

國立交通大學

應用數學系

博士論文

組合群試問題的研究

Combinatorial Group Testing Problems
on Various Models

研究生：張惠蘭

指導教授：傅恆霖 教授



中華民國九十九年六月

Combinatorial Group Testing Problems on Various Models


組合群試問題的研究

研究生: 張惠蘭 Student: Huilan Chang
指導教授: 傅恆霖 教授 Advisor: Hung-Lin Fu

國立交通大學

應用數學系

博士論文



A Dissertation
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Applied Mathematics
June 2010
Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

Abstract

In the classical group testing problem, there is a set \mathcal{N} of n clones, each of which is either positive or negative. The main task of the problem is to identify all positive ones by group tests, and in identifying all positive clones, minimizing the number of group tests is the main issue. Motivated by applications, many studies have introduced a third type of clones called “inhibitors” whose effect is in a sense to obscure the positive clones in pools. Furthermore, in many applications, a subset of clones (rather than a single clone), called a complex, can induce a positive effect.

There are two general types of group testing algorithms: sequential and nonadaptive. A sequential algorithm conducts the tests one by one where the outcomes of all previous tests can be treated as a reference to the later one, while a nonadaptive algorithm specifies all tests in advance and thus all tests can be conducted simultaneously. Generally, sequential algorithms require fewer number of tests than nonadaptive ones, but performing all tests in a sequential algorithm spends more time than performing all tests in a nonadaptive one.

The group testing model which takes inhibitors (respectively complexes) into consideration is referred to as an inhibitor model (respectively a complex model). These two models have been well studied in the group testing literature. In this thesis, we first study group testing problems in a new pooling design environment by allowing the coexistence of inhibitors and complexes and devote our attention to nonadaptive algorithms. To identify positive items, we attach a novel property “inclusiveness” to a design. This property

and a well-studied property “disjunctness” lead to a significant improvement in the decoding procedure. In addition to the identification problem where only positive items are identified, we also attempt to classify all items. We prove that the well-studied “ $(d, r; z]$ -disjunct matrices” are sufficient for the classification problems and associated with fast decoding procedures.

In the identification and classification problems, $(H : d; z)$ -disjunct, $(d, r; z]$ -disjunct, and $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive with $z > y$ are three main properties of matrices that are employed as nonadaptive pooling designs. We study their constructions and the lower bounds on the number of rows (tests).

Finally, we study the graph reconstruction problem which is a generalization of the classical combinatorial group testing problem. A group testing problem is a search paradigm where it is usually assumed that there are at most d positive items among given items. A graph reconstruction problem is to reconstruct a hidden graph G from a given family of graphs by asking queries of the form “Whether a set of vertices induces an edge of G ”. Reconstruction problems on families of Hamiltonian cycles, matchings, stars and cliques on n vertices have been studied where algorithms of using at most $2n \lg n$, $(1 + o(1))(n \lg n)$, $2n$ and $2n$ queries were proposed, respectively. We exploit some strategies such as affine plane method to improve them to $(1 + o(1))(n \lg n)$, $(1 + o(1))(\frac{n}{2} \lg n)$, $n + 2 \lg n$ and $n + \lg n$, respectively.

摘要

所謂傳統的群式問題 (classical group testing problem), 是要從含有正克隆 (clones) 及負克隆的群體中識別出正的克隆。其所使用的工具是群試驗 (group tests), 而如何減少群試驗的使用量是主要被重視的問題。應用當中也經常衍生出其它型態的克隆, 比如抑制型克隆 (inhibitor)。它能攪擾正克隆的特性, 使其不能發揮正常的功用, 因此一個含有正克隆的群試驗可能無法顯現正克隆的存在, 若它同時也含有抑制克隆。此外, 在 NDA 篩選的環境中, 有些特定的克隆能組合出具有相當特性的複合體, 我們稱它為克隆複合體 (complex); 因此, 相對於克隆模型, 複合模型探討的是正複合體的識別。

逐步演算法 (sequential algorithm) 及非調整型演算法 (nonadaptive algorithm) 是兩個普遍的群試演算法。前者當中的試驗是逐一進行的, 且下一個試驗可依據之前進行過的試驗的結果而去設計; 後者當中所含的試驗是同時進行的, 也就是只依據問題給定的訊息及假設去設計所有的試驗, 且使其能達到識別所有正元素的能力。一般而言, 逐步演算法所涉及的試驗量比非調整型演算法的來得少, 但其完成所有試驗所需的時間比非調整型的來得多。

群試抑制模型 (the inhibitor model) 指的是含有抑制型元素的群試模型; 而群試複合模型 (the complex model) 是指所探討的問題是建立在複合體上。在群試研究的文獻中, 這兩個模型已各別有完善的發展。在此論文中, 我們提出抑制元素和複合體共存的群試環境 (the inhibitor complex model), 並專攻於非調整型演算法的設計。我們採用新提出的概念「覆蓋性 (inclusiveness)」去設計演算法。「分離性 (disjunctness)」是另一個常被使用的概念, 我們證明一個結合此兩

種概念的設計能顯著地改善譯解試驗結果的時間。除了探討正複合體的識別, 我們進一步從事複合體分類的工作, 也就是設計演算法去區分所有的複合體 (正的、負的及抑制型的)。 $(d, r; z]$ -分離性群試設計 ($(d, r; z]$ -disjunct pooling design) 是一個發展良好的演算法設計工具, 我們證明此工具足以用來處理複合體的分類工作而且也結合了快速的譯解程序。

總結下來, $(H, d; z]$ -分離性、 $(d, r; z]$ -分離性和 $(h, r; y]$ -覆蓋性是三種主要用來設計非調型演算法的工具。我們在此也討論它們的建構方法及所需試驗量的下界。

最後, 我們探討推廣化的傳統群試問題—圖形重建問題。群試問題可被視為一種搜尋式問題的範例, 且通常會在正元素的總量上做一個假設。而圖形的重建是另一種搜尋式的問題, 其主要工作是要識別出隱藏的圖形, 而已知條件是它是眾多可能圖形中的其中一個。其上用來作為識別的工具有種類似於群試驗是詢問: 一個詢問給的訊息是一個點的子集合是否包含隱藏圖形上的某個邊的所有的點。圖形重建的問題已有一些結果, 比如隱藏的圖形是一個漢米爾頓圈 (Hamiltonian cycle)、配對圖 (matching)、星圖 (star) 或局部完全圖 (clique)。針對這些問題, 我們利用一些策略去改進這些結果, 例如仿射平面法 (the affine plane method) 及配對結構法 (maximal matching method)。

誌謝

「練三分線外投籃，大多數的人會沿著三分線練投，若稍作調整，每次練投都離三分線遠一點，那沿著三分線投也不會有困難！」這是我的指導教授傅恆霖老師的見解，提醒我們做研究時該有的視野。

在做研究的道路上，傅老師總是提供我很多方向及可能性，並鼓勵我去嘗試。就在傅老師的鼓勵下，我得到千里馬計劃的贊助到 DIMACS Center 訪問一年，經過這段期間的磨練，讓我各方面都有實質的進步，研究做得更有心得，這些都要感謝傅老師的提攜。另一方面傅老師也希望我們培養運動的習慣，我後來也在打羽球過程中領會到老師的用意 – 打球競賽可以訓練專注力和企圖心以及培養合作的精神，這不就是做研究所需要的。非常感謝傅老師所教授的一切。

黃光明教授是帶領我進入數學研究殿堂的老師，不管是我在讀碩士時期或者近年來和老師的接觸，都讓我學習到很多不同層面的東西；黃老師也會點出我要改進的地方還有面對不同狀況該持有的態度，讓我受益良多，跟著黃老師做研究是件非常享受的事。不管是點出問題或是給與讚美，黃老師都帶給我很多正向的力量，感謝黃老師的帶領。

我要特別感謝陳秋媛老師，認識陳老師也好幾年了，感覺我們都像陳老師的大孩子一樣，陳老師總是在關鍵時刻給我很多的關愛和幫助，老師在學術上面也教我很多，特別是在老師的教學下讓我奠定演算法的基礎，感謝老師一直以來的鼓勵和教導。再來，我要感謝符麥克老師，符老師教學非常用心，有問題請教老師，他都會仔細地講解，我很幸運能跟符老師做研究，在討論過程中，老師都會傾囊相授，是我非常敬佩的老師。我還要感謝大學時認識的幾位老師，特別是蘇

用善老師、謝維華老師還有葉芳柏老師，他們的鼓勵和幫助對我有很深遠的影響，一直很感謝他們。

接著，我要謝謝學長姊們的幫助特別是賓賓和君逸學長；賓賓學長是非常照顧我們這些後進的學長，他會分享自己的觀察並提供做研究的各種面向，有問題請教學長，他也常提供成熟且客觀的意見；君逸學長是個很優秀的學長，學術經驗相當豐富而且都會毫無保留地分享，也在博士論文及口試準備上給我很多的幫助。再來一定要感謝一路幫助我，為我加油打氣的朋友們，馨華、雅靜、慧珊學姊、敏筠、康康、小巴、千砒、小貓、貓頭、智懷、威雄... 等等，受限於版面，還有很多未能提及的好伙伴，非常謝謝他們，有了他們讓我的研究生涯更充實有趣。

最後，要感謝我親愛的家人，感謝你們的鼓勵和支持，謹以此論文獻給你們。



Contents

Abstract	iii
中文摘要	v
誌謝	vii
Contents	ix
List of Tables	xi
List of Figures	xii
1 Introduction	1
1.1 Preliminaries on Algorithm	2
1.2 Models Originating from Applications in Molecular Biology . .	3
1.2.1 Group Testing with Inhibitors	3
1.2.2 Group Testing on Complexes	5
1.2.3 Graph Reconstruction Model	6
1.3 Thesis Overview	8
2 Nonadaptive Pooling Designs with Fast Decoding Procedures	10
2.1 The General Inhibitor Complex Model	11
2.1.1 Faster Procedures	15
2.2 The k -inhibitor Complex Model	18



