C_2, \cdots, C_r$, but none of C_{r+1}, C_{r+2}, $\cdots, C_{d+r}$. This code was further studied in [49, 50, 24, 36].

By treating each key as an alphabet, and the set of keys owned by a person as a codeword, a secure key distribution design is a binary superimposed (d, r)-code.

So far, the connections are quite obvious between the graph testing and the group testing on complexes, and between the superimposed code and the key distribution problem. However, the connection between the former pair of problems and the later pair of problems is not obvious. In fact, even for an inter-pair problem, where the connection is easy, the literature on the two problems are mostly independent. In the following section we will prove an equivalence relation between the two pairs of problems under certain conditions.

5.1 The Equivalence

We first adopt the notation of the graph testing model. Suppose P is the set of hidden edges. Then the outcome set (the indices of rows of positive outcomes) is simply $\bigcup_{e_i \in P} (\cap e_i)$.

Recall that a hypergraph is said to be a rank-r graph if each edge contains at most r vertices, and a hypergraph is an r-graph if each edge contains exactly r vertices.

Suppose H is an r-graph and our only knowledge of P is $|P| \leq d$. We define three properties of M relating to its ability to solve this graph testing problem:

$(H_r : d)$-*separability.* For any two distinct d-sets D, D' of edges,

$$\bigcup_{e_i \in D} (\cap e_i) \neq \bigcup_{e_i \in D'} (\cap e_i). \tag{5.1}$$

$(H_r : \overline{d})$-*separability.* For any two distinct sets D, D' of edges with $|D|,\ |D'| \leq d$,

$$\bigcup_{e_i \in D} (\cap e_i) \neq \bigcup_{e_i \in D'} (\cap e_i). \tag{5.2}$$

$(H_r : d)$-*disjunctness.* For any $d+1$ edges $e_0, e_1, \cdots, e_d$,

$$\cap e_0 \not\subseteq \bigcup_{i=1}^{d} (\cap e_i). \tag{5.3}$$

Clearly, an $(H_r : d)$-separable matrix identifies P if $|P| = d$ is known. An $(H_r : \overline{d})$-separable matrix and an $(H_r : d)$-disjunct matrix can be used to identify P if $|P| \leq d$ is known, while the latter has an easy decoding algorithm since every edge not in P appears in a test not covering any hidden edge, thus the outcome is negative and the edge is identified. Note that when all edges not in P are so identified, the remaining edges are the hidden edges. Thus, $(H_r : d)$-disjunctness implies $(H_r : \overline{d})$-separability implies $(H_r : d)$-separability.

When H is a rank-r graph, we assume that no two edges e and e' satisfy $e \subseteq e'$. Similarly we can define $(H_{\overline{r}} : d)$-*separable*, $(H_{\overline{r}} : \overline{d})$-*separable* and $(H_{\overline{r}} : d)$-*disjunct* matrices respectively, where edges in the definitions are disjoint because of the assumption. When H is the complete r-graph or a complete rank-r graph, then the subscript H will be changed to K. Note that a complete rank-r graph is a maximal graph satisfying the condition that no edge is contained in another edge.

On the other hand, the incidence matrix of a binary superimposed (d, r)-code has the property which we denote by $(d, r]$-disjunctness (different from (d, r)-disjunctness). Namely, for any $d + r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\bigcap_{i=1}^{r} C_i \not\subseteq \bigcup_{i=r+1}^{d+r} C_i. \tag{5.4}$$

Note that condition (5.4) looks different from any one of (5.1), (5.2), (5.3). The only result in the literature making a connection between the two types of results is the following (given in [20]):

Lemma 5.1.1. *$(d, r]$-disjunctness $\Rightarrow$ $(K_{\overline{r}} : \overline{d})$-separability $\Rightarrow$ $(d-1, r]$-disjunctness and $(d, (r-1)]$-disjunctness.*

We will give a proof of the first implication since we believe that the original proof has a slip.

Suppose M is not $(K_{\overline{r}} : \overline{d})$-separable, i.e., there exist two sets of edges D and D' with $|D| \leq d$ and $|D'| \leq d$ such that $\bigcup_{e_i \in D} (\cap e_i) = \bigcup_{e_i \in D'} (\cap e_i)$. By our assumption, neither D nor D' contains two edges one containing the other. Thus there must exist an edge e in $D \cup D'$ such that e does not contain any edge from the other set. For otherwise, we would have $e'' \subseteq e' \subseteq e$ where e and e'' are in the same set. Without loss of generality, assume $e \in D$. Since $e \not\supseteq e_i$ for every $e_i \in D'$, we can choose $C_i \in e_i \setminus e$. Define $S = \{C_i : 1 \leq i \leq |D'|\}$. Then S is a set of at most d columns disjoint from e.

Suppose to the contrary that M is $(d, r]$-disjunct. Then there exists a row with 1-entries in every column of e and 0-entries in every column of S. Thus this row covers e but none of $e_i \in D'$. Hence $\bigcup_{e_i \in D} (\cap e_i) \neq \bigcup_{e_i \in D'} (\cap e_i)$, a contradiction to our previous assumption.

The slip was made by choosing $e \in D \cup D'$ which is not contained in any edge of the other set. Then $e_i \setminus e$ can be empty and C_i cannot be chosen.

We now prove the crucial relation between the two types of results.

Theorem 5.1.2. *$(d, r]$-disjunctness $\Leftrightarrow$ $(K_r : d)$-disjunctness.*

Proof. Suppose M is not $(d, r]$-disjunct. Then there exist $d+r$ columns $C_1, C_2, \cdots,$

C_{d+r} such that

$$\bigcap_{i=1}^{r} C_i \subseteq \bigcup_{i=r+1}^{d+r} C_i.$$

Let $e_0 = \{C_1, C_2, \cdots, C_r\}$ and $e_i = \{C_2, C_3, \cdots, C_r, C_{r+i}\}, 1 \leq i \leq d$. Then

$$\begin{aligned} \cap e_0 &= \cap\{C_2, C_3, \cdots, C_r\} \cap C_1 \\ &\subseteq \cap\{C_2, C_3, \cdots, C_r\} \cap \left(\bigcup_{i=r+1}^{d+r} C_i \right) \\ &= \bigcup_{i=1}^{d} (\cap e_i). \end{aligned}$$

Hence M is not $(K_r : d)$-disjunct.

Conversely, suppose M is $(d, r]$-disjunct. Let $e, e_1, \cdots, e_d$ denote $d+1$ arbitrary edges where no e_i, $1 \leq i \leq d$, is contained in e. Set $C_{r+i} \in e_i \setminus e, 1 \leq i \leq d$, where $e_i \in D$. Then M contains a row which covers e, but intersects none of $C_{r+i}, 1 \leq i \leq d$, i.e., covers none of $e_i \in D$. Hence M is $(K_r : d)$-disjunct. ∎

Corollary 5.1.3. *$(d, r]$-disjunctness $\Rightarrow$ $(H_{\overline{r}} : d)$-disjunctness for all $H_{\overline{r}}$.*

Proof. Note that the converse proof in Theorem 5.1.2 does not depend on the ranks of the edges. ∎

The implication offers us an idea to relax the prior knowledge concerning the given set H of complexes. The problem is how to identify all positive complexes when the structure of H is unknown. Now, we want to prove that a $(d, r]$-disjunct matrix can be used to identify all positives even if the structure of the set of complexes is not specified.

Theorem 5.1.4. *A $(d, r]$-disjunct matrix can identify the up-to-d positives without knowing H.*

Proof. Consider a set $P = \{P_1, P_2, \cdots\}$ consisting of at most d positive complexes and an r'-subset R of columns which contains no P_i, $r' \leq r$. By the $(d, r]$-disjunctness property, there exists a row containing R but none of P_i's. Accordingly, the pool corresponding to this row has a negative outcome, i.e., $t_0^V(R) \geq 1$. After eliminating those subsets, the remaining subsets are those either being a positive complex or containing a positive complex as a proper subset. Thus, we can identify a remaining subset as a positive complex when it contains none of others. Therefore, all positive complexes are identified. ∎

Here we propose an algorithm based on the proof of Theorem 5.1.4.

(d, r)-GENERAL COMPLEX ALGORITHM

0 **use** a $(d, r]$-disjunct matrix M

1 $V \leftarrow$ the outcome vector

2 $D \leftarrow \emptyset$

3 $T_r \leftarrow$ the set of all r'-subsets, $r' \leq r$, in N

4 **while** $T_r \neq \emptyset$

5 **choose** a subset $X \in T_r$ and **compute** $t_0^V(X)$

6 **if** $t_0^V(X) = 0$ and $X \not\supset X'$ for all $X' \in D$

7 **remove** all supersets of X from T_r and D

8 $D \leftarrow D \cup \{X\}$

9 **else** $T_r \leftarrow T_r - X$

10 **return** D

5.2 Two Constructions

Recently, the $(d, r; z]$-disjunct matrices have been found useful in other models such as inhibitor model and threshold model. Details will be explained in Chapter 6 and Chapter 7. In this section, we will propose two methods to construct the $(d, r; z]$-disjunct matrices. The first one is proposed by Chen, Du and Hwang [10] by modifying a construction by Du, Hwang, Wu, Liu and Znati [19]. The latter is proposed by Chen, Fu and Hwang [11].