3	Classification Problems on the Inhibitor Models	21
3.1	Nonadaptive Pooling Design for 1-inhibitor Complex Model . . .	22
3.1.1	The Inhibitor Clone Model	23
3.1.2	The Inhibitor Complex Model	25
3.2	Nonadaptive Pooling Design for k -inhibitor Clone Model . . .	27
4	Constructions of Related Disjunct Matrices	30
4.1	Lower Bound	31
4.2	Inclusiveness Property and Direct Constructions	33
4.3	Constructing by m -ary Method	35
4.4	Constructing by Controlling Row-covering	42
5	Reconstruction of Hidden Graphs	44
5.1	Preparation and Subroutines	44
5.2	Reconstructions of Simple Graphs	50
6	Conclusion and Remarks	55



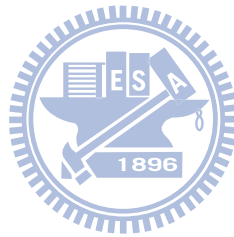
List of Tables

5.1 Examples of small order for Theorem 5.2.1 and Theorem 5.2.2 . . . 54



List of Figures

2.1	An example of Theorem 2.2.2	20
3.1	An example of Theorem 3.1.4	25
5.1	An example of FIND-ALL-PATHS algorithm	49



Chapter 1

Introduction

Given a set \mathcal{N} of n clones, each of which is either *positive* (usually called *defective*) or *negative* (usually called *good*), the *group testing problem* is to identify all positive ones by group tests. A *group test* is applied to a subset of \mathcal{N} with two possible outcomes; a *negative outcome* indicates if all clones in the subset are negative; a *positive outcome* indicates otherwise. In particular, a group test on a clone can show its property. Consequently, the main issue is to minimize the number of group tests in identifying all positive clones.

The origin of group testing can be traced back to World War II. The concept of group testing was first conceived in a session in the offices of the Price Statistics Branch of The Research Division of the office of Price Administration in Washington, D.C.. Researchers in the session such as David Rosenblatt and Robert Dorfman were struck by the wastefulness of testing blood samples from millions of draftees to detect a few thousand cases of syphilis. They suggested that pooling the blood samples may be economical (for more detail, please refer to Du and Hwang, 1993 [20]).

In the *probabilistic model* of group testing, a probability distribution is attached to the positive set and the expected number of tests required to identify positive elements is a criterion of efficiency. Robert Dorfman (1943) [19] studied the group testing problem under the probabilistic model and proposed a method that could eliminate all syphilitic men called up for induction. However, the need of group testing faded with the conclusion of

the World War II. Group testing stayed dormant for many years until the coming of its use in industry. Sobel and Groll (1959) [46], two Bell Laboratory scientists, considered many industrial applications of group testing and studied group testing under probabilistic models as well. Li (1962) [37] was the first to study *Combinatorial group testing* where probability distributions on positive set are completely eliminated; for instance, the number of positive items among the n items can be assumed at most d . Henceforward, combinatorial group testing developed alongside with the probabilistic group testing and has been prospering due to its applications in chemical leak testing, electric shorting detection, codes, multi-access channel communication and AIDS screening (see Du and Hwang, 1993 [20] and 2nd ed. 2000 [21] for a general reference). Recently, group testing has been found useful in molecular biology and is usually referred to as *pooling designs*. The new application also generates new models and new problems such as pooling designs on complexes (Torney, 1999 [52]), the inhibitor model (Farach *et al.*, 1997 [28]), contig sequencing, and non-unique probe selection problem (Du and Hwang, 2006 [23]).

1.1 Preliminaries on Algorithm

There are two general types of group testing algorithms: sequential and non-adaptive. A *sequential* algorithm conducts the tests one by one where the outcomes of all previous tests can be treated as a reference to the later one. A *nonadaptive* algorithm specifies all tests in advance and thus all tests can be conducted simultaneously. Sequential algorithms require fewer number of tests in general, since extra information allows for more efficient test designs while nonadaptive algorithms permit to conduct all tests simultaneously, thus saving the time for testing. Sequential algorithms have dominated the literature historically because the main goal of group testing is to minimize the number of tests required to identify all positive items. However, in some applications such as molecular biology, an experiment corresponding to a group

test is considerably time-consuming, thus it is impractical to perform the experiments sequentially. The focus then goes to nonadaptive group testing algorithms where all experiments are performed simultaneously; nevertheless, sequential procedures can still be used, but the total time required to identify the positive items must be considered along with the total number of tests. There is a natural tradeoff between the sequential and the nonadaptive algorithms. One can seek *2-stage* or *k-stage* algorithms for which all tests in a stage must be specified simultaneously, but the stages are sequential.

With experimental errors, test outcomes may contain 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. The error tolerance capability is concerned when proposing a design.

1.2 Models Originating from Applications in Molecular Biology

The wide range of conditions in which group testing has practical applications call for meaningful variants of the basic model in order to better accommodate the applications at hand. In this section, we introduce three models of group testing – inhibitor, complex, and graph reconstruction that originated from applications in molecular biology. These models have been studied in separate literatures. We will follow the original terminologies in each model.

1.2.1 Group Testing with Inhibitors

In certain applications, there is a third type of clones called *inhibitors* whose existence may cancel the effect of positive clones and the number of such clones in the population is usually assumed at most h . Various models can be formulated with inhibitors in the pooling design, depending on the interferences between inhibitors and positive clones. Farach *et al.* (1997) [28],

motivated by molecular biology applications, first proposed the 1-inhibitor model in which a single inhibitor clone dictates the testing outcome to be negative regardless of how many positive clones are in the test and gave a randomized algorithm to identify all positives in $O((d + h) \log n)$ tests. For example, in molecular biology, enzyme inhibitors are molecules that interact in some way with the enzyme to prevent it from working normally; in drug discovery applications, certain compounds can block the detection of a potent compound (Xie *et al.*, 2001 [54]); similar phenomena were mentioned in blood testing applications (Phatarfod and Sudbury, 1994 [44]).

De Bonis and Vaccaro (1998) [17] connected the 1-inhibitor model to a certain generalization of superimposed codes (D'yachkov and Rykov, 1983 [26]), and provided a lower bound $\Omega(\frac{h^2}{d \log h} \log n)$ on the number of tests required to identify exactly d positives in the presence of h inhibitors. Further, De Bonis *et al.* (2005) [16] gave an asymptotically optimal 4-stage algorithm for the 1-inhibitor model under the assumption that the exact number of positives and an upper bound on the number of inhibitors are known beforehand. Note that all these algorithms are sequential. Recently, nonadaptive pooling designs have been proposed for the inhibitor model (D'yachkov *et al.*, 2001 [25]; Hwang and Liu, 2003 [32]; Du and Hwang, 2005 [22]).

De Bonis and Vaccaro (2003) [18] extended the model to k -inhibitor model in which k inhibitor clones dictate the testing outcome to be negative. In general, one can consider a (k, g) -inhibitor model where k inhibitors cancel the effect of g positive clones.

Besides the mathematical complexity of dealing with various inhibitor models, determining which model fits the reality is also a practical question. Hwang and Chang (2007) [33] considered the *general inhibitor model* in such an environment with no need to know the exact relation between inhibitors and positive clones. De Bonis (2008) [15] proposed an almost optimal algorithm using $O(\frac{h^2}{d} \log(n/h))$ tests under the hypothesis that the exact number d of positives is given. Particularly, this algorithm is a trivial two-stage algorithm, that is, most non-positive candidates are eliminated by the first stage

and the remaining clones are tested separately in the second stage.

1.2.2 Group Testing on Complexes

The classical group testing problem has a set of elements each of which induces a positive or negative effect. In many DNA screening environments, a subset of clones (rather than a single clone), called a *complex*, can induce a positive effect. We call such a model the *complex model* in comparison with the *clone model* as previously discussed. Formally, in the complex model, we consider a given set H of complexes where a fixed but unknown subset of complexes are designated *positive*, while other candidates of positive complexes are called *negative complexes*. In particular, $H = \mathcal{N}$ is referred as the clone model. A group test is executed on a subset of \mathcal{N} and yields a positive outcome only when it contains at least one positive complex. To have an efficient design, we need to make some assumptions on the positive set. The simplest assumption is an upper bound d of the number of positive complexes in the test population. It is usually assumed that two positive complexes can overlap, but neither contains the other. Torney (1999) [52] first introduced the concept of the complex model and gave some substances in eukaryotic DNA transcription and RNA translation as examples of complexes.

Group testing on complexes is widely applied in modern molecular and cellular biology. A prominent example is its application in the identification of protein-to-protein interactions (Lappe and Holm, 2003 [36]; Li *et al.*, 2005 [38]). The interactions between proteins are significant for many biological functions. For example, in signal transduction process, the protein-to-protein interactions of the signaling molecules can convey signals from the exterior of a cell to the inside of that cell. This conveying process plays a fundamental role in living cells. Furthermore, information about the interactions between proteins improves our understanding of diseases and then provides the basis for new therapeutic approaches. Therefore, in many biological projects, identifying all protein-to-protein interactions is an essential task. The development of some laboratory approaches (Lappe and Holm,

2003 [36]) enables the application of group testing to this problem. Li *et al.* (2005) [38] formulated this identification problem as a group testing problem in bipartite graphs which can be regarded as a special case of group testing on complexes. Besides the protein-to-protein interactions problem, some other problems such as graph testing, superimposed codes and secure key distribution are also highly related to the complex model. Recent developments on this topic can be found in (Macula *et al.*, 2000 [42]; Macula *et al.*, 2004 [41]; Du and Hwang, 2006 [23]; Gao *et al.*, 2006 [29]; Chen *et al.*, 2007 [13]; Chen *et al.*, 2008 [14]).

Chang *et al.* (2010) [9] first introduced the *inhibitor complex model* where an inhibitor is a third type of complexes. Similar to the environments in the inhibitor clone model, the presence of an inhibitor may cancel the effect of positive complexes; in other words, a group test executed on a set of clones containing an inhibitor may yield a negative outcome even if that set contains a positive complex. Furthermore, the inhibitor complex model, as well as the inhibitor clone model, can be subdivided into the 1-inhibitor, k -inhibitor and general inhibitor models based on the interference effect between positive complexes and inhibitors. For instance, under k -inhibitor model a pool of clones inducing more than k inhibitors would yield a negative response.

1.2.3 Graph Reconstruction Model

Combinatorial search problems on graphs in the literature (Aigner, 1988 [6]) consist of identifying an unknown edge or vertex in a given graph, verifying a property of a hidden graph, reconstructing a hidden graph of a given class, and some others. The *graph reconstruction problem* we consider here is as follows. A hidden graph G is known belonging to a given family \mathcal{G} of labeled graphs on the set $[n] := \{1, 2, \dots, n\}$. The main task is to reconstruct G by asking queries as few as possible, where a query is of the form “Does S induce at least one edge of G ?”, denoted by $Q(S)$, for $S \subseteq [n]$, and $Q(S) = 1$, representing “yes”, or 0, representing “no”. Of course, the design of queries refers to the information provided by \mathcal{G} .

Different settings on the prior knowledge of the hidden graph produce various graph reconstruction problems. The group testing problem under complex model is a (hyper)graph-version of the graph reconstruction problem, where the vertices stand for the clones, edges stand for the complexes and the number of edges of the hidden graph is assumed at most d . Moreover, the hidden graph of bounded degree was studied in (Grebinski and Kucherov, 2000 [31]; Bouvel *et al.*, 2005 [8]), while the general hidden graph was considered in (Bouvel *et al.*, 2005 [8]; Angluin and Chen, 2008 [5]). We study the graph reconstruction problems under the assumption that the structure of the hidden graph is known.

Various families of hidden graphs have been studied. Many recent studies focus on two cases: Hamiltonian cycles and matchings (Grebinski and Kucherov, 1998 [30]; Beigel *et al.*, 2001 [7]; Alon *et al.*, 2004 [2]) which have specific application to the *genome sequencing problem*. In the genome sequencing, the *contigs*, which are longer continuous fragments formed from some overlapping short reads, cover the genome with possible gaps. The task is to determine the relative placement of contigs on the genome. A tool for doing this is an experiment called *multiplex Polymerase Chain Reaction (PCR)* (Sorokin *et al.*, 1996 [47]). In a multiplex PCR, an input of an experiment is a set of *primers*, which are short nucleotide sequences that characterize the ends of the contigs. Whenever the input set contains two primers corresponding to adjacent ends of neighboring contigs, the experiment outputs a reaction bringing a PCR product. Hence, the relative placement of contigs can be represented by the reaction graph which is a graph with primers as its vertices and pairs of vertices with reactions as its edges. In particular, for a circular genome, a reaction graph can be characterized as either a Hamiltonian cycle if the two primers of each contig are mixed together and are considered as a vertex, or a matching if primers are treated independently, i.e., each primer corresponds to a vertex. The problem can be generalized as to identify the pairs that react with each other among the given set of molecules (Torney, 1999 [52]; Alon and Asodi, 2005 [1]).

Sequential algorithms for graph reconstruction problems on some families of hidden graphs of known structure have been proposed. Grebinski and Kucherov (1998) [30] gave a sequential algorithm to reconstruct a Hamiltonian cycle in $2n \lg n$ queries ($\lg := \log_2$), while the information lower bound for the number of queries needed is $n \lg n$. Bouvel *et al.* (2005) [8] provided a sequential algorithm to reconstruct a matching in $(1 + o(1))(n \lg n)$ queries while $(1 + o(1))(\frac{n}{2} \lg n)$ is the best lower bound known so far and an algorithm to reconstruct a star in $2n$ queries while the information lower bound is $(1 + o(1))n$. They also proved that a clique of unknown size can be reconstructed in $2n$ queries while n queries are required in the worst case. There is still some room to improve the performance.

1.3 Thesis Overview

The inhibitor complex model, introduced by Chang *et al.* (2010) [9], is a new group testing environment with the allowance of the coexistence of inhibitors and complexes. In Chapter 2, we study group testing problem in the inhibitor complex model. We devote our attention to the studies of efficient nonadaptive designs with fast decoding procedures.

For group testing problems in the inhibitor model, much research has been devoted to the studies of identifying all positive items; however, only few studies have been done in classifying all items, especially for the nonadaptive designs. Furthermore, almost no work has been done in the classification problems under the inhibitor complex model. However, the identification of inhibitory substances is important in many practical applications; for example, many drugs are enzyme inhibitors because they can make the activity of enzymes reduced, thus leading to a destruction of a pathogen or a correction of a metabolic disturbance. In Chapter 3 we provide efficient nonadaptive algorithms for the classification problems under the 1-inhibitor model. It is notable that the pooling designs we propose have polynomial decoding procedures, i.e., determining the three types of complexes according to the

testing outcomes can be done in polynomial time. Finally, for k -inhibitor clone model, we solve the classification problems with both efficient non-adaptive algorithms and fast decoding procedures (work jointly with Chen and Fu, 2010 [10]).

Concluding from Chapter 2 and Chapter 3, we know that $(H : d; z)$ -disjunctness, $(d, r; z]$ -disjunctness, and $(h, r; y]$ -inclusiveness are three main properties of matrices employed as efficient designs. Many studies have been done on the constructions of $(H : d; z)$ -disjunct matrices and $(d, r; z]$ -disjunct matrices. In Chapter 4, we will introduce their constructions and some lower bounds that are mostly discussed in the literature. A matrix with $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive property was newly proposed in (Chang *et al.*, 2010 [10]; Chen, 2006 [12] for $r = 1$) and little literature is available on its construction. Accordingly, we provide some general results and prove that some well-constructed disjunct matrices have certain inclusiveness property.

In Chapter 5, we show some improvement on sequential algorithms for graph reconstruction problems. We employ an affine plane method (Tettelin *et al.*, 1996 [51]; Grebinski and Kucherov, 1998 [30]) together with constructing a maximal matching first and some other strategies to derive better algorithms (Chang *et al.* (2010) [10]). We improve the result in (Grebinski and Kucherov, 1998 [30]) on Hamiltonian cycle by a factor of $1/2$. We also provide algorithms to close up the gaps between lower and upper bounds for the numbers of queries required to reconstruct a matching and a star of unknown size. Further, we slightly improve the result in (Bouvel *et al.*, 2005 [8]) on clique by giving an algorithm with at most $n + \lg n$ queries.

Chapter 2

Nonadaptive Pooling Designs with Fast Decoding Procedures

In this chapter we study group testing problems in the inhibitor complex model. We devote our attention to nonadaptive designs that are not only efficient in terms of the number of tests, but also associated with fast decoding procedures.

A nonadaptive group testing scheme can be represented as a 0-1 (or binary) matrix where columns are labeled by clones and rows by tests. Thus row j intersects (has a 1-entry in) column i specifies that test j contains clone i . Sometimes it is convenient to view a column C_i as the set of tests (rows) containing the clone C_i . Thus $C_i \cap C_{i'}$ is the set of tests (rows) containing (intersecting) both C_i and $C_{i'}$. Accordingly, for a complex X , $\cap X := \bigcap_{C \in X} C$ is the set of rows intersecting all clones in X and we say a row j *covers* X if $j \in \cap X$.

For nonadaptive pooling designs, some enumerators are frequently used to differentiate complexes (or clones) of different properties. For example, let $\tau_0(X)$ denote the number of negative pools that complex X appears in. Then according to the testing outcomes of a design, this enumerator could be a cutoff function, i.e., there may be a fixed value, say a , such that $\tau_0(X) \leq a$ only when X is positive and thus distinguishing positive items from others.

Furthermore, for a set S of complexes (or clones), let $\tau_0^S(X)$ denote the same except that a negative pool that covers an element in S is not counted in. Similarly, let $\tau_1(\cdot)$ and $\tau_1^S(\cdot)$ denote the numbers of corresponding positive pools, respectively.

2.1 The General Inhibitor Complex Model

With the introduction of inhibitors to the clone models, Hwang and Chang (2007) [33] proposed the general inhibitor model in which the exact cancellation effect of inhibitors on positive clones is not specified. In the general inhibitor clone model, the $(d+h)$ -disjunct matrix is the main design to identify the positive clones from n clones, including at most d positive clones and at most h inhibitory clones. A binary matrix is d -disjunct if for any $d+1$ columns C_0, C_1, \dots, C_d ,

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

Chang *et al.* (2010) [9] are the first ones to study the general inhibitor complex model, and expand the idea of $(d+h)$ -disjunctness to this model.

We attach the parameters (n, d, h) to an inhibitor complex model with complex set H to denote the fact that among the complexes of H , which are subsets of the n clones, there are at most d positive complexes and at most h inhibitors. Following the terminology of (Gao *et al.*, 2006 [29]), a binary matrix is $(H : d; z)$ -disjunct if for any $d+1$ complexes X_0, X_1, \dots, X_d there exist z rows each covering X_0 but none of X_1, \dots, X_d , i.e.,

$$\left| \cap X_0 \setminus \bigcup_{i=1}^d \cap X_i \right| \geq z.$$

Let $t(n, (H : d; z))$ denote the minimum number of rows in an $(H : d; z)$ -disjunct matrix with n columns. Construction of $(H : d; z)$ -disjunct matrices was studied in (Gao *et al.*, 2006 [29]). When $z = O(1)$ and each

complex contains at most r clones, the construction yields a matrix with $O\left(\left(\frac{2dr \lg n}{\lg(dr \lg n)}\right)^{r+1}\right)$ rows (Corollary 4.3.4).