5.2.1 Converted from q-Ary Matrices

Du et al. [19] gave a construction of the $(H_{\overline{r}} : d)$-disjunct matrices by first constructing a q-ary matrix Q and then converting it to a binary matrix M. Let $f_j(e)$ denote the set of q-ary entries in row j collected from the columns associated with the edge $e \in E$. Then Q has the property that for any $d+1$ edges $e_0, e_1, \cdots, e_d$, there exists a row j in which none of the $f_j(e_i), 1 \leq i \leq d$, is contained in $f_j(e_0)$. For row j in Q, let $c_j = |\{f_j(e) : e \in E\}|$. Then $c_j \leq \min\left\{|E|, \sum_{i=1}^{r} \binom{q}{i}\right\}$. Their conversion is to replace row j in Q by c_j rows, each of which labeled by the set $\{(j, f)\}$ where f is a distinct element in the set $\{f_j(e) : e \in E\}$. For row $\{(j, f)\}$ in the converted matrix M, every column in e with $f_j(e) = f$ (there can be more than one such edge e) has a 1-entry and all other columns have a 0-entry. They proved that such a matrix M converted from a q-ary matrix Q is $(H_{\overline{r}} : d)$-disjunct. They also gave a construction of $Q = [q_{ij}]$ with $drm + 1$ rows and q^{m+1} columns each representing a degree-m polynomial $p_v(x)$ in $GF(q)$, where $v \in V$ and q is a prime power $\geq drm+1$, and the value in the cell q_{ij} is defined by $p_j(i)$. Assuming $|E| \geq \sum_{i=1}^{r} \binom{q}{i}$, the number of tests in the converted matrix M is

$$\sum_{j=1}^{drm+1} c_j \leq (drm + 1) \cdot \sum_{i=1}^{r} \binom{q}{i} \leq (drm + 1) \cdot \binom{q + r - 1}{r}.$$

Chen, Du and Hwang [10] proposed a better conversion. Let $c'_x = |\{p_v(x) : v \in V\}|$ for each row x in Q, then $c'_x \leq q$. For row x in Q, our conversion is to replace each element in the set $\{p_v(x) : v \in V\}$ by a distinct column of a $t' \times c'_x$ $(d, r]$-disjunct matrix. Suppose x is the row in which none of the $f_x(e_i)$ is contained in $f_x(e_0)$. Let $C_i \in e_i \setminus e_0$ such that $f_x(C_i) \notin f_x(e_0)$ for $1 \leq i \leq d$. Then after the conversion there exists at least a row x_j in M, converted from the row x in Q, in which all columns in e_0 have 1-entries while each C_i has 0-entries, $1 \leq i \leq d$. Hence $\cap e_0 \not\subseteq \bigcup_{i=1}^{d} (\cap e_i)$. Since the choice of $e_1, e_2, \cdots, e_d$ is arbitrary, the converted matrix M is $(H_{\overline{r}} : d)$-disjunct.

Let $t(n, d : H_{\overline{r}})$ denote the minimum number of rows required for a $(H_{\overline{r}} : d)$-disjunct matrix with n columns. Similarly, we define $t(n, d, r]$ as the minimum number of rows required for a $(d, r]$-disjunct matrix with n columns. Existing results on $t(q, d, r]$ (see [50] for an example) show that it is less than $\binom{q+r-1}{r}$ in general or at least asymptotically. Thus, we have

Theorem 5.2.1. $t(q^{m+1}, d : H_{\overline{r}}) \leq (drm + 1) \cdot t(q, d, r]$.

When H is the complete r-graph, M is $(K_r : d)$-disjunct. By Theorem 5.1.2, M is also $(d, r]$-disjunct. Then we have

Corollary 5.2.2. $t(q^{m+1}, d, r] \leq (drm + 1) \cdot t(q, d, r]$.

Corollary 5.2.2 is the same result [20] as given by D'yachkov, Vilenkin, Macula and Torney on the construction of $(d, r]$-disjunct matrices using the MDS-code. The incidence matrix of the MDS-code with parameters (q, k, t) is a q-ary matrix of size $t \times q^k$ and the Hamming distance of any pair of columns is $d = t - k + 1$. Lemma 5.2.3 arises from the definition of the MDS-code.

Lemma 5.2.3. *(Sagalovich [46]). If $q^k \geq d + r$ and $t \geq dr(k-1) + 1$, then any MDS-code with parameters (q, k, t) has the property that for any $d + r$ columns*

$C_1, C_2, \cdots, C_{d+r}$, there exists a row where the set of entries over $C_1, C_2, \cdots, C_r$ and the set of entries over $C_{r+1}, \cdots, C_{d+r}$ are disjoint.

D'yachkov et al. used the Reed-Solomon q-ary code, which is also an MDS-code, to get a $(drm+1) \times q^{m+1}$ q-ary matrix with the property that described in Lemma 5.2.3. Then they also use a $t' \times q$ $(d, r]$-disjunct matrix to transform the q-ary matrix to binary one. The requirement of this q-ary matrix is seemingly different from that given by Du et al., though the latter also corresponds to an MDS-code. Chen, Du and Hwang [10] proved that the requirements of the two q-ary matrices are equivalent.

Let $e_0, e_1, \cdots, e_d$ be any $d+1$ complexes. Set $\{C_1, C_2, \cdots, C_r\} = e_0$ and $C_{r+i} \in e_i \setminus e_0,\ 1 \le i \le d$. If the Reed-Solomon q-ary code property holds, i.e., there exists a row x such that the set of entries over $C_1, C_2, \cdots, C_r$ and the set of entries over $C_{r+1}, \cdots, C_{d+r}$ are disjoint, then in the row x none of $f_x(e_i)$ is contained in $f_x(e_0)$, $1 \le i \le d$.

Conversely, let $C_1, C_2, \cdots, C_{d+r}$ be any $d+r$ columns. Set $e_0 = \{C_1, C_2, \cdots, C_r\}$ and $e_i = \{C_2, \cdots, C_r, C_{r+i}\},\ 1 \le i \le d$. If there exists a row x in which none of $f_x(e_i)$ is contained in $f_x(e_0)$, then in the row x the set of entries over $C_1, C_2, \cdots, C_r$ and the set of entries over $C_{r+1}, \cdots, C_{d+r}$ are disjoint.

An Extension to Error-Tolerant Version

Stinson and Wei [49] first gave an error-tolerant version of the $(d, r]$-disjunct matrices. Recall that a matrix is $(d, r; z]$-disjunct if for any $d+r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\left| \bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i \right| \ge z,$$

i.e., there exist at least z rows in which each of the r designated columns has a 1-entry and each of the other d columns has a 0-entry. For a $(d, r; 2e+1]$-disjunct matrix, if the number of errors is less than e, then we can identify the up-to-d positive complexes

consisting of at most r items. It is because that each negative complex appears in at least $(2e+1)-e = e+1$ negative pools, while each positive complex in at most e negative pools (due to errors). Therefore we can separate the negative complexes from the positive ones.

Still, using a $(d, r; 2e+1]$-disjunct matrix, we can identify all positive complexes as well even if the composition of given complexes is not specified. The decoding algorithm is similar to **GENERAL COMPLEX ALGORITHM** except replacing $t_0^V(X) = 0$ by $t_0^V(X) \leq e$.

(d, r, e)-GENERAL COMPLEX ALGORITHM

0 **use** a $(d, r; 2e+1]$-disjunct matrix M
1 $V \leftarrow$ the outcome vector
2 $D \leftarrow \emptyset$
3 $T_r \leftarrow$ the set of all r'-subsets, $r' \leq r$, in N
4 **while** $T_r \neq \emptyset$
5 **choose** a subset $X \in T_r$ and **compute** $t_0^V(X)$
6 **if** $t_0^V(X) \leq e$ and $X \not\supset X'$ for all $X' \in D$
7 **remove** all supersets of X from T_r and D
8 $D \leftarrow D \cup \{X\}$
9 **else** $T_r \leftarrow T_r - X$
10 **return** D

In general, it is not easy to construct a matrix with error-tolerance. A trivial, but not efficient, construction to obtain error-tolerance is by taking copies of each row of the original matrix. Chen, Du and Hwang [10] extended the substitution-type construction mentioned above to the error-tolerant version. Let Q_z be constructed

similar to Q except there are $drm + z$ rows for $z \geq 1$. Surprisingly, by replacing Q with Q_z and a $(d, r]$-disjunct matrix with a $(d, r; z']$-disjunct matrix respectively in the substitution-type construction, we obtain a $(d, r; zz']$-disjunct matrix, which can correct up to $\left\lfloor \frac{(zz'-1)}{2} \right\rfloor$ errors.

Lemma 5.2.4. *For any $d+1$ edges $e_0, e_1, \cdots, e_d$, there exists a set R of at least z rows in Q_z such that for each $j \in R$ none of $f_j(e_i)$ is contained in $f_j(e_0)$, where $1 \leq i \leq d$.*

Proof. By the construction of Q_z, each column is represented by a degree-m polynomial in $GF(q)$, and all of which are distinct. Hence any two columns have common entries in at most m rows.

Suppose to the contrary that there exist at most $z-1$ rows satisfying the condition. Then by the pigeonhole principle there exists an edge $e_x \in \{e_1, e_2, \cdots, e_d\}$ such that there exists a set N of at least $rm+1$ rows satisfying $f_j(e_x) \subseteq f_j(e_0)$ for each $j \in N$. Using the pigeonhole principle again, there exist two columns, one in e_x and the other in $e_0 \setminus e_x$, with common entries in at least $m+1$ rows, a contradiction. ∎

By applying the substitution-type construction to Q_z, we obtain

Theorem 5.2.5. *By replacing each q-ary symbol in Q_z with a distinct column of a $t' \times q$ $(d, r; z']$-disjunct matrix, there exists a $(drm + z) \cdot t' \times q^{m+1}$ $(d, r; zz']$-disjunct matrix M.*

Proof. It suffices to prove that M is $(d, r; zz']$-disjunct. Take a pair of disjoint d-set and r-set of columns, we want to show that there exist zz' rows with 1-entries in the designated r columns and 0-entries in the designated d columns.

After transformation, each row satisfying above condition can generate z' rows each of which has a 1-entry in the designated r columns and a 0-entry in the designated

d columns. By Lemma 5.2.4, there exist z rows whose entries in the d columns are all different from the entries in the r columns. Hence there exist zz' rows with a 1-entry in each of the r columns and a 0-entry in each of the d columns. ∎

Denote $t(n, d, r; z]$ as the minimum number of rows required for a $(d, r; z]$-disjunct matrix with n columns.

Corollary 5.2.6. $t(q^{m+1}, d, r; zz'] \leq \min\{(drm+z)\cdot t(q, d, r; z'], (drm+z')\cdot t(q, d, r; z]\}$

Proof. The proof follows from Theorem 5.2.5 immediately. ∎

5.2.2 Translating into a Vertex Cover Problem

Given a finite set X, a hypergraph $H = (X, \mathcal{F})$ is a family $\mathcal{F} = \{E_1, E_2, \cdots, E_m\}$ of subsets of X. The elements of X are called vertices, and the sets E_i's are the edges of the hypergraph H. A hypergraph with $|E_i| = |E_j|$ for all $i \neq j$ is said to be uniform. For $u \in X$, define the degree $d_H(u)$ of u to be the number of edges containing u. A hypergraph H in which all vertices have the same degree is said to be regular, i.e., $d_H(u) = d_H(v)$ for all $u, v \in X$. A z-cover of H is a subset $C \subseteq X$ such that $|C \cap E_i| \geq z$ for all i. Let $\tau_z(H)$ denote the minimum size among all z-covers of H. It is easy to see that

$$\tau_z(H) \leq z\tau_1(H). \tag{5.5}$$

By a greedy strategy, i.e., choosing vertices sequentially in X such that each chosen vertex intersects the maximum number of edges which are not covered yet, a fundamental result by Lovász [39] implies that

$$\tau_1(H) < \frac{|X|}{\min_{E\in\mathcal{F}} |E|}(1 + \ln \Delta), \tag{5.6}$$

where $\Delta = \max_{u\in X} d_H(u)$.