It is generally assumed that no complex is a subset of another for otherwise the requirement of $(H : d; z)$ -disjunctness cannot be fulfilled when X_0 is contained in one of the X_i 's.

A lower bound for the general inhibitor complex model is as follows, which is an extension of a result in (De Bonis and Vaccaro, 1998 [17]) for the general inhibitor clone model.

Theorem 2.1.1. *The number of rows in a nonadaptive pooling design under the (n, d, h) general inhibitor complex model with complex set H is at least $t(n, (H : h; 1))$.*

Proof. Since a lower bound of the 1-inhibitor complex model is clearly a lower bound of the general inhibitor complex model, it suffices to prove the 1-inhibitor case. Let M be the testing matrix of a nonadaptive pooling design. Suppose M is not $(H : h; 1)$ -disjunct. Then there exists a set of $h + 1$ complexes X_0, \dots, X_h such that every row covering X_0 must cover some of X_1, \dots, X_h . Consider the sample that X_0 is a positive complex and $\{X_1, \dots, X_h\}$ is the set of inhibitors. Then outcomes of the tests covering X_0 are negative and thus X_0 can not be identified from such outcomes. ■

Theorem 2.1.2. *An $(H : d + h; 1)$ -disjunct matrix can identify all positive complexes under the (n, d, h) general inhibitor complex model with complex set H .*

Proof. A positive complex appears in a negative pool only when the pool also contains some inhibitors. Thus, for a positive complex P , if S is an h -set containing all inhibitors, then

$$\tau_0^S(P) = 0$$

since all pools containing any inhibitor are excluded.

On the other hand, consider a non-positive complex X^* . By the definition of an $(H : d + h; 1)$ -disjunct matrix, for any other complexes X_1, \dots, X_{d+h} , there exists a row covering X^* but none of X_1, \dots, X_{d+h} . In particular, when $\{X_1, \dots, X_d\}$ contains all positive complexes and $\{X_{d+1}, \dots, X_{d+h}\}$ is a given set S , we have that the row yields a negative outcome. Thus

$$\tau_0^S(X^*) \geq 1$$

for any h -set $S \subseteq H \setminus \{X^*\}$. Consequently, $\{X : \tau_0^S(X) = 0 \text{ for some } h\text{-set } S \subseteq H \setminus \{X\}\}$ is the set of all positive complexes. ■

Next, for the error-tolerant case, we consider two types of errors: the (10)-type, changing 1-outcome to 0, and the (01)-type, changing 0-outcome to 1. Let e_{10}^* and e_{01}^* denote the unknown numbers of the (10)-type errors and the (01)-type errors, respectively, and denote upper bounds of e_{10}^* and e_{01}^* as e_{10} and e_{01} , either known or unknown. We assume that e , an upper bound of the total number of errors, is known, and set

$$c := \begin{cases} e_{10} + e_{01} - e & \text{if } e_{10} \text{ and } e_{01} \text{ are known,} \\ e & \text{if there are no estimates of } e_{10} \text{ and } e_{01}, \\ 0 & \text{if the number of positive complexes is } d. \end{cases}$$

Chang *et al.* (2010) [9] dealt with the error-tolerant case as follows.

Theorem 2.1.3. *An $(H : d + h; c + e + 1)$ -disjunct matrix can identify all positive complexes under the (n, d, h) general inhibitor complex model with complex set H which has at most e errors.*

Proof. Ignoring the inhibitors for the moment, then a positive complex P can appear in a negative pool only if its outcome is one of the (10)-type errors. Therefore, if S contains all inhibitors, then $\tau_0^S(P) \leq e_{10}^*$.

On the other hand, for a non-positive complex X^* , by the definition of $(H : d + h; c + e + 1)$ -disjunct, X^* appears in at least $c + e + 1$ rows each covering none of the up-to- d positive complexes, nor the h complexes in S ; hence the

corresponding tests have negative outcomes. Errors of the (01)-type may reduce the number of such negative pools. But still,

$$\tau_0^S(X^*) \geq c + e + 1 - e_{01}^* \geq c + 1 + e_{10}^*,$$

where the last inequality follows from $e_{10}^* + e_{01}^* \leq e$.

However, we do not know e_{10}^* and hence not knowing how to distinguish positive complexes from others. We consider three cases:

Case (1): e_{10} and e_{01} are known. Then $c = e_{01} + e_{10} - e$. Thus

$$\tau_0^S(X^*) \geq (e_{01} + e_{10} - e) + e + 1 - e_{01}^* \geq e_{10} + 1.$$

This implies that $\{X : \tau_0^S(X) \leq e_{10} \text{ for some } h\text{-set } S \subseteq H \setminus \{X\}\}$ is the set of all positive complexes.

Case (2): no estimates of e_{10} and e_{01} are given. Then $c = e$. Thus

$$\tau_0^S(X^*) \geq e + e + 1 - e_{01}^* \geq e + 1.$$

Hence, $\{X : \tau_0^S(X) \leq e \text{ for some } h\text{-set } S \subseteq H \setminus \{X\}\}$ is the set of all positive complexes.

Case (3): the number of positive complexes is known to be d . Then $c = 0$. Thus

$$\tau_0^S(X^*) \geq e_{10}^* + 1.$$

Therefore, $\{X : \min_{X \notin S} \tau_0^S(X) \text{ is among the } d \text{ smallest } \min_{X \notin S} \tau_0^S(\cdot) \text{ values}\}$ is the set of all positive complexes. ■

The decoding procedure requires to compute $\tau_0^S(X)$ for all h -subsets $S \subseteq H \setminus \{X\}$ and there are $\binom{|H|-1}{h}$ of them. However, $|H|$ can be much larger than $n = |\mathcal{N}|$. For example, if H contains all r -sets of clones, then $|H| = \binom{n}{r}$. Thus $\binom{|H|-1}{h}$ could be a very large number. In the following, we introduce some ways that reduce the decoding complexity in the order of magnitude.

2.1.1 Faster Procedures

For convenience, we use an (n, d, h, r) inhibitor complex model to denote an (n, d, h) inhibitor complex model where every complex contains at most r clones. Chang *et al.* (2010) [9] employ a seemingly unrelated notion, the $(d, r; z]$ -disjunct matrix, to tackle the problem. Moreover, this idea also provides a fast decoding procedure. A binary matrix is $(d, r; z]$ -disjunct if for any $r + d$ columns C_1, \dots, C_{r+d} , there exist z rows each intersecting C_1, \dots, C_r , but none of C_{r+1}, \dots, C_{r+d} , i.e.,

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

Let $t(n, (d, r; z])$ denote the minimum number of rows in a $(d, r; z]$ -disjunct matrix with n columns. The $(d : r; z]$ -disjunct matrix has been well studied (Stinson *et al.*, 2000 [50]; D'yachkov *et al.*, 2002 [27]; Stinson and Wei, 2004 [49]; Du *et al.*, 2006 [24]). See Chapter 4 for a general introduction.

Theorem 2.1.4. *A $(d + h, r; 2e + 1]$ -disjunct matrix can identify all positive complexes under the (n, d, h, r) general inhibitor complex model with error tolerance e .*

Proof. Consider a positive complex P and let $\{X_1, \dots, X_h\}$ denote a set of other complexes containing all inhibitors. Since no complex is contained in another, there exists a clone $v_i \in X_i \setminus P$ for $1 \leq i \leq h$. Let S' be an h -set containing these v_i 's such that $S' \cap P = \emptyset$. Then

$$\tau_0^{S'}(P) \leq e$$

since P can be in a negative pool only by the occurrence of error.

On the other hand, consider a non-positive complex X^* and let $\{X_1, \dots, X_d\}$ denote a set of other complexes containing all positive ones. Similarly, we can define $w_i \in X_i \setminus X^*$. Let D be a d -set containing these w_i 's and $D \cap X^* = \emptyset$. By the definition of a $(d + h, r; 2e + 1]$ -disjunct matrix, there exist at least $2e + 1$ rows each intersecting every columns in X^* and none of

the columns in $D \cup S$ for an arbitrary h -set $S \subseteq \mathcal{N}$ which is disjoint with X^* . Hence the outcomes of these $2e + 1$ pools should be negative except for the occurrence of errors. This implies that

$$\tau_0^S(X^*) \geq 2e + 1 - e = e + 1.$$

Hence $\{X : \tau_0^S(X) \leq e \text{ for some } h\text{-set } S \subseteq \mathcal{N} \setminus X\}$ is the set of positive complexes. ■

The decoding procedure demonstrated in the proof of Theorem 2.1.4 requires to compute $\tau_0^S(X)$ from the knowledge of the testing outcomes for each candidate complex $X \in H$ and every h -set $S \subseteq \mathcal{N} \setminus X$. Let $t = t(n, (d, r; z])$. Then each computation of $\tau_0^S(X)$ takes $O(t(h + r))$ and thus the time complexity of the decoding procedure is $O(t(h + r) \binom{n-r}{h} |H|)$ which could be a big deduction from $O(t'hr \binom{|H|-1}{h} |H|)$ in Theorem 2.1.3 where $t' = t(n, (H : d; z))$.

Chang *et al.* (2010) [10] provided an efficient design with a faster decoding procedure for the general inhibitor complex model where the improvement on decoding ability is attributed to the introduction of inclusiveness property to the design. A matrix is $(h, r; y]$ -*inclusive* if for any $h + r$ columns C_1, \dots, C_{r+h} , there are at most y rows each intersecting C_1, \dots, C_r and at least one of C_{r+1}, \dots, C_{r+h} , i.e.,

$$\left| \left(\bigcap_{i=1}^r C_i \right) \cap \left(\bigcup_{i=r+1}^{r+h} C_i \right) \right| \leq y.$$

Lemma 2.1.5. *A matrix which is $(d, r; z]$ -disjunct and also $(h, r; y]$ -inclusive with $z - y \geq 2e + 1$ is $(d + h, r; 2e + 1]$ -disjunct.*

Proof. For any $r + d + h$ columns C_1, \dots, C_{r+d+h} , there exist z rows intersecting each of C_1, \dots, C_r but none of C_{r+1}, \dots, C_{r+d} and at most y rows intersecting each of C_1, \dots, C_r and at least one of $C_{r+d+1}, \dots, C_{r+d+h}$. Then there remain at least $z - y \geq 2e + 1$ rows intersecting each of C_1, \dots, C_r but none of $C_{r+1}, \dots, C_{r+d+h}$. The theorem follows immediately. ■

Let $t(n, (d, h, r; x])$ denote the minimum number of rows in a $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive matrix with n columns for some z and y satisfying $z - y \geq x$.

From Lemma 2.1.5 and Theorem 2.1.4, we immediately have that a $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive matrix with $z - y \geq 2e + 1$ can identify all positive complexes under the (n, d, h, r) general inhibitor complex model with error tolerance e . However, the decoding ability of the design is not showed in the implication of Lemma 2.1.5. Especially, when every positive complex contains exactly r positive clones, we have the following advanced decoding procedure.

Algorithm 1:

Step 1. Implement a $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive matrix with $z - y \geq 2e + 1$ as a design.

Step 1: Evaluate $\tau_0(X)$ for every $X \in H$.

Step 2: Return $\{X \in H : \tau_0(X) \leq z - e - 1\}$.

Theorem 2.1.6. *Algorithm 1 can identify all positive complexes in $O(r|H|t(n, (d, h, r; 2e + 1]))$ decoding time under the (n, d, h, r) general inhibitor complex model with error tolerance e when each positive complex contains exactly r clones.*

Proof. Consider a positive complex P and let $\{X_1, \dots, X_h\}$ be a set of other complexes containing all inhibitors. Under the hypothesis that no complex is contained in another, there exist $v_i \in X_i \setminus P$ for $1 \leq i \leq h$. By $(h, r; y]$ -inclusiveness property, the number of pools containing P and at least one of v_i is at most y . Hence P can only appear in at most y negative pools if there is no error. This implies

$$\tau_0(P) \leq y + e.$$

On the other hand, consider a non-positive complex $X^* \in H$. Similarly, there exists a clone $v \in P \setminus X^*$ for each positive complex P . By the $(d, r; z]$ -disjunctness of the matrix, there are at least z rows each covering X^* and none of these v 's. Thus the pools corresponding to these rows yield negative outcomes if there is no error. Even in the worst case that all errors occur in these pools, we still have

$$\tau_0(X^*) \geq z - e > y + e.$$

Therefore, $\{X : \tau_0(X) \leq y + e\}$ is the set of positive complexes.

Since each computation of $\tau_0(X)$ takes $O(tr)$ time where $t = t(n, (d, h, r; 2e + 1])$, the time complexity of the decoding procedure is $O(tr|H|)$. ■

This procedure also results in a big deduction in computation, namely, from computing $\tau_0^S(X)$ to computing $\tau_0(X)$ where the measurement value is only calculated once for each potential candidate, leading to a considerable reduction in decoding complexity.

Notice that in Chapter 4, we will introduce some existing disjunct matrices that have certain inclusiveness property.

2.2 The k -inhibitor Complex Model

In the k -inhibitor complex model, the outcome of a test is positive if and only if it contains at least one positive complex and at most $k - 1$ inhibitors. While Section 2.1 provided nonadaptive pooling designs for this model, we now give a more efficient one.

Du and Hwang (2006) [23] used a $(d + h - k + 1, 1; 2e + 1]$ -disjunct matrix to solve group testing problem in the k -inhibitor clone model with error tolerance e . It can be easily extended to the complex model as follows.

Theorem 2.2.1. *A $(d + h - k + 1, r; 2e + 1]$ -disjunct matrix can identify all positive complexes under the (n, d, h, r) k -inhibitor complex model with error tolerance e .*

Let $\binom{\mathcal{N}}{h}$ denote the set consisting of all h -subsets of \mathcal{N} . Then the associated decoding procedure for Theorem 2.2.1 is to compute $\tau_0^S(X)$ for each $S \in \binom{\mathcal{N} \setminus X}{h-k+1}$ while $\{X \in H : \tau_0^S(X) \leq e \text{ for some } (h-k+1)\text{-subset } S \text{ of } \mathcal{N} \setminus X\}$ is the set of positive complexes.

According to Theorem 2.1.5 and Theorem 2.2.1, we obtain that a $(d, r; z]$ -disjunct and $(h-k+1, r; y]$ -inclusive matrix with $z-y \geq 2e+1$ can identify all positive complexes under the (n, d, h, r) k -inhibitor complex model with error tolerance e , but the decoding ability of such design has not been revealed yet. When every positive complex has exactly r clones, we show that the decoding algorithm can be improved.

Algorithm 2:

Step 1. Implement a $(d, r; z]$ -disjunct and $(h-k+1, r; y]$ -inclusive matrix with $z-y \geq 2e+1$ as a design.

Step 1: Evaluate $\tau_0(X)$ for every $X \in H$.

Step 2: Return $\{X \in H : \tau_0(X) \leq z-e-1\}$.

Theorem 2.2.2. *Algorithm 2 can identify all positive complexes in $O(r|H|t(n, (d, h-k+1, r; 2e+1]))$ decoding time under the (n, d, h, r) k -inhibitor complex model with error tolerance e when each positive complex contains exactly r clones.*

Proof. Since the implemented matrix is $(d, r; z]$ -disjunct, by the same argument used in the proof of Theorem 2.1.6 we have

$$\tau_0(X) \geq z - e$$

for any non-positive complex X .

On the other hand, let P be a positive complex and $\{X_1, \dots, X_{h-k+1}\}$ be a set of other complexes containing as many inhibitors as possible. Since no complex is included in another, we can take a $v_i \in X_i \setminus P$ for $1 \leq i \leq h-k+1$. By $(h-k+1, r; y]$ -inclusiveness property, the number of pools containing both P and at least one of v_i 's is no more than y . Since a pool containing P and

none of these v_i 's would be tested positive, P can only appear in at most y negative pools if there is no error. Thus

$$\tau_0(P) \leq z - e - 1.$$

We conclude that $\{X : \tau_0(X) \leq z - e - 1\}$ is the set of positive complexes. ■

Theorem 2.2.1 suggest a decoding algorithm of computing $\tau_0^S(X)$ for each $S \in \binom{\mathcal{N} \setminus X}{h-k+1}$ for each complex $X \in H$ while the decoding procedure shown in Theorem 2.2.2 is to compute $\tau_0(X)$ for each complex $X \in H$, a big reduction in computing.

Example 1. Consider the $(5, 1, 1, 2)$ 1-inhibitor complex model with $\mathcal{N} = \{1, \dots, 5\}$ and $H = \{12, 23, 13, 34, 15\}$ where ij denotes the complex consisting of clones i and j . Assume that no error is allowed and 23 is the inhibitor. In Figure 2.1, M is a $(1, 2; 2]$ -disjunct and $(1, 2; 1]$ -inclusive matrix (refer to Example 3 in Chapter 4 for a general construction). In the chart we can see that only 12, the only positive complex, can make the value of τ_0 lower than or equal to one.

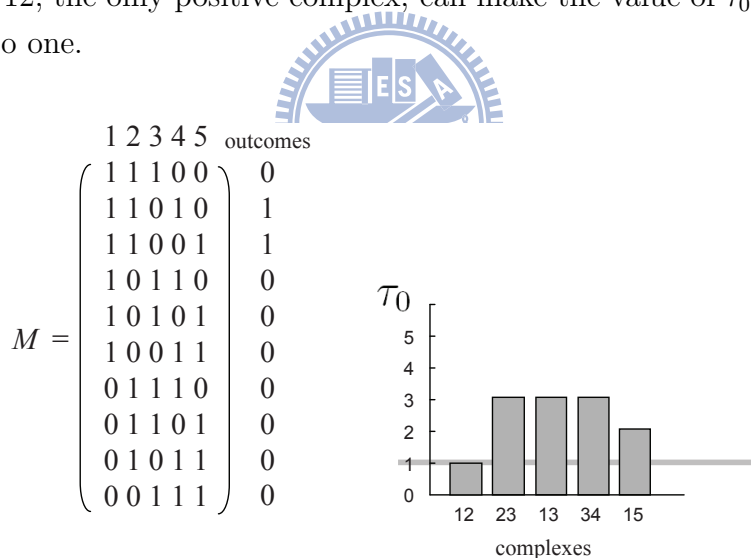


Figure 2.1: An example of Theorem 2.2.2

Chapter 3

Classification Problems on the Inhibitor Models

The problem we consider in this chapter is to classify all items in the inhibitor clone/complex models. Some multi-stage algorithms that were proposed to identify positive elements are to identify and then remove almost all inhibitors at the first stages (Farach *et al.*, 1997 [28]; Hwang and Liu, 2003 [32] under the inhibitor clone model; De Bonis and Vaccaro, 2003 [18] under the k -inhibitor clone model). Of course, one could accomplish the classification work by extending these results. However, very little is known about nonadaptive pooling designs for the classification problem. An interesting feature is that a trivial strategy does not work for identifying inhibitors, i.e., one can not simply test every item to classify the whole set. We propose a nonadaptive pooling design to classify all items by starting with the identification of inhibitors (Chang *et al.*, 2010 [10]). Our approach is to strengthen the parameters of $(d, r; z]$ -disjunct type matrix such that the design generated from the matrix is sufficient to identify all inhibitors and also contains enough pools, where inhibitors lost their cancellation effect, to identify positive items. In the following we first introduce the results in 1-inhibitor model and then extend them to the k -inhibitor model.