Chen, Fu and Hwang [11] show that $(d, r; z]$-disjunct matrices can be obtained from z-covers of properly defined hypergraphs, and then (5.6) provides a desired upper bound of $t(n, d, r; z]$.

Let X_w be the set of all binary vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ of length n containing w 1's. For any two disjoint subsets D, R of $[n]$ with $|D| = d$ and $|R| = r$, where $[n]$ denotes the set $\{1, 2, \cdots, n\}$, define the set of binary vectors $E_{D,R} = \{\mathbf{x} = (x_1, x_2, \cdots, x_n) \in X_w : x_i = 1 \text{ for } i \in R \text{ and } x_j = 0 \text{ for } j \in D\}$. Then, for $r \leq w \leq n - d$, define the hypergraph $H = (X_w, \mathcal{F})$, where $\mathcal{F} = \{E_{D,R} : |D| = d,\ |R| = r, \text{ and } D \cap R = \emptyset\}$.

It is easy to see that a $(d, r; z]$-disjunct matrix with n columns can be obtained from a z-cover of $H = (X_w, \mathcal{F})$ by treating $x_1, x_2, \cdots, x_n$ as columns and each vertex in the z-cover as a row, i.e., the jth column has a 1-entry in that row if $x_j = 1$ in that vertex. The reason is that for any $d + r$ designated columns induced by two disjoint sets D and R with $|D| = d$ and $|R| = r$, there exist at least z rows intersecting the edge $E_{D,R}$, meaning each of these rows has 1-entries in every column in R and 0-entries in every column in D. Thus, Chen, Fu and Hwang [11] obtained the following theorem.

Theorem 5.2.7. *For any positive integers d, r, w, z and n, with $r \leq w \leq n - d$, there exists a $t \times n$ $(d, r; z]$-disjunct matrix with*

$$t < \frac{z\binom{n}{r}\binom{n-r}{d}}{\binom{w}{r}\binom{n-w}{d}} \left[1 + \ln \binom{w}{r}\binom{n-w}{d}\right].$$

Proof. By the construction of $H = (X_w, \mathcal{F})$, H is uniform and regular; hence $\dfrac{|X|}{\min_{E \in \mathcal{F}} |E|} = \dfrac{|\mathcal{F}|}{\Delta}$. Moreover, we have $|\mathcal{F}| = \binom{n}{r}\binom{n-r}{d}$ and $\Delta = \binom{w}{r}\binom{n-w}{d}$. The theorem follows directly from (5.5) and (5.6). ∎

From Theorem 5.2.7, all we need to do is to minimize the function by properly

choosing w to obtain a better bound. For convenience, denote $k = d+r$. If we ignore the fact that w must be an integer, then $w = nr/k$, satisfying $w/r = (n-w)/d$, seems to be a good choice to maximize $\binom{w}{r}\binom{n-w}{d}$ (somewhat verified by our limited computations).

Theorem 5.2.8. *For any positive integers d, r, z and n with $d + r \leq n$,*

$$t(n, d, r; z] < z(k/r)^r (k/d)^d [1 + k(1 + \ln(n/k + 1))].$$

Proof. For given positive integers d, r, z and n, setting $w = n'r/k$ where $n' \geq n$ is the least integer such that $n'r/k$ is an integer, we have

$$\begin{aligned}\frac{z\binom{n'}{r}\binom{n'-r}{d}}{\binom{w}{r}\binom{n-w}{d}} &\leq \frac{zn' \cdot (n'-1) \cdots (n'-k+1)}{[(n'r/k) \cdots (n'r/k - r + 1)][(n'd/k) \cdots (n'd/k - d + 1)]} \\ &\leq z(k/r)^r (k/d)^d.\end{aligned}$$

Moreover, using the inequality $\binom{a}{b} \leq (ea/b)^b$, where $e \approx 2.7182$ is the base of the natural logarithm, one concludes

$$\binom{n'r/k}{r}\binom{n' - n'r/k}{d} \leq e^k (n'/k)^k.$$

From the above inequalities and Theorem 5.2.7, we have

$$\begin{aligned}t(n', d, r; z] &< \frac{z\binom{n'}{r}\binom{n'-r}{d}}{\binom{w}{r}\binom{n'-w}{d}} \left[1 + \ln \binom{w}{r}\binom{n'-w}{d}\right] \\ &< z(k/r)^r (k/d)^d [1 + k(1 + \ln(n'/k))].\end{aligned}$$

Note that $n' < n + k$ because of the choice of n'. For any given positive integers d, r, z and n, we have

$$\begin{aligned}t(n, d, r; z] &\leq t(n', d, r; z] \\ &< z(k/r)^r (k/d)^d [1 + k(1 + \ln(n'/k))] \\ &= z(k/r)^r (k/d)^d [1 + k(1 + \ln(n/k + 1))].\end{aligned}$$

∎

5.3 A Combinatorial Lower Bound

A complex X is called an *isolated* complex if there exists a row covering only X. As it is not an efficient test, it is customary to assume that there are no isolated complexes in a $(d, r]$-disjunct matrix, i.e., each row has strictly more than r 1's.

Let M be a $(d, r]$-disjunct matrix and let M' be obtained from M by interchanging all the 1's and 0's. D'yachkov, Vilenkin, Macula and Torney [20] proved that M' is $(r, d]$-disjunct. In other words, if M is an optimal $(d, r]$-disjunct matrix, then M' is an optimal $(r, d]$-disjunct matrix. Hence we strengthen the assumption of no isolated complex to that each row has strictly more than r 1's and d 0's in a $(d, r]$-disjunct matrix. This assumption is made throughout the rest of this subsection.

Let $t^*(n, d, r]$ be the minimum number of rows over all $(d, r]$-disjunct matrices of n columns under the assumption mentioned above. Chen, Du and Hwang [10] defined a secondary parameter w_k, the minimum cardinality of $\cap X$ over all k-sets X of columns, and use a lower bound of it to bound $t^*(n, d, r]$.

Theorem 5.3.1. *Let w_k be the minimum cardinality of $\cap X$ over all k-sets X of columns. For a $(d, r]$-disjunct matrix with no isolated complex, we have*

$$w_i - w_{i+1} \geq d \text{ for } i = 1, 2, \cdots, r.$$

Proof. Let M be a $(d, r]$-disjunct matrix. Given $k \leq r$, consider a column C_0 and a set $C = \{C_1, C_2, \cdots, C_k\}$ of k columns such that $\left|\bigcap_{i=1}^{k} C_i\right| = w_k$. Let $C' = C \cup \{C_0\}$. Suppose $\left|\bigcap_{i=0}^{k} C_i\right| = w$.

Since $k \leq r$, we can choose a set S consisting of other $r - k$ columns. Let $S_1 = S \cup C$ and $S_2 = S \cup C' \setminus \{C_1\}$. Then $|S_1| = |S_2| = r$. Hence S_1 and S_2 are distinct complexes in M. Since $\cap C' \subseteq \cap C$, whatever in $\cap C'$ but not in $\cap(S \cup C')$ is

not in $\cap(S \cup C)$ neither. So, $|\cap S_1 \setminus \cap S_2| \leq |\cap S_1 \setminus \cap(S \cup C')| = w_k - w \leq w_k - w_{k+1}$. If $w_k - w \leq d-1$, then there exist $d-1$ other complexes $X_1, X_2, \cdots, X_{d-1}$ such that $\cap S_1 \subseteq \left(\bigcup_{i=1}^{d-1}(\cap X_i)\right) \cup (\cap S_2)$ (since M has no isolated complexes), violating the $(K_r : d)$-disjunctness property which is equivalent to $(d, r]$-disjunctness property. ∎

Corollary 5.3.2. *For a $(d, r]$-disjunct matrix with no isolated complex, we have*

$$w_i \geq d(r - i + 1) + 1 \text{ for } i = 1, 2, \cdots, r.$$

Proof. By Theorem 5.3.1, $w_i - w_r = (w_i - w_{i+1}) + (w_{i+1} - w_{i+2}) + \cdots + (w_{r-1} - w_r) \geq d(r-i)$, for $i \leq r$. Since each complex is not an isolated complex, $w_r \geq d+1$. Thus, $w_i \geq d(r-i+1)+1$. ∎

Note that w_1 is the minimum weight over all columns.

Corollary 5.3.3. *A $(d, r]$-disjunct matrix with no isolated complex has column weight at least $dr + 1$.*

Without loss of generality, assume $d \geq r$. Then, we obtain

Theorem 5.3.4. $t^*(n, d, r] \geq \frac{r}{2}(d + r - 1)(d - r + 2) + \frac{r}{6}(r-1)(4r-5) + d + r.$

Proof. To prove the theorem, we delete one column and all the intersecting rows from M step by step. Let M be a $(d, r]$-disjunct matrix and let M^1 be obtained from M by deleting a column and all the intersecting rows. By Lemma 2.1.6, M^1 is a $(d-1, r]$-disjunct matrix.

Continue this process till an $(r-1, r]$-disjunct matrix is obtained. To have a better bound, we transform the current matrix to an $(r, (r-1)]$-disjunct matrix by interchanging all the 1's and 0's in the former one, and then keep this process going till a $(1, 1]$-disjunct matrix is obtained. By Corollary 5.3.3 we can count the number

of rows deleted from M. Then we obtain a lower bound of $t^*(n, d, r]$, that is,

$$(dr+1)+((d-1)r+1)+\cdots+((r-1)r+1)+((r-1)^2+1)+((r-2)(r-1)+1)+$$
$$\cdots+(1^2+1)=\frac{r}{2}(d+r-1)(d-r+2)+\frac{r}{6}(r-1)(4r-5)+d+r. \quad \blacksquare$$

Theorem 5.3.4 gives a lower bound by a combinatorial argument which only depends on parameters d and r. For $r = 1$, this bound is reduced to the famous Bassalygo $\binom{d+2}{2}$ bound (see [17] for reference).

5.4 Remarks

The first section of this chapter discussed the relations among four problems: graph testing, group testing on complexes, superimposed codes and secure key distribution. Chen, Du and Hwang [10] proved a surprising equivalence relation among these four problems. Establishing such a relation leads to the consequence that the $(d, r; z]$-disjunct matrices are available to solve the complex model problem.

In the second section, we are interested in constructing the $(d, r; z]$-disjunct matrices and providing an upper bound of $t(n, d, r; z]$. The idea of the first construction, by Chen, Du and Hwang [10], is that first construct a q-ary matrix and then convert it to a binary one. Such an idea can also be found in a number of relevant papers, for instance, in [19, 20, 49]. The method provides a recursive construction scheme to generate the $(d, r; z]$-disjunct matrices. In case of $z = 1$, for example,

1. $t(16, 2, 2] = t(4^2, 2, 2] \leq t(4, 2, 2] \cdot (4 \cdot 1 + 1) \leq 6 \cdot 5 = 30$;

2. $t(256, 2, 2] = t(16^2, 2, 2] \leq t(16, 2, 2] \cdot (4 \cdot 1 + 1) \leq 30 \cdot 5 = 150$;

3. $t(2^{16}, 2, 2] = t(256^2, 2, 2] \leq t(256, 2, 2] \cdot (4 \cdot 1 + 1) \leq 150 \cdot 5 = 750$.

Stinson and Wei [49] provided two asymptotic upper bounds for $t(n, d, r; z]$ by using two other structures, one bound is $O\left(z\binom{d+r}{r}(dr)^{\log^* n}\log n\right)$, where the function

$\log^*$ is defined recursively by $\log^*(1) = 1$ and $\log^* n = \log^*(\lceil \log n \rceil) + 1$ if $n > 1$, and the other is $O\left(z\binom{d+r}{r}\log n\right)$. However, we believe that there is a flaw in the latter one. They showed that for any positive integers d, r, z and n, there exists a $t \times n$ $(d, r; z]$-disjunct matrix with $t = O\left(z\binom{d+r}{r}\log n\right)$, by using a construction of perfect hash families, which was described in [53]. However, the asymptotic result in [53] cited by Stinson and Wei should not be $O(\log n)$, but $O(C \log n)$, where C depends on d and r actually. In addition, the same flaw can also be found in the construction of $(d, 1; 1]$-disjunct matrices in [53]. Notice that the authors also claimed that there exists another explicit construction of $t \times n$ $(d, 1; 1]$-disjunct matrices with $t = O(d^4 \log n)$.

The second construction, by Chen, Fu and Hwang [11], is obtained by translating the problem into a hypergraph problem. Engel [24] first observed the equivalence between a $(d, r; 1]$-disjunct matrix and a cover of a properly defined hypergraph. Stinson and Wei [49] generalized the equivalence to $(d, r; z]$-disjunct matrices for general z, but used the equivalence only to derive a lower bound

$$t(n, d, r; z] \geq 0.7c\frac{(d+r)\binom{d+r}{r}}{\log\binom{d+r}{r}}\log n + \frac{c(z-1)}{2}\binom{d+r}{r}$$

when n is sufficient large, where c is a constant.

The hypergraph we construct is similar to that of Stinson and Wei except that we take a weight-l binary vector as a vertex if and only if $l = w$, while Stinson and Wei relaxed the condition $l = w$ to $r \leq l \leq n - d$. The fixed weight leads to fixed degree in the hypergraph, which allows us to use the Lovász lemma (5.6) on minimum cover to derive an upper bound

$$t(n, d, r; z] < z(k/r)^r(k/d)^d[1 + k(1 + \ln(n/k + 1))],$$

where $k = d + r$. Our result provides a nontrivial and non-asymptotic bound of

$t(n, d, r; z]$. Note that the two upper bounds proposed by Stinson and Wei [49] are asymptotic.

Chapter 6

Group Testing with Inhibitors

In some applications, an item can be positive, negative or anti-positive in the sense that the presence of anti-positives cancels the effect of positives. They are called inhibitors in the literature.

In the simplest inhibitor model, first proposed by Farach et al. [27], the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive items in the pool. This notion has been extended to the k-inhibitor model [7] which requires the existence of k inhibitors to dictate a negative outcome. In addition, we could make even more complicate assumption that each set of k inhibitors cancel the effect of a set of g positive items. In practice, accurate information of k and g is usually not available. Thus in the general inhibitor model, we only assume the existence of some kind of cancelling effect between the inhibitors and the positive items, but no further quantifiable information.

Consider a set N of n items consisting of at most d positives and at most h inhibitors with the others being negatives. Let P denote the set of all positive items and I the set of all inhibitors. The usual concern in the inhibitor model is to identify the set P. Another interesting problem one can consider is to also identify the inhibitor set I. This chapter will focus on the two aspects of concerns.

6.1 Identify Positives Only

In this section, the problem we consider is to identify the positives. For the problem, we first propose a necessary and sufficient condition, and show that it strengthens the necessary condition by De Bonis and Vaccaro [6]. Then a new algorithm that can substantially reduce the time complexity of decoding is presented. We show that there is an algorithm with decoding complexity $O(tn)$ for the inhibitor model and also for the general inhibitor model, where $t = O((d+h+e)^2 \log n)$ is the number of tests needed. Note that our result in the number of tests required is as good as the best known one [32], but has a great improvement in decoding complexity.

6.1.1 A Necessary and Sufficient Condition

In this subsection, consider the problem on the (d, h)-inhibitor model where erroneous outcomes are not allowed. Recall that a matrix is said to be (d, c)-disjunct if no union of c columns is contained in the union of d other columns. Obviously, $(d, 1)$-disjunctness is equivalent to d-disjunctness, and a (d, i)-disjunct matrix is also a (d, j)-disjunct matrix if $i \leq j$. De Bonis and Vaccaro [6] gave the following result for the nonadaptive case.

Theorem 6.1.1. *The (h, d)-disjunctness is a necessary condition for identifying P on the (d, h)-inhibitor model.*

Recall that $\cup X$ denote the union of a set X of columns. For a set X of columns, define $\Phi_h(X) = \{(\cup X) \setminus (\cup Y)$: Y is a set of at most h columns with $X \cap Y = \emptyset\}$.

Definition 6.1.2. A matrix is said to be $(d \setminus h)$-*separable* if for any two distinct sets X, X' of at most d columns,

$$\Phi_h(X) \cap \Phi_h(X') = \emptyset.$$

Theorem 6.1.3. *The $(d \setminus h)$-separability is a necessary and sufficient condition for identifying P on the (d, h)-inhibitor model.*

Proof. The $(d \setminus h)$-separability implies that each set of up-to-d columns induces a disjoint set of outcomes. By matching the sets of positive tests with the samples of positives, the up-to-d positives can be identified.

On the other hand, suppose there are two distinct sets X, X' of at most d columns, such that

$$\Phi_h(X) \cap \Phi_h(X') \neq \emptyset,$$

i.e., there exist Y and Y' such that $(\cup X) \setminus (\cup Y) = (\cup X') \setminus (\cup Y')$. Consider the situations that X is the positive set and Y the inhibitor set, and that X' is the positive set and Y' the inhibitor set. Clearly, they have the same outcomes. Hence, one cannot distinguish whether X or X' is the positive set. ∎

Theorem 6.1.4. *$(d \setminus h)$-separability $\Rightarrow$ $(d + h - 1)$-disjunctness.*

Proof. Suppose there are $d + h$ columns $C_1, \cdots, C_{d+h}$ such that C_1 is contained in $\bigcup_{i=2}^{d+h} C_i$. Let $X = \{C_i : 1 \leq i \leq d\}$, $X' = X \setminus \{C_1\}$ and $Y = \{C_i : d + 1 \leq i \leq d + h\}$. Then $(\cup X) \setminus (\cup Y) = (\cup X') \setminus (\cup Y')$, a contradiction to the $(d \setminus h)$-separability. ∎

Corollary 6.1.5. *For any nonadaptive algorithms, the $(d + h - 1)$-disjunctness is a necessary condition for identifying P on the (d, h)-inhibitor model.*

By definition, it is easily seen that Corollary 6.1.5 strengthens Theorem 6.1.1.

6.1.2 An Extension to Error-Tolerant Version

This subsection focuses on the (d, h, e)-inhibitor model where at most e erroneous outcomes are allowed. Let X and Y be two binary vectors of the same length, and $d(X, Y)$ denote the Hamming distance between X and Y.

Definition 6.1.6. A matrix is said to be $(d \setminus h; z)$-separable if for any two distinct sets X, X' of at most d columns,

$$d(S, T) \geq z \text{ for all } S \in \Phi_h(X) \text{ and } T \in \Phi_h(X').$$

Theorem 6.1.7. *The $(d \setminus h; 2e+1)$-separability is a necessary and sufficient condition for identifying P on the (d, h, e)-inhibitor model.*

Proof. **Necessity.** Suppose to the contrary. Then there exist two distinct sets X, X' of at most d columns and two sets Y, Y' of at most h columns with $X \cap Y = X' \cap Y' = \emptyset$ such that $((\cup X) \setminus (\cup Y)) \setminus ((\cup X') \setminus (\cup Y')) \leq 2e$. Therefore, we can always exploit at most e errors to change the results of some of the $2e$ rows such that $((\cup X) \setminus (\cup Y)) \setminus E_{10} = ((\cup X') \setminus (\cup Y')) \cup E_{01}$ for some sets E_{10} and E_{01} with $|E_{10}|, |E_{01}| \leq e$. Consider the two situations that X is the positive set, Y the inhibitor set and E_{10} the erroneous pools changing positive to negative, and that X' is the positive set, Y the inhibitor set and E_{01} the erroneous pools changing negative to positive. Then it is easily seen that they have same outcomes. Therefore, we cannot determine whether X or X' is the set of positives.