3.1 Nonadaptive Pooling Design for 1-inhibitor Complex Model

In order to distinguish inhibitors from negatives, we need to make an additional assumption: (A) *Among the given complexes in H , there exists at least a positive one.* The reason for this is that one cannot distinguish negative complexes from inhibitors without any positive complex. In addition, for the inhibitor complex model with $r \geq 2$, we need another essential assumption on complexes: (B) *For each negative complex, there is always a positive complex such that no inhibitor is included in their union.* Otherwise, any test containing the negative complex that violates the assumption must yield a negative outcome and thus the recognition of this complex would be ambiguous. Due to the naturalness of these two assumptions, we take them as default properties on complexes throughout this section. The following result was obtained by Chang *et al.* (2010) ([10]).

Theorem 3.1.1. *An $(h, 2r; 2e + 1]$ -disjunct matrix can identify all inhibitors under the (n, d, h, r) 1-inhibitor complex model with error tolerance e .*

Proof. Consider a positive complex P and let $\{X_1, \dots, X_h\}$ be a set of other complexes containing all inhibitors. Since no complex is contained in another, there exists $v_i \in X_i \setminus P$ for $1 \leq i \leq h$. By $(h, 2r; 2e + 1]$ -disjunctness property, there exist at least $2e + 1$ rows each containing P but none of v_i 's. The pools corresponding to these rows must be tested positive if no erroneous outcome occurs. Hence,

$$\tau_1(P) \geq e + 1$$

even in the worst case that e erroneous outcomes occur.

Next, consider a negative complex X^- . According to the assumption (B) on complexes, there exists a positive complex P such that there is a clone $v \in I \setminus (P \cup X^-)$ for each inhibitor I . By $(h, 2r; 2e + 1]$ -disjunctness property, there exist at least $2e + 1$ rows each containing P and X^- , but none of these v 's. Hence, we have that

$$\tau_1(X^-) \geq e + 1$$

despite e erroneous outcomes.

On the other hand, since an inhibitor appears in a positive pool only when an erroneous outcome occurs,

$$\tau_1(X^*) \leq e$$

for any inhibitor X^* . Thus, we conclude that $\{X : \tau_1(X) \leq e\}$ is the set of inhibitors. ■

An interesting observation coming from this theorem is that the number of tests required for identifying inhibitors does not depend on the number of positive complexes while the number of inhibitors is significant to the number of tests required for identifying all positives in the inhibitor model.

For inhibitor clone model, after identifying all inhibitors, one can remove them and then continue to identify positive ones; however, this strategy can not be implemented to the complex model due to intersections between complexes. In the following, we deal with the clone model and the complex model separately.

3.1.1 The Inhibitor Clone Model

For the inhibitor clone model, following Theorem 3.1.1, a two-stage algorithm to classify all clones could be to identify and eliminate all inhibitors by an $(h, 2; 2e + 1]$ -disjunct matrix in the first stage and then turn to study the clone model in the second stage. The group testing problem in the clone model has been well studied in the literatures and a main design for this model is as follows.

Lemma 3.1.2. *A $(d, 1; 2e + 1]$ -disjunct matrix can identify all positive clones under the (n, d) clone model with error tolerance e ; furthermore, it can be concluded from the design that $\{v \in \mathcal{N}; \tau_0(v) \leq e\}$ is the set of positive clones.*

According to Theorem 3.1.1 and Lemma 3.1.2, it is quite natural to consider a matrix that is $(h, 2; 2e + 1]$ -disjunct and satisfies the following condition: (*) deleting any h columns and all rows intersecting them would yield

a $(d, 1; 2e + 1]$ -disjunct matrix. Again, Chang *et al.* (2010) ([10]) proved that a $(d + h, 2; 2e + 1]$ -disjunct matrix can indeed accomplish this job based on the following general result.

Lemma 3.1.3. *For any $d \geq d'$ and $r \geq r'$, a $(d, r; z]$ -disjunct matrix is $(d', r'; z]$ -disjunct and the $(d', r'; z]$ -disjunctness property is preserved after deleting any $d - d'$ columns and all rows intersecting them.*

Proof. The first part of the statement is clear. Consider the second part. Let M be a $(d, r; z]$ -disjunct matrix with column index set $[n]$ and S be a $(d - d')$ -subset of $[n]$. Let M' be the matrix obtained from M by deleting columns corresponding to indices in S and rows intersecting them. Let D and R be two disjoint subsets of $[n] \setminus S$ with $|D| \leq d'$ and $|R| \leq r'$. Any row of M that intersects all columns of $M(R)$ and none of the columns of $M(S \cup D)$ is preserved in M' where $M(S)$ denotes the submatrix of M obtained by restricting the column indices to S . Thus the number of rows intersecting all columns of $M'(R)$ and none of columns of $M'(D)$ is at least z . ■

Therefore, a $(d + h, 2; 2e + 1]$ -disjunct matrix is also $(h, 2; 2e + 1]$ -disjunct and satisfies (*) and thus it can classify all clones. Here, we give a proof relating to decoding procedure.

Theorem 3.1.4. *A $(d + h, 2; 2e + 1]$ -disjunct matrix can classify all clones under the (n, d, h) 1-inhibitor model with error tolerance e .*

Proof. A $(d + h, 2; 2e + 1]$ -disjunct matrix is also $(h, 2; 2e + 1]$ -disjunct and thus by Theorem 3.1.1, we immediately obtain that

$$\mathcal{I} := \{v \in \mathcal{N} : \tau_1(v) \leq e\}$$

is the set of inhibitors. Consider the matrix M' obtained from deleting columns corresponding to inhibitors and rows intersecting them. Notice that the testing outcome for each pool in M' inherits the outcome of its corresponding pool in M , showing no additional tests are required. By Lemma 3.1.3, M' is $(d, 1; 2e + 1]$ -disjunct; hence, by Lemma 3.1.2,

$$\{v \in \mathcal{N} \setminus \mathcal{I} : \tau_0(v) \leq e\}$$

is the set of all positive clones where the computing of $\tau_0(v)$ refers to the pools in M' . ■

Since the computing of $\tau_1(v)$ and $\tau_0(v)$ takes $O(t)$ time, the decoding procedure for such design takes $O(tn)$ time where $t = t(n, (d + h, 2; 2e + 1])$.

Example 2. Consider the $(5, 1, 1)$ 1-inhibitor clone model on $\mathcal{N} = \{1, \dots, 5\}$. Assume that no error is allowed, i.e., $e = 0$. Consider that clone 1 is the inhibitor and clone 2 is the positive clone. In Figure 3.1, M is a $(2, 2; 1]$ -disjunct matrix (see Chapter 4 for general constructions). In chart (a), we can see that only 1, the only inhibitor, can make the value of τ_1 lower than or equal to $e = 0$. M is then shrunk to a 1-disjunct matrix M' where columns represent all clones except inhibitors. Chart (b) shows that only 2, the only positive clone, has the value of τ_0 lower than or equal to $e = 0$.

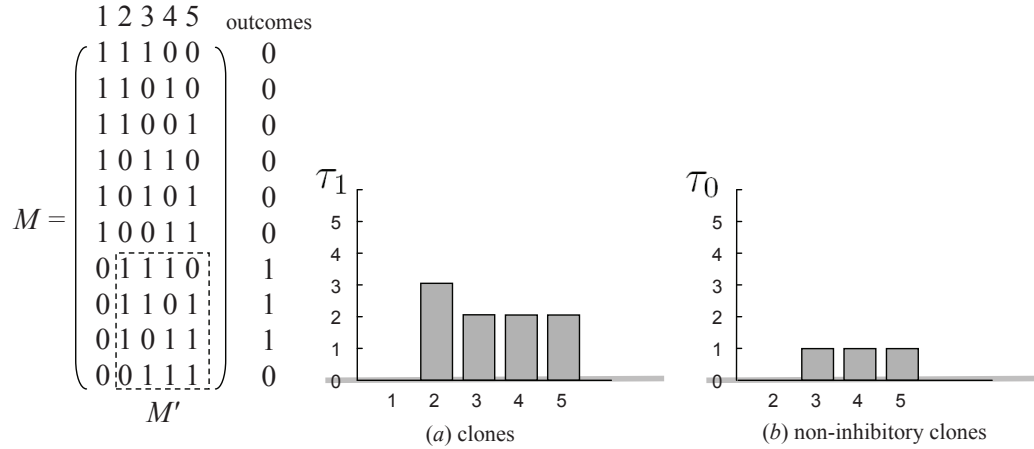


Figure 3.1: An example of Theorem 3.1.4

3.1.2 The Inhibitor Complex Model

We know that an $(h, 2r; 2e + 1]$ -disjunct can identify all inhibitors for the (n, d, h, r) inhibitor complex model (Theorem 3.1.1) and a $(d, r; 2e + 1]$ -disjunct can identify all positives for the (n, d, r) complex model (Theorem

2.1.4). Thus learning from Theorem 3.1.4, one may use a $(d + h, 2r; 2e + 1]$ -disjunct matrix to tackle the classification problem. However, unlike the inhibitor clone model, for the complex model, after identifying inhibitors, we can not simply remove them because it could break other complexes and thus would affect their identification. However, the cutoff function $\tau_0^S(X)$ provides a way to overcome this problem. When S is disjoint with X and contains a clone from each inhibitor, each negative pool counted in $\tau_0^S(X)$ is not due to the appearance of inhibitors and thus any complex covered by the pool can be identified as negative. The following result is obtained by following this idea.