Sufficiency. The $(d \setminus h; 2e + 1)$-separability implies that each set of up-to-d columns induces a disjoint set of vectors and the Hamming distance of each pair of vectors is at least $2e + 1$. Therefore, every outcome vector V matches to a unique set X of positives satisfying $d(V, T) \leq e$ for some $T \in \Phi_h(X)$. Hence, the up-to-d positives can be identified. ∎

Theorem 6.1.8. *The $(d + h - 1; 2e + 1)$-disjunctness is a necessary condition for identifying P on the (d, h, e)-inhibitor model.*

Proof. Suppose the incidence matrix is not $(d + h - 1; 2e + 1)$-disjunct, then there exist $d + h$ columns $C_1, \cdots, C_{d+h}$ such that $\left| C_1 \setminus \bigcup_{i=2}^{d+h} C_i \right| \leq 2e$. Let $X = \{C_i : 1 \leq$

$i \leq d\}$, $X' = X \setminus \{C_1\}$ and $Y = \{C_i : d+1 \leq i \leq d+h\}$. Then $((\cup X) \setminus (\cup Y)) \setminus ((\cup X') \setminus (\cup Y')) \leq 2e$, violating the $(d \setminus h; 2e+1)$-separability. By Theorem 6.1.7, the proof is complete. ∎

6.1.3 A Faster Algorithm

D'yachkov et al. [21] first gave a nonadaptive algorithm for the inhibitor model without erroneous outcomes by using a $(d+h)$-disjunct matrix. The basic idea is to restore all positive outcomes neutralized by inhibitors. Hwang and Liu [32] gave a more efficient decoding algorithm with error-tolerance that reduces the total decoding complexity down to $O\left(t(n-|T|)\binom{|T|}{h}\right)$, where T is a set containing all inhibitors but no positives and t is the number of tests needed. In this subsection, we propose a new algorithm that can substantially reduces the total decoding complexity down to $O(tn)$ with an extra condition.

Definition 6.1.9. A binary matrix is $(h; y)$-*inclusive* if for any $h+1$ columns $C_0, \cdots, C_h$,

$$\left| C_0 \bigcap \left(\bigcup_{i=1}^{h} C_i \right) \right| \leq y.$$

Theorem 6.1.10. *A matrix which is $(d; z)$-disjunct and also $(h; y)$-inclusive with $z - e > y + e$ is $(d+h; 2e+1)$-disjunct.*

Proof. For any $d+h+1$ columns $C_0, C_1, \cdots, C_{d+h}$, we have

$$\begin{cases} \left| C_0 \bigcap \left(\bigcup_{i=1}^{h} C_i \right) \right| \leq y, \\ \left| C_0 \setminus \bigcup_{i=h+1}^{d+h+1} C_i \right| \geq z. \end{cases}$$

From the above two equations, we conclude that

$$\left| C_0 \setminus \bigcup_{i=1}^{d+h} C_i \right| \geq z - y > 2e.$$

Hence, the theorem follows directly from the definition of $(d+h; 2e+1)$-disjunctness.∎

Corollary 6.1.11. *A $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$ can identify all positives on the (d, h, e)-inhibitor model.*

Proof. The proof follows by Theorem 6.1.10 immediately. ∎

We now propose a faster decoding algorithm by using a $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$.

FIND-P ALGORITHM

```
0    use a (d; z)-disjunct and (h; y)-inclusive matrix with z − e > y + e
1    V ← the outcome vector
2    P ← ∅
3    for each item C ∈ N
4        compute t_0^V(C)
5        if t_0^V(C) ≤ y + e
6            then P ← P ∪ {C}
7    return P
```

Theorem 6.1.12. FIND-P ALGORITHM *can identify all positives on the (d, h, e)-inhibitor model.*

Proof. Consider a positive item C^+ and a set I of at most h inhibitors. By the $(h; y)$-inclusiveness property, there are at most y rows each intersecting C^+ and some of I. Therefore, C^+ appears in at most y negative pools beside erroneous pools. Even for the worst case that e pools are erroneous, C^+ appears in at most $y + e$ negative pools, i.e., $t_0^V(C^+) \leq y + e$.

On the other hand, consider a non-positive item C and a set P of at most d positives. By the $(d; z)$-disjunctness property, there are at least z rows each intersecting C but none of P. The outcomes of the pools corresponding to these rows should be negative except for the occurrence of errors. Therefore, we conclude that $t_0^V(C) \geq z - e$, which implies $t_0^V(C) > y + e$.

From the above discussion, **FIND-P ALGORITHM** can determine, through the function $t_0^V(C)$, whether an item C is positive or not. ∎

Then, we estimate the time complexity required for the decoding algorithm. It is easily seen that the decoding algorithm is to compute all $t_0^V(C)$ for every item C. Moreover, the cost of each single computation takes $O(t)$ time where t is the number of tests needed. Hence, we conclude that the total cost of the decoding complexity is $O(tn)$.

An Extension to the General Inhibitor Model

Consider the (d, h, e)-general inhibitor model where we only assume the existence of some kind of cancelling effect between the inhibitors and the positive items, but no further quantitative information. Surprisingly, **FIND-P ALGORITHM** also works even under such ambiguity.

Theorem 6.1.13. *A $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$ identifies all positives on the (d, h, e)-general inhibitor model.*

Proof. Notice that the proof of Theorem 6.1.12 does not depend on quantifiable information about the cancelling effect. ∎

6.1.4 A Construction

One way to construct such a matrix with the property mentioned in Theorem 6.1.12 is to have each column having weight at least w, and each pair of columns contain at most λ elements in common. Let M be such a matrix. Then any h columns intersect a new column at no more than $h\lambda$ rows, and there are at least $w - d\lambda$ rows that a column does not be covered by a set of d other columns. By requiring $w - d\lambda - e > h\lambda + e$, which implies $w > (d + h)\lambda + 2e$, we have that M is $(d; z)$-disjunct and $(h; y)$-inclusive with $z - e > y + e$ where $z = w - d\lambda$ and $y = h\lambda$. Thus, we have the following result.

Theorem 6.1.14. *Let M be a matrix such that every column has weight at least w, and every pair of two columns intersects at no more than λ rows. If $w > (d+h)\lambda+2e$, then M is a $(d; z)$-disjunct and $(h; y)$-inclusive with $z - e > y + e$ where $z = w - d\lambda$ and $y = h\lambda$.*

It is worth pointing out that Hwang and Sós's construction of disjunct matrices [34] satisfies the above conditions and provides us a way to construct the desired matrix. So we can exploit their construction to analyze and estimate the number of tests needed. Given integers t and k, they construct a $t \times n$ matrix with $w = 4kl$ and $\lambda = 4l - 1$, where $n \geq (2/3)3^{t/16k^2}$ and $l = \lceil t/16k^2 \rceil$. Accordingly, we obtain the following result.

Theorem 6.1.15. *Given integers d, h, e and t, there exists a $t \times n$ $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$ such that $n \geq (2/3)3^{t/16k^2}$, where $k > ((4l - 1)(d + h) + 2e)/4l$.*

By setting $k = d + h + 2e$, we have the following results.

Theorem 6.1.16. *There exists a $t \times n$ $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$ such that $t \leq 16(d + h + 2e)^2 \log(3n/2)/\log 3$.*

Corollary 6.1.17. *There exists a $t \times n$ $(d; z)$-disjunct and $(h; y)$-inclusive matrix with $z - e > y + e$ such that $t = O((d + h + e)^2 \log n)$.*

6.2 Identify All Positives and Inhibitors

The problem we consider in this section is to identify not only the positives, but also the inhibitors. Although a number of studies have been made on group testing with inhibitors, little is known on this problem especially for the nonadaptive case.

In order to identify the inhibitors, we make an additional assumption that among the given n items there exists at least one positive item; for otherwise all outcomes would be negative. Hence, one could not distinguish negative items from inhibitors.

6.2.1 A Necessary and Sufficient Condition

Definition 6.2.1. A matrix is said to be $(d \setminus h; z)^*$-*separable* (different from $(d \setminus h; z)$-separable) if for any two pairs of disjoint sets (X, Y) and (X', Y'), where $|X|, |X'| \leq d$ and $|Y|, |Y'| \leq h$,

$$d((\cup X) \setminus (\cup Y), (\cup X') \setminus (\cup Y')) \geq z.$$

Theorem 6.2.2. *The $(d \setminus h; 2e+1)^*$-separability is a necessary and sufficient condition for identifying P and also I on the (d, h, e)-inhibitor model.*

Proof. Similar to the proof of Theorem 6.1.7. ∎

Theorem 6.2.3. *The $(d + h - 1; 2e + 1)$-disjunctness is a necessary condition for identifying P and also I on the (d, h, e)-inhibitor model.*

Proof. Obviously, a matrix which is $(d \setminus h; 2e+1)^*$-separable is also $(d \setminus h; 2e+1)$-separable, which implies $(d+h-1; 2e+1)$-disjunct. ∎

6.2.2 Explicit Algorithms

Theorem 6.2.2 provides a necessary and sufficient condition for identifying P and also I, but essentially neither existence nor explicit construction is known so far, to our best knowledge. Therefore, we turn to seek for a stronger tool which implies the property mentioned in Theorem 6.2.2. In this subsection we provide explicit algorithms by using a tool which is well studied in the complex model.

Recall that a binary matrix is said to be $(d, r; z]$-disjunct if for any $d+r$ columns $C_1, C_2, \cdots, C_{d+r}$,

$$\left| \bigcap_{i=1}^{r} C_i \setminus \bigcup_{i=r+1}^{d+r} C_i \right| \geq z.$$

Theorem 6.2.4. *An $(h, 2; 2e+1]$-disjunct matrix can identify the up-to-h inhibitors with at most e errors.*

Proof. Consider a positive item C^+ and an h-subset I which contains all inhibitors. By the $(h, 2; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each intersecting C^+ but none of I. The pools corresponding to these rows should be positive except erroneous pools. Even for the worst case that e pools are erroneous, C^+ still appears in at least $e+1$ positive pools, i.e., $t_1^V(C^+) \geq e+1$.

Consider a negative item C^-, a positive item C^+ and an h-subset I which contains all inhibitors. By the $(h, 2; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each intersecting C^- and C^+, but none of I. A similar argument implies that $t_1^V(C^-) \geq e+1$.

On the other hand, even for the worst case that e outcomes are erroneous, $t_1^V(C) \leq e$ for every inhibitor C. Therefore, we can separate all inhibitors from the others. ∎

Theorem 6.2.4 provides a two-stage algorithm to identify all the positives and inhibitors.

TWO-STAGE ALGORITHM

Stage 1: Use an $(h, 2; 2e+1]$-disjunct matrix to identify and eliminate all inhibitors.

Stage 2: Use a $(d; 2e + 1)$-disjunct matrix to identify all positives.

The two-stage algorithm also provides us a one-stage approach to identify P and also I. It is quite nature to consider the construction of a matrix which is $(h, 2; 2e + 1]$-disjunct and also $(d; 2e + 1)$-disjunct after deleting any h columns and all rows intersecting these columns. Then the pooling design corresponding to such a matrix can be used to identify all positives and inhibitors in a similar way to the two-stage algorithm. By definition, it is easily seen that a $(d + h, 2; 2e + 1]$-disjunct matrix is $(h, 2; 2e+1]$-disjunct. Moreover, it preserves the $(d; 2e+1)$-disjunctness property after deleting any h columns and all rows intersecting these columns. For otherwise, there exists a column C and a $(d+h)$-set R of columns such that there are at most $2e$ rows intersecting C but none of R, violating the $(d + h, 2; 2e + 1]$-disjunctness property. Accordingly, we have the following result.

Theorem 6.2.5. *A $(d + h, 2; 2e + 1]$-disjunct matrix can identify all positives and inhibitors under the (d, h, e)-inhibitor model.*

Proof. The $(d + h, 2; 2e + 1]$-disjunctness property, which implies $(h, 2; 2e + 1]$-disjunct, can identify all inhibitors according to Theorem 6.2.4. Eliminate the up-to-h columns which represent the inhibitors and all rows intersecting these columns. Then the resulting matrix remains to be $(d; 2e + 1)$-disjunct due to the $(d + h, 2; 2e + 1]$-disjunctness property. Therefore, one can identify the up-to-d positives. ∎

The corresponding decoding algorithm is presented as follows.

FIND-PI ALGORITHM

0 **use** a $(d+h, 2; 2e+1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $P \leftarrow \emptyset$

3 $I \leftarrow \emptyset$

4 **for** each item $C \in N$

5 **compute** $t_1^V(C)$

6 **if** $t_1^V(C) \leq e$

7 **then** $I \leftarrow I \cup \{C\}$

8 $V \leftarrow V \cup (\cup I)$

9 **for** each item $C \in N \setminus I$

10 **compute** $t_0^V(C)$

11 **if** $t_0^V(C) \leq e$

12 **then** $P \leftarrow P \cup \{C\}$

13 **return** P and I

Next, we estimate the time complexity required for this decoding algorithm. It is easy to see that the total cost of the decoding complexity is also $O(tn)$ where t is the number of tests required.

6.2.3 An Extension to the k-Inhibitor Model

Consider the k-inhibitor model, $k \leq h$, which requires the existence of k inhibitors to dictate a negative outcome.

Theorem 6.2.6. *An $(h-k+1, k+1; 2e+1]$-disjunct matrix can identify the up-to-h inhibitors with at most e errors under the k-inhibitor model.*

Proof. Consider a positive item C, a k-subset Y not a subset of the inhibitor-set and an $(h-k+1)$-subset Z which contains either all inhibitors not in Y or $h-k+1$ inhibitors. By the $(h-k+1, k+1; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each intersecting C and all of Y, but none of Z. The pools corresponding to these rows should be positive except erroneous pools. Even for the worst case that e pools are erroneous, the k-subset Y still appears in at least $e+1$ positive pools, i.e., $t_1^V(Y) \geq e+1$. On the other hand, $t_1^V(X) \leq e$ for every k-subset X consisting of inhibitors.

Let $O = \{C \in X : t_1^V(X) \leq e \text{ for each } k\text{-subset } X\}$. From above discussion, we can conclude that O is the set of all inhibitors. ∎

Similarly, there is a two-stage algorithm for the k-inhibitor model by replacing an $(h, 2; 2e+1]$-disjunct matrix with an $(h-k+1, k+1; 2e+1]$-disjunct matrix in the first stage.

TWO-STAGE k-INHIBITOR ALGORITHM

Stage 1: Use an $(h-k+1, k+1; 2e+1]$-disjunct matrix to eliminate all inhibitors.
Stage 2: Use a $(d; 2e+1)$-disjunct matrix to identify all positives.

A one-stage algorithm can also be obtained by a similar argument.

Theorem 6.2.7. *A $(d+h, k+1; 2e+1]$-disjunct matrix can identify all positives and inhibitors under the k-inhibitor model.*

Proof. It follows by the fact that a $(d+h, k+1; 2e+1]$-disjunct matrix M is

$(h-k+1, k+1; 2e+1]$-disjunct, and the matrix obtained from M by deleting any h columns and all rows intersecting these columns is $(d, k+1; 2e+1]$-disjunct. ∎

FIND-PI k-INHIBITOR ALGORITHM

0 **use** a $(d+h, k+1; 2e+1]$-disjunct matrix
1 $V \leftarrow$ the outcome vector
2 $P \leftarrow \emptyset$
3 $I \leftarrow \emptyset$
4 **for** each k-subset $X \subseteq N$
5 **compute** $t_1^V(X)$
6 **if** $t_1^V(X) \leq e$
7 **then** $I \leftarrow I \cup \{C\}$ for all $C \in X$
8 $V \leftarrow V \cup (\cup I)$
9 **for** each item $C \in N \setminus I$
10 **compute** $t_0^V(C)$
11 **if** $t_0^V(C) \leq e$
12 **then** $P \leftarrow P \cup \{C\}$
13 **return** P and I

This decoding algorithm is similar to **FIND-PI ALGORITHM** except replacing item with k-subset in line 4. An analogous argument shows that the decoding complexity of this algorithm is $O\left(\binom{n}{k} kt\right)$ in the worst case, since each operation of computing $t_1^V(X)$ takes $O(kt)$ time where t is the number of tests needed.

Chapter 7

Threshold Group Testing

In this chapter we discuss a generalization of group testing which is a novel model first proposed by Damaschke [15]. The problem is described as follows. Consider a set N of n items consisting of a number of positive items with the other being negative items. Let l and u be nonnegative integers with $l < u$, called the *lower* and *upper threshold*, respectively. A group test for a subset S of items is positive if S contains at least u positives, and negative if at most l positives are present. If the number of positives in S is between l and u, the outcome can give an arbitrary answer. Denote P the set of positive items. Suppose our only prior knowledge is that $u \leq |P| \leq d$. The goal is still to identify all items in P.

Let $g \equiv u - l - 1$ denote the *gap* between the thresholds. The gap $g = 0$ if and only if a sharp threshold separates the positive and negative outcomes, so that all answers are determined. Clearly, the classic group testing problem is a special case that $g = 0$ with $l = 0, u = 1$.

In [15], Damaschke considered only sequential strategies, i.e., the outcomes of all previous pools can be used to set up the next tests. While Damaschke's coverage was predominantly for the sequential strategies, some of his results also hold for the nonadaptive case. For general case that g is not specified, Damaschke proved that one can identify a set P' with $|P' \setminus P| \leq g$ and $|P \setminus P'| \leq g$ by simply testing all u-subsets.

In addition, he proposed an algorithm with $O\left(\frac{u^{g+1}}{(u-1)!(g+1)!}d^u n^{g+1}\right)$ operations to compute a set P' with the properties mentioned above. It is worth mentioning that he also showed that the number of misclassifications which is bounded by the gap g is in a sense the best possible result. That means one cannot identify P exactly in the general case.

In Section 7.1, we first extend the threshold group testing to the error-tolerant case, and provide an efficient pooling design. Mainly, we prove that a $(d-l, u; 2e+1]$-disjunct matrix can be used to identify a set P' as described above with at most e erroneous outcomes. Note that for the case $e = 0$ our result requires many fewer tests than that of Damaschke. For the case without gap, $g = 0$, we provide an efficient decoding algorithm to identify P with complexity $O\left(ut\binom{n}{u}\right)$, where t is the number of tests needed. Section 7.3 introduces a synthetic model on the threshold group testing with the presence of inhibitors and erroneous outcomes. Here we consider only the case without gap, where one can identify all positives exactly.

7.1 Threshold Group Testing with Error-Tolerance

First, we extend the threshold group testing to the error-tolerant version where at most e errors are allowed.

Theorem 7.1.1. *A $(d-l, u; 2e+1]$-disjunct matrix can identify a set P' with $|P' \setminus P| \leq g$ and $|P \setminus P'| \leq g$.*

Proof. Consider a u-subset X containing more than g items not in P, which implies at most l positives, and a $(d-l)$-subset Y which contains either $d-l$ positives not in X or all positives. By the $(d-l, u; 2e+1]$-disjunctness property, there exist $2e+1$ rows each containing X but intersecting none of Y. The pools corresponding to these rows

should be negative except erroneous pools. Even for the worst case that e pools are erroneous, X still appears in at least $e+1$ negative pools. Therefore, $t_0^V(X) \geq e+1$.

Observe that every u-subset $X^+ \subseteq P$ should appear in positive pools except erroneous pools. Hence, even for the worst case that e pools are erroneous, we have $t_0^V(X^+) \leq e$.

Construct a hypergraph H_u where a u-set X is an edge if and only if X appears in at most e negative pools. Set P' to be a maximum clique of H_u. Note that P must be a clique of H_u. Further, a clique cannot contain more than g vertices not in P, or the u-subset containing any g of these vertices cannot be an edge, contradicting the definition of a clique. From these two observations, the theorem follows immediately.∎

For $e = 0$, Theorem 7.1.1 offers a way to identify a set P' as described but requires fewer tests than that of Damaschke, testing all $\binom{n}{u}$ u-subsets.

7.2 The Case without Gap

For the special case without gap, $g = 0$, Theorem 7.1.1 implies that we can identify the set P exactly. Then, we obtain the following corollary.

Corollary 7.2.1. *For the case $g = 0$, a $(d-u+1, u; 2e+1]$-disjunct matrix can identify the set P.*

Proof. Note that a u-subset X appears in at most e negative pools if $X \subseteq P$, and appears in at least $e+1$ negative pools if $X \not\subseteq P$. ∎

Accordingly, we have the following algorithm geared to identify P.

THRESHOLD ALGORITHM

0 **use** a $(d-u+1, u; 2e+1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $P \leftarrow \emptyset$

3 **for** each u-subset $X \subseteq N$

4 **compute** $t_0^V(X)$

5 **if** $t_0^V(X) \leq e$

6 **then** $P \leftarrow P \cup \{C\}$ for all $C \in X$

7 **return** P

Obviously, the **for loop** is executed $\binom{n}{u}$ times. Further, each operation for computing $t_0^V(X)$ takes $O(ut)$ time in the worst case. Therefore, the total decoding complexity takes $O\left(ut\binom{n}{u}\right)$ time.

7.3 The Inhibitor Threshold Model without Gap

In this section we introduce the threshold group testing in the presence of inhibitors and errors. Here we consider only the case $g = 0$. Denote I as the set of all inhibitors with $|I| \leq h$.

Recall that in the simplest inhibitor problem the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive items in the pool. Further, the k-inhibitor model requires the existence of k inhibitors to dictate a negative outcome. It has been extended to the general model in which the exact cancellation effect of inhibitors on positive items is not specified.