Theorem 3.1.5. *A $(d + h, 2r; 2e + 1]$ -disjunct matrix can classify all complexes under the (n, d, h, r) 1-inhibitor complex model with error tolerance e .*

Proof. First, since a $(d + h, 2r; 2e + 1]$ -disjunct matrix is $(h, 2r; 2e + 1]$ -disjunct, by Theorem 3.1.1, $\mathcal{I} := \{X : \tau_1(X) \leq e\}$ is the set of inhibitors. Assume that $X_1, X_2, \dots, X_{h'}$ are the inhibitors. Let \mathcal{I}_X be a set that contains a clone in $X_i \setminus X$ for $1 \leq i \leq h'$. Define $\tau_{0, \mathcal{I}}(X) = \tau_0^{\mathcal{I}_X}(X)$.

A positive complex P can appear in a negative pool only when an inhibitor also appears in it or its testing result is fault. Thus

$$\tau_0^{\mathcal{I}_P}(P) \leq e$$

since a pool containing an inhibitor and thus some clone in \mathcal{I}_P is not evaluated in the computation.

On the other hand, consider a negative complex X^* . Assume that D is a set consisting of a clone in $X \setminus X^*$ for each positive complex X . A $(d + h, 2r; 2e + 1]$ -disjunct matrix is also $(d + h, r; 2e + 1]$ -disjunct; hence, there are $2e + 1$ rows covering X^* but none of clones in $D \cup \mathcal{I}_{X^*}$. Then

$$\tau_0^{\mathcal{I}_{X^*}}(X^*) > e$$

since each pool corresponding to any of these $2e + 1$ rows contains no positive complex and thus yields a negative outcome if there is no error, and it is

evaluated in the computation due to the fact that it contains no clone in \mathcal{I}_{X^*} . Hence $\{X \in H \setminus \mathcal{I} : \tau_{0,\mathcal{I}}(X) \leq e\}$ is the set of positive complexes. ■

Indeed, a decoding procedure for this design is to distinguish inhibitors from other complexes by the cutoff function $\tau_1(\cdot)$ and then distinguish positive complexes from negative ones by the cutoff function $\tau_{0,\mathcal{I}}(\cdot)$. Since the computing of $\tau_1(X)$ and $\tau_{0,\mathcal{I}}(X)$ takes $O(t(rh))$ time (including the setting of \mathcal{I}_X), the procedure takes $O(t(rh)|H|)$ time where $t = t(n, (d+h, 2r; 2e+1])$.

3.2 Nonadaptive Pooling Design for k -inhibitor Clone Model

In this section we consider the k -inhibitor model where a test yields a positive outcome if and only if it contains at least one positive clone and less than k inhibitors. It is assumed that the threshold k is known beforehand. In order to identify inhibitors, besides the assumption (A), another assumption is also essential: (C) *Among the given clones, there exist at least k inhibitors*. Otherwise, inhibitors do not have enough ability to obscure positive clones and thus there is no way to differentiate them from negative ones. A bundle of arbitrary k inhibitors has blocking effect while other clones (not all inhibitors) can't. Chang *et al.* (2010) ([10]) used this characteristic to prove the following result which is an extension of Theorem 3.1.1 from $k = 1$ to a general $k \geq 1$.

Theorem 3.2.1. *An $(h - k + 1, k + 1; 2e + 1]$ -disjunct matrix can identify all inhibitors under the (n, d, h, r) k -inhibitor clone model with error tolerance e .*

Proof. For any k -set K of inhibitors, it is obvious that

$$\tau_1(K) \leq e.$$

Consider a set K of k clones not all inhibitors. Let P be a positive clone (that can be in K) and S be a set of $h - k + 1$ clones containing as many

inhibitors not in K as possible. By the $(h - k + 1, k + 1; 2e + 1]$ -disjunctness property, there exist at least $2e + 1$ rows each intersecting P and all clones in K but none in S . Then each pool corresponding to any of these rows contains a positive clone and at most $k - 1$ inhibitors, implying its testing outcome is positive except an occurrence of error. Thus

$$\tau_1(K) \geq e + 1.$$

Therefore, $\bigcup\{K \subseteq \mathcal{N} : \tau_1(K) \leq e, |K| = k\}$ is the set of inhibitors. \blacksquare

The computing of $\tau_1(K)$ takes $O(kt)$ time and thus the overall decoding procedure takes $O\left(\binom{n}{k}kt\right)$ time.

In the previous section, we discussed a two-stage algorithm to classify all clones under the 1-inhibitor model where the first stage is to identify (and eliminate) all inhibitors by a disjunct matrix and the sequential stage is to distinguish positive clones from negative ones by another disjunct matrix. We can extend this idea to produce a two-stage algorithm for the k -inhibitor model, but with the following modification in the first stage: use an $(h - k + 1, k + 1; 2e + 1]$ -disjunct matrix (instead of an $(h, 2; 2e + 1]$ -disjunct) to identify inhibitors and then remove either all of them or exactly $h - k + 1$ of them so that the remaining inhibitors, at most $k - 1$, do not obscure the positive clones.

Again, from Lemma 3.1.3, a nonadaptive pooling design obtained from combining an $(h - k + 1, k + 1; 2e + 1]$ -disjunct matrix and a $(d, 1; 2e + 1]$ -disjunct matrix as follows can classify all clones. The following proof is given in the perspective of decoding.

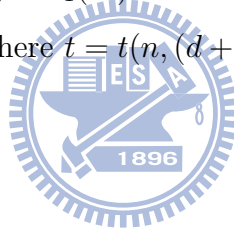
Theorem 3.2.2. *A $(d + h - k + 1, k + 1; 2e + 1]$ -disjunct matrix can classify all clones under the (n, d, h) k -inhibitor model with error tolerance e .*

Proof. A $(d + h - k + 1, k + 1; 2e + 1]$ -disjunct matrix is also $(h - k + 1, k + 1; 2e + 1]$ -disjunct and then by Theorem 3.2.1, we immediately obtain that the union \mathcal{I} of all k -sets K of clones with $\tau_1(K) \leq e$ is the set of inhibitors. Now focus on the sub-matrix M' obtained from deleting $\min(|\mathcal{I}|, h - k + 1)$

columns corresponding to inhibitors and rows intersecting them. Then the columns of M' relate to at most $k - 1$ inhibitors and hence M' could be used as a design for the clone model. Since at most $h - k + 1$ columns are deleted, M' is $(d, 1; 2e + 1]$ -disjunct by Lemma 3.1.3. Then by Lemma 3.1.2, $\{v \in \mathcal{N} \setminus \mathcal{I}; \tau_0(v) \leq e\}$ is the set of positive clones where the computing of $\tau_0(v)$ refers to the pools in M' and the outcome of each pool coincides with the outcome of its expanded pool in M because deleted columns do not intersect it. ■

Notice that in Theorem 3.1.4 for 1-inhibitor model, all columns associated with inhibitors are deleted but in Theorem 3.2.2 only at most $h - k + 1$ columns of inhibitors are deleted. Such deletion is proper because the inhibitors corresponding to the remaining columns do not have the ability of obscuring positives and the remaining matrix still maintain the ability of solving classical group testing problem.

The decoding procedure for this design is to compute $\tau_0(v)$ for each $v \in \mathcal{N} \setminus \mathcal{I}$ besides the computing of $\tau_1(K)$ for each $K \in \binom{\mathcal{N}}{k}$, and thus its time complexity is $O\left(\binom{n}{k}kt\right)$ where $t = t(n, (d + h - k + 1, k + 1; 2e + 1])$.



Chapter 4

Constructions of Related Disjunct Matrices

In the previous chapters, three main properties of matrices employed as nonadaptive pooling designs are $(H : d; z)$ -disjunct, $(d, r; z]$ -disjunct, and $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive with $z > y$. Many strategies were used to construct the related matrices: constructing by design theory and set intersections, transforming an m -ary matrix with certain properties to a binary one, called *m-ary method* in (Du and Hwang, 2006 [23]), and controlling the number of rows covering or not covering a certain number of columns, called *row-covering method*.

Before proceeding to see the constructions, we present some basic definitions and notations. We start with some notations on graph theory and then coding theory.

Let H be the given set of complexes in the considered problem. Then H can be viewed as a hypergraph with clones as vertices and complexes as edges and accordingly, it is usually assumed that no edge contains another. A hypergraph is usually represented by (V, E) where V is its vertex set and E is its edge set. The *degree* of a vertex is the number of edges that it belongs to while the *rank* of an edge is the number of vertices that it contains. A hypergraph in which all vertices have the same degree is said to be *regular*; a hypergraph where all edges have the same rank is called *uniform*. Let $H_{\bar{r}}$

denote a hypergraph where the maximum rank is r and H_r^* the hypergraph with edge set $\binom{V}{r}$.

A *code* is a set of vectors called *codewords* and has three primary parameters: length, size and Hamming distance. The number of entries in a codeword is its *length* and is also the *length* of a code if all codewords have the same length; the *size* of a code is the number of codewords in it; the *Hamming distance* of a code is the minimum number of nonidentical symbols between two codewords where the minimum is taken over all pairs of codewords. Moreover, an m -ary code is a code whose symbols are from the m -ary alphabet $\{0, 1, \dots, m - 1\}$. For an m -ary code C of length t , the *incident matrix* of C is a $t \times |C|$ m -ary matrix whose columns are codewords of C .