7.3.1 Identify Positives Only

The key point of our idea is to restore all positive outcomes neutralized by inhibitors. The method of this algorithm we proposed here is to collect all inhibitors into the set O which contains no positives, and then identify a column C as positive

if there exists one S and a u-subset X containing C for which $t_0^V(X) \leq e$ under the outcome vector V adjusted by S.

INHIBITOR THRESHOLD ALGORITHM

0 **use** a $(d+h-u+1, u; 2e+1]$-disjunct matrix
1 $V \leftarrow$ the outcome vector
2 $D \leftarrow \emptyset$
3 $O \leftarrow \emptyset$
4 **for** every item $C \in N$
5 **compute** $t_1^V(C)$
6 **if** $t_1^V(C) \leq e$
7 **then** $O \leftarrow O \cup \{C\}$
8 **for** all h-subsets $S \subseteq O$
9 $V \leftarrow V \cup (\cup S)$
10 **for** each u-subset $X \subseteq N \setminus O$
11 **compute** $t_0^V(X)$
12 **if** $t_0^V(X) \leq e$
13 **then** $D \leftarrow D \cup \{C\}$ for all $C \in X$
14 **return** D

The **for loop** in line 8 causes the majority of operations of this algorithm. The operations for computing $t_0^V(X)$ execute $O\left(\binom{|O|}{h}\binom{n-|O|}{u}\right)$ times in the worst case. Therefore, the total time complexity is $O\left(ut\binom{|O|}{h}\binom{n-|O|}{u}\right)$ where t is the number of tests required.

To prove the correctness of this algorithm, what we need to show first is that O contains all inhibitors but no positives.

Lemma 7.3.1. *O contains all inhibitors but no positives.*

Proof. By definition, a $(d+h-u+1, u; 2e+1]$-disjunct matrix is also $(h, 1; 2e+1]$-disjunct. Consider a positive item C^+ and an h-subset X containing all inhibitors. By the $(h, 1; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each intersecting C^+ but none of X. The pools corresponding to these rows should be positive except erroneous pools. Even for the worst case that e pools are erroneous, C^+ still appears in at least $e+1$ positive pools, i.e., $t_1^V(C^+) \geq e+1$. Therefore, we can conclude that an item which appears in at most e positive pools cannot be positive.

On the other hand, even for the worst case that e pools are erroneous, every inhibitor appears in at most e positive pools. Hence the set O contains all inhibitors but no positives. ∎

Theorem 7.3.2. *The* **INHIBITOR THRESHOLD ALGORITHM** *can identify P with at most h inhibitors and at most e errors.*

Proof. From Lemma 7.3.1, we conclude that $P \subseteq N \setminus O$. Consider a u-subset $X \subseteq N \setminus O$ not a subset of P, a $(d-u+1)$-subset $Y \subseteq N \setminus O$ which contains either all positives not in X or $d-u+1$ positives. For each h-subset $S \subseteq O$, by the $(d+h-u+1, u; 2e+1]$-disjunctness property, there exists a $(d+h-u+1)$-set R of columns containing Y and S such that there are at least $2e+1$ rows each containing X but intersecting none of R. The outcomes of the pools corresponding to these rows should be negative except for the occurrence of errors. Therefore, we can conclude that $t_0^V(X) \geq (2e+1) - e = e+1$. Hence no negative item is selected into D.

Consider an h-subset $S \subseteq O$ containing all up-to-h inhibitors with the others being negative items. For a u-subset $X^+ \subseteq P$, X^+ appears only in the pools in the new outcome vector V adjusted by the set S. For the worst case that e outcomes are

erroneous, X^+ appears in at most e negative pools, i.e., $t_0^V(X^+) \leq e$. Hence every positive item will be selected into D.

From the above discussion, the output of this algorithm is the set P. ∎

An Extension to the General Inhibitor Model

Consider the general inhibitor model where we only assume the existence of some kind of cancelling effect between the inhibitors and the positive items, but no further quantifiable information. Even under such ambiguity, a $(d+h-u+1, u; 2e+1]$-disjunct matrix still works as well. Unfortunately, the same method on separating all inhibitors from all positives in advance does not work in this model. So, instead of searching all h-sets in O, we have to search all h-sets in N.

GENERAL INHIBITOR THRESHOLD ALGORITHM

```
0   use a (d+h-u+1, u; 2e+1]-disjunct matrix
1   V ← the outcome vector
2   D ← ∅
3   for all h-subsets S ⊆ N
4       V ← V ∪ (∪S)
5       for each u-subset X ⊆ N \ S
6           compute t_0^V(X)
7           if t_0^V(X) ≤ e
8               then D ← D ∪ {C} for all C ∈ X
```

Theorem 7.3.3. *The* **GENERAL INHIBITOR THRESHOLD ALGORITHM** *can identify P with at most h inhibitors and at most e errors under the general inhibitor model.*

Proof. The proof is similar to the proof of Theorem 7.3.2 except that the restoring operation runs through all h-subsets $S \subseteq N$. ∎

Similarly, we conclude that the total time complexity is $O\left(ut\binom{n}{h}\binom{n-u}{u}\right)$ where t is the number of tests required.

7.3.2 Identify All Positives and Inhibitors

In this subsection, the problem we are concerned is not only to identify all positives, but also to identify all inhibitors. In order to identify the inhibitors, we need to make an additional assumption that among the given n items there exist at least u positives. For otherwise, all outcomes would be negative. Hence, one could not separate the inhibitors from the negatives.

Theorem 7.3.4. *An $(h, u+1; 2e+1]$-disjunct matrix can identify the up-to-h inhibitors with at most e errors.*

Proof. Consider a u-subset X of positive items and an h-subset I which contains all inhibitors. By the $(h, u+1; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each containing C^+ but not intersecting I. The pools corresponding to these rows should be positive except erroneous pools. Even for the worst case that e pools are erroneous, X still appears in at least $e+1$ positive pools, i.e., $t_1^V(C^+) \geq e+1$ for every positive item $C^+ \in X$.

Consider a negative item C^-, a u-subset X of positive items and an h-subset I which contains all inhibitors. By the $(h, u+1; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each containing C^- and X but not intersecting I. A similar argument shows that $t_1^V(C^-) \geq e+1$.

On the other hand, even for the worst case that e outcomes are erroneous, $t_1^V(C) \leq e$ for every inhibitor C. Therefore, all the inhibitors can be identified. ∎

Theorem 7.3.4 provides a two-stage algorithm to identify all positives and also inhibitors.

TWO-STAGE ALGORITHM

Stage 1: Use an $(h, u+1; 2e+1]$-disjunct matrix to identify and eliminate all inhibitors.

Stage 2: Use a $(d-u+1, u; 2e+1]$-disjunct matrix to identify all positives.

The two-stage algorithm also provides us a one-stage approach to solve the problem. It is quite nature to think of a matrix which is $(h, u+1; 2e+1]$-disjunct and also preserves the $(d-u+1, u; 2e+1]$-disjunctness property after deleting any h columns and all rows intersecting these columns. Then the pooling design corresponding to such a matrix can be used to identify all positives and inhibitors. By definition, it is easily seen that a $(d+h-u+1, u+1; 2e+1]$-disjunct matrix satisfies the property mentioned above.

Theorem 7.3.5. *A $(d+h-u+1, u+1; 2e+1]$-disjunct matrix can identify all positives and inhibitors with at most e errors.*

Proof. The $(d+h-u+1, u+1; 2e+1]$-disjunctness property, which implies the $(h, u+1; 2e+1]$-disjunctness, can be used to identify all inhibitors according to Theorem 7.3.4. Eliminate the up-to-h columns which represent the inhibitors and all rows intersecting these columns. Then, the resulting matrix remains to be $(d-u+1, u; 2e+1]$-disjunct due to the $(d+h-u+1, u+1; 2e+1]$-disjunctness property. Therefore, the positives can be identified by Theorem 7.3.2. ∎

The following decoding algorithm is based on Theorem 7.3.5.

THRESHOLD FIND-PI ALGORITHM

0 **use** a $(d+h-u+1, u+1; 2e+1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $P \leftarrow \emptyset$

3 $I \leftarrow \emptyset$

4 **for** each item $C \in N$

5 **compute** $t_1^V(C)$

6 **if** $t_1^V(C) \leq e$

7 **then** $I \leftarrow I \cup \{C\}$

8 $V \leftarrow V \cup (\cup I)$

9 **for** each u-subset $X \subseteq N \setminus I$

10 **compute** $t_0^V(X)$

11 **if** $t_0^V(X) \leq e$

12 **then** $P \leftarrow P \cup \{C\}$ for all $C \in X$

13 **return** P and I

Our concern now is to estimate the time complexity required for this decoding algorithm. It is easy to see that the operations for computing $t_0^V(X)$ execute $\binom{n}{u}$ times in the worst case. Hence, the total decoding complexity is $O\left(ut\binom{n}{u}\right)$ where t is the number of tests required.

An Extension to the k-Inhibitor Model

Consider the k-inhibitor model which requires the existence of k inhibitors to dictate a negative outcome. Still, we need to assume that there exist at least k inhibitors among the given n items.

Theorem 7.3.6. *An $(h-k+1, u+k; 2e+1]$-disjunct matrix can identify the up-to-h inhibitors with at most e errors under the k-inhibitor model.*

Proof. Consider a u-subset X of positive items, a k-subset Y not a subset of the inhibitor-set I and an $(h-k+1)$-subset Z which contains either all inhibitors not in Y or $h-k+1$ inhibitors. By the $(h-k+1, u+k; 2e+1]$-disjunctness property, there exist at least $2e+1$ rows each containing X and Y but intersecting none of Z. The pools corresponding to these rows should be positive except erroneous pools. Even for the worst case that e pools are erroneous, the k-subset Y still appears in at least $e+1$ positive pools, i.e., $t_1^V(Y) \geq e+1$.

On the other hand, $t_1^V(X) \leq e$ for every k-subset X consisting of inhibitors. Let $O = \{C \in X : t_1^V(X) \leq e \text{ for each } k\text{-subset } X\}$. Then, it is easy to see that O is the set of all inhibitors. ∎

Similarly, there is a two-stage algorithm for the k-inhibitor model by replacing an $(h, u+1; 2e+1]$-disjunct matrix with an $(h-k+1, u+k; 2e+1]$-disjunct matrix in the first stage.

k-INHIBITOR TWO-STAGE ALGORITHM

Stage 1: Use an $(h-k+1, u+k; 2e+1]$-disjunct matrix to identify and eliminate all inhibitors.

Stage 2: Use a $(d-u+1, u; 2e+1]$-disjunct matrix to identify all positives.

A one-stage algorithm can also be obtained by a similar argument.

Theorem 7.3.7. *A $(d+h-u+1, u+k; 2e+1]$-disjunct matrix can identify all positives and inhibitors with at most e errors under the k-inhibitor model.*

Proof. The proof is similar to that of Theorem 7.3.5.

We also propose a decoding algorithm based on Theorem 7.3.7 in the following.

k-INHIBITOR THRESHOLD FIND-PI ALGORITHM

0 **use** a $(d+h-u+1, u+k; 2e+1]$-disjunct matrix

1 $V \leftarrow$ the outcome vector

2 $P \leftarrow \emptyset$

3 $I \leftarrow \emptyset$

4 **for** each k-subset $X \subseteq N$

5 **compute** $t_1^V(X)$

6 **if** $t_1^V(X) \leq e$

7 **then** $I \leftarrow I \cup \{C\}$ for all $C \in X$

8 $V \leftarrow V \cup (\cup I)$

9 **for** each u-subset $X \subseteq N \setminus I$

10 **compute** $t_0^V(X)$

11 **if** $t_0^V(X) \leq e$

12 **then** $P \leftarrow P \cup \{C\}$ for all $C \in X$

13 **return** P and I

Similarly, it is easy to see that the operations for computing $t_1^V(X)$ executed $\binom{n}{k}$ times, and for computing $t_0^V(X)$ executed $\binom{n}{u}$ in the worst case. Hence, the total decoding complexity is $O\left(pt\binom{n}{p}\right)$ where $p = \max\{u, k\}$ and t is the number of tests required.

Bibliography

[1] N. Alon, R. Beigel, S. Kasif, S. Rudich and B. Sudakov, Learning a hidden matching, SIAM J. Comput. 33 (2004) 487-501.

[2] D. J. Balding, W. J. Bruno, E. Knill and D. C. Torney, A comparatively survey of nonadaptive pooling designs, in Genetic Mapping and DNA Sequencing, IMA Vol. in Mathematics and Its Applications, Springer, 1996, pp. 133-154.

[3] D. J. Balding and D. C. Torney, Optimal pooling designs with error detection, J. Combin. Theory Ser. A 74 (1996) 131-140.

[4] R. Beigel, N. Alon, M. S. Apaydin, L. Fortnow and S. Kasif, An optimal procedure for gap closing in whole genome shotgun sequencing, Proc. 2001 RECOMB, ACM Press, pp. 22-30.

[5] A. De Bonis, L. Gasieniec and U. Vaccaro, Optimal two-stage algorithms for group testing problems, SIAM J. Comput. 34 (2005) 1253-1270.

[6] A. De Bonis and U. Vaccaro, Improved algorithms for group testing with inhibitors, Inform. Proc. Lett. 67 (1998) 57-64.

[7] A. De Bonis and U. Vaccaro, Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels, Theor. Comput. Sci. 306 (2003) 223-243.

[8] K. A. Bush, W. T. Federal, H. Pesotan and D. Raghavarao, New combinatorial designs and their applications to group testing, J. Statist. Plan. Infer. 10 (1984) 335-343.

[9] F. H. Chang, H. L. Chang and F. K. Hwang, Pooling designs for clone library screening in the inhibitor complex model, submitted.

[10] H. B. Chen, D. Z. Du and F. K. Hwang, An unexpected meeting of four seemingly unrelated problems: graph testing, DNA complex screening, superimposed codes and secure key distribution, J. Combin. Opt., to appear.

[11] H. B. Chen, H. L. Fu and F. K. Hwang, An upper bound of the number of tests in pooling designs for the error-tolerant complex model, Opt. Letters, to appear.

[12] H. B. Chen and F. K. Hwang, A survey on nonadaptive group testing algorithms through the angle of decoding, submitted.

[13] H. B. Chen and F. K. Hwang, Exploring the missing link among d-separable, $\overline{d}$-separable and d-disjunct matrices, Discrete Appl. Math., to appear.

[14] H. B. Chen, C. M. Li and F. K. Hwang, Bounding the number of columns which appear only in positive pools, Taiwaness J. Math. 10 (2006) 927-932.

[15] P. Damaschke, Threshold group testing, Electron. Notes in Discrete Math. 21 (2005) 265-271.

[16] R. Dorfman, The detection of defective members of large populations, Ann. Math. Statist. 14 (1943) 436-440.

[17] D. Z. Du and F. K. Hwang, Combinatorial Group Testing and Its Applications, $2nd$ ed., World Scientific, Singapore, 2000.

[18] D. Z. Du and F. K. Hwang, Pooling Designs and Nonadaptive Group Testing - Important Tools for DNA Sequencing, World Scientific, 2006.

[19] D. Z. Du, F. K. Hwang, W. Wu, Z. Liu and T. Znati, Construction of $d(H)$-disjunct matrix for group testing in complex model, J. Combin. Opt., preprint, 2005.

[20] A. G. D'yachkov, P. A. Vilenkin, A. J. Macula and D. C. Torney, Families of finite sets in which no intersection of ℓ sets is covered by the union of s others, J. Combin. Theory Ser. A 99 (2002) 195-218.

[21] A. G. D'yachkov, A. J. Macula, D. C. Torney and P. A. Villenkin, Two models of nonadaptive group testing for designing screening experiments, in A. C. Atkinson, P. Hackl and W. G. Muller (eds): Proc. 6th Inter. Workshop in Model Oriented Design and Analysis, Physica-Verlog 2001, pp. 63-75.

[22] A. G. D'yachkov and V. V. Rykov, Bounds of the length of disjunct codes, Problems Control Inform. Theory 11 (1982) 7-13.

[23] A. G. D'yachkov and V. V. Rykov, A survey of superimposed code theory, Problems Control Inform. Theory 12 (1983) 229-242.

[24] K. Engel, Interval packing and covering in the boolean lattice, Combin. Prob. Comp. 5 (1996) 373-384.

[25] P. Erdős and L. Moser, Problem 35. Proc. Conf. Combin. Structures and Appl. Calgary, 1969, Gorden and Breach, New York, 1970, p. 506.

[26] P. Erdős, P. Frankl and Z. Fűredi, Family of finite sets in which no set is coverd by the union of n others, Isr. J. Math. 51 (1985) 79-89.

[27] M. Farach, S. Kannan, E. Knill and S. Muthukrishnan, Group testing problem with sequences in experimental molecular biology, Proc. Compression and Complexity of Sequences, pp. 357-367, 1997.

[28] P. Frankl and Z. Fűredi, Union-free hypergraphs and probability theory, Euro. J. Combin. 5 (1984) 127-131.

[29] Z. Fűredi, On r-cover-free families, J. Combin. Theory Ser. A 75 (1996) 172-173.

[30] V. Grebinski and G. Kucherov, Reconstructing a Hamiltonian cycle by querying the graph: Application to DNA physical mapping, Discrete Appl. Math. 88 (1998) 147-165.

[31] V. Grebinski and G. Kucherov, Optimal reconstruction of graphs under the additive model, Algorithmica 28 (2000) 104-124.

[32] F. K. Hwang and Y. C. Liu, Error-tolerant pooling designs with inhibitors, J. Comp. Biol. 10 (2003) 231-236.

[33] F. K. Hwang, T. T. Song and D. Z. Du, Hypergeometric and generalized hypergeometric group testing, SIAM J. Alg. Discrete Methods 2 (1981) 426-428.

[34] F. K. Hwang and V. T. Sós, Nonadaptive hypergeometric group testing, Studia Scient. Math. Hungarica 22, 1987.

[35] W. H. Kautz and R. R. Singleton, Nonrandom binary superimposed codes, IEEE Trans. Inform. Theory 10 (1964) 363-377.

[36] H. K. Kim and V. Lebedev, On optimal superimposed codes, J. Combin. Designs 12 (2004) 79-91.

[37] C. H. Li, A sequential method for screening experimental variables, J. Amer. Statist. Assoc. 57 (1962) 455-477.

[38] Y. Li, M. Thai, Z. Lin and W. Wu, Protein to protein interactions and group testing in bipartite graphs, Inter. J. Bioinformatics Appl., to appear.

[39] L. Lovász, On the ratio of optimal integral and fractional covers, Discrete Math. 13 (1975) 383-390.

[40] F. J. MacWilliams and N. J. A. Sloane, The theory of error-correcting codes, North-Holland, Amsterdam, The Netherlands, 1983.

[41] A. J. Macula, D. C. Torney and P. A. Vilenkin, Two-stage group testing for complexes in the presence of errors, DIMACS Ser. Discrete Math. and Theor. Comp. Sci. 55 (1999) 145-157.

[42] A. J. Macula, V. V. Rykov and S. Yekhanin, Trivial two-stage group testing for complexes using almost disjunct matrices, Discrete Appl. Math. 137 (2004) 97-107.

[43] A. J. Macula and L. J. Popyack, A group testing method for finding patterns in data, Discrete Appl. Math. 144 (2004) 149-157.

[44] C. J. Mitchell and F. C. Piper, Key storage in secure networks, Discrete Appl. Math. 21 (1988) 215-228.

[45] H. Q. Ngo and D. Z. Du, A survey on combinatorial group testing algorithms with applications to DNA library screening, DIMACS Ser. Discrete Math. and Theor. Comp. Sci. AMS 55 (2000) 171-182.

[46] Yu. L. Sagalovich, On separating systems, Problemy Peredachi Informatsii 30 (1994) 14-35.(in Russian).

[47] M. Sobel and P. A. Groll, Group testing to eliminate efficiently all defectives in a binomial sample, Bell System Tech. J. 28 (1959) 1179-1252.

[48] D. R. Stinson, On some methods for unconditionally secure key distribution and broadcast encryption, Designs, Codes and Cryptography 12 (1997) 215-243.

[49] D. R. Stinson and R. Wei, Generalized cover-free families, Discrete Math. 279 (2004) 463-477.

[50] D. R. Stinson, R. Wei and L. Zhu, Some new bounds for cover-free families, J. Combin. Theory Ser. A 90 (2000) 224-234.

[51] D. C. Torney, Sets pooling designs, Ann. Combin. 3 (1999) 95-101.

[52] W. Wu, C. Li, X. Wu and X. Huang, Decoding in pooling designs, J. Combin. Opt. 7 (2003) 385-388.

[53] H. Wang and C. Xing, Explicit constructions of perfect hash families from algebraic curves over finite fields, J. Combin. Theory Ser. A 93 (2001) 112-124.