4.1 Lower Bound

Stinson *et al.* (2000) [50] considered the generalized cover-free family which is equivalent to $(d, r; z]$ -disjunct design. They derived a lower bound for the case $z = 1$ by a recursive relation. Stinson and Wei (2004) extended the method to a general z by induction on $r + d$. The basic cases and a recursive relation are as follows.

Theorem 4.1.1. $t(n, (d, 1; z]) \geq c \left(\frac{d^2 \log n}{\log d} + (z - 1)d \right)$ for some absolute constant c .

The basic case $d = 1$ is the same as the case $r = 1$ according to the following result.

Lemma 4.1.2. $t(n, (d, r; z]) = t(n, (r, d; z])$.

Proof. Interchanging 0 and 1 in a $(d, r; z]$ -disjunct matrix yields an $(r, d; z]$ -disjunct matrix. ■

Theorem 4.1.3. $t(n, (d, r; z]) \geq t(n - 1, (d - 1, r; z]) + t(n - 1, (d, r - 1; z])$.

Proof. Let M be a $(d, r; z]$ -disjunct matrix. By Lemma 3.1.3, deleting a column of M and all rows intersecting it yields a $(d-1, r; z]$ -disjunct matrix. Similarly, deleting a column of M and all rows not intersecting it yields a $(d, r-1; z]$ -disjunct matrix. ■

This recursion leads to a lower bound for $t(n, (d, r; z])$.

Theorem 4.1.4. For $d+r > 2$,

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

where c is the same constant as in Theorem 4.1.1.

Proof. The proof is by induction on $r+d$. The case $r=1$ or $d=1$ is easily obtained from Theorem 4.1.1 and Lemma 4.1.2. For $d \geq 2$ and $r \geq 2$,

$$\begin{aligned} t(n, (d, r; z]) &\geq t(n-1, (d-1, r; z]) + t(n-1, (d, r-1; z]) \\ &\geq c \binom{d+r}{r} \left(\frac{2 \log(n-2)}{\log(d+r-1)} + \frac{z-1}{2} \right) \\ &\geq c \binom{d+r}{r} \left(\frac{2 \log(n-1)}{\log(d+r)} + \frac{z-1}{2} \right). \end{aligned}$$

Stinson and Wei (2004) [49] further gave a stronger lower bound by a similar argument.

Theorem 4.1.5. There exists an integer $n_{d,r}$ such that for $n \geq n_{d,r}$,

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

where c is the same constant as in Theorem 4.1.1.

4.2 Inclusiveness Property and Direct Constructions

As mentioned in Chapter 2, a $(d, r; z]$ -disjunct and $(h, r; y]$ -inclusive matrix with $z > y$ has a great contribution to simplifying the decoding procedure.

The following result is an immediate consequence of Lemma 2.1.5 and Theorem 4.1.5.

Theorem 4.2.1. $t(n, (d, h, r; 2e + 1]) \geq t(n, (d + h, r; 2e + 1])$

$$\geq 0.7c \frac{(d + h + r) \binom{d+h+r}{r}}{\log \binom{d+h+r}{r}} \log n + c \binom{d + h + r}{r} e.$$

However, constructions of such matrices were rare. We observe the following general result.

Lemma 4.2.2. *For a binary matrix M , if any r columns are covered by at least ω rows and any $r + 1$ columns are covered by at most λ rows, then M is $(d, r; \omega - d\lambda]$ -disjunct and $(h, r; h\lambda]$ -inclusive.*

Proof. There are at most λ rows each intersecting given r columns and any other column and thus at most $h\lambda$ rows each intersecting r columns and some of other h columns. Furthermore, since there are at least ω rows that r columns share in common, the number of rows covering given r columns but none of other d columns is at least $\omega - d\lambda$. ■

For $r = 1$, the direct construction of disjunct matrices in (Hwang and Sós, 1987 [34]) satisfies the condition in Lemma 4.2.2, implying the following result (Chang *et al.*, 2010 [10]).

Theorem 4.2.3. $t(n, (d, h, r; 2e + 1]) \leq 16(d + h + 2e)^2 \lg(3n/2) / \lg 3$.

Lemma 4.2.2 can also be used to check the associated properties of the matrices derived from T -designs. A T - (ν, k, λ) design is a collection of k -subsets, called blocks, of a set of ν points such that for any T points there

exist exactly λ blocks containing those T points (Anderson, 1990 [3]). According to the Fisher inequality, for a T -design, the number of blocks is not smaller than the number of points and thus the incidence matrix of a T -design with blocks as rows and points as columns is not a good pooling design for clone models. However, T -designs become feasible for the complex models since the number of tests could be less than $\binom{\nu}{r}$, the number of all potential candidates of positive complexes. Mitchell and Piper (1988) [40] gave a construction of $(d, r; 1]$ -disjunct matrix based on T -designs. Chang *et al.* (2010) [10] extended their results to an error-tolerant version and extracted their inclusiveness property.

Theorem 4.2.4. *A T - (ν, k, λ) design yields a $t \times \nu$ $(d, T-1; \omega - d\lambda]$ -disjunct and $(h, T-1; h\lambda]$ -inclusive matrix for $d, h < \min(\omega/\lambda, \nu - T + 1)$ where*

$$t = \frac{\binom{\nu}{T}\lambda}{\binom{k}{T}} \text{ and } \omega = \lambda \frac{\nu - T + 1}{k - T + 1}.$$

Moreover, its error tolerance achieves $\lceil \frac{\omega - \lambda(d+h)}{2} \rceil - 1$.

Proof. We first consider the inclusiveness property. For any set S of $T-1$ columns, there are at most λ rows covering S and any given column not in S , and thus at most $h\lambda$ rows covering S and any given h columns other than those in S for any $1 \leq h \leq \nu - T + 1$.

Next, for any set S of $T-1$ columns, $|\{(v, B) : B \text{ is a block containing } S \text{ and } v \in B \setminus S\}| = (\nu - T + 1)\lambda$ since for each point v not in S there are exactly λ blocks containing $S \cup \{v\}$. Thus the number of blocks containing S is $(\nu - T + 1)\lambda / (k - T + 1)$. Therefore, the theorem immediately follows from Lemma 4.2.2. ■

Example 3. A 3 - $(q^2 + 1, q + 1, 1)$ design always exists for prime power q (Stinson, 1997 [48]) and its $q(q^2 + 1) \times (q^2 + 1)$ incidence matrix is $(d, 2; q + 1 - d]$ -disjunct and $(h, 2; h]$ -inclusive.

Some constructed $(d, r; z]$ -disjunct matrices potentially satisfy inclusive, especially when the number of rows covering any designated r columns is lower bounded by a certain number.

Lemma 4.2.5. *Let M be a binary matrix in which the number of rows covering any designated r columns is w . Then M is $(h, r; z_h]$ -disjunct if and only if M is $(h, r; w - z_h]$ -inclusive.*

D'yachkov *et al.* (2002) [27] gave a simple construction of $(d, r; 1]$ -disjunct matrices by taking all k -subsets of $[n]$ as the rows and then it is further extended to the error-tolerant case (Du *et al.*, 2006 [24]). We observe its inclusiveness property as follows.

Theorem 4.2.6. *The $\binom{n}{k} \times n$ binary matrix where the rows consist of all k -subsets of $[n]$, $r \leq k \leq \min(n-d, n-h)$, is $(d, r; z_d]$ -disjunct and $(h, r; y_h]$ -inclusive, where*

$$z_d = \binom{n-d-r}{k-r}, \quad y_h = \binom{n-r}{k-r} - z_h.$$

Moreover, $z_d - y_h > 0$ for $h, d \ll n$.

Proof. It is easily derived that this matrix is $(d, r; \binom{n-d-r}{k-r}]$ -disjunct for $r \leq k \leq n-d$. Given an r -set R , the number of rows covering R is $\binom{n-r}{k-r}$. By Lemma 4.2.5, we immediately have the theorem. Furthermore, $z_d - y_h = \binom{n-r-d}{k-r} + \binom{n-r-h}{k-r} - \binom{n-r}{k-r} > 0$ for $h, d \ll n$. ■

Note that taking $k = r$ or $n-d$ would minimize the row number $\binom{n}{k}$ and copying each row in a $(d, r; 1]$ -disjunct z times would yield a $(d, r; z]$ -disjunct matrix. Hence,

Corollary 4.2.7. $t(n, (d, r; z]) \leq z \min(\binom{n}{d}, \binom{n}{r})$.

4.3 Constructing by m -ary Method

$(H : d; z)$ -disjunct matrices are basic pooling designs for complex model. For the clone model, Du *et al.* (2006) [24] gave a construction of d -disjunct matrices by first constructing an m -ary matrix satisfying certain property and then converting it to a binary one. Gao *et al.* (2006) [29] extended the construction to the complex model where the m -ary matrix used to be converted satisfies the following property.

Definition 1. An m -ary matrix is $(H : d; z)$ -disjunct if for any $d + 1$ edges X_0, X_1, \dots, X_d , there exist a least z rows in each of which

$$\{\text{entries of } X_i\} \not\subseteq \{\text{entries of } X_0\}$$

for $i = 1, \dots, d$.

Let $t_m(n, (H : d; z))$ denote the minimum number of rows in an m -ary $(H : d; z)$ -disjunct matrix with n columns.

Gao *et al.* (2006) [29] gave a construction of m -ary $(H_{\bar{r}} : d; z)$ -disjunct matrix. Let $GF(q)$ be a finite field of order q . Suppose $q^{k+1} \geq n$. Associate each vertex $v \in \mathcal{N}$ with a distinct polynomial p_v of degree k over $GF(q)$. Let S be a subset of s elements in $GF(q)$. Construct an $s \times |\mathcal{N}|$ q -ary matrix $A_{H_{\bar{r}}}(q, k, s)$ with rows labeled by S and columns by \mathcal{N} where each cell (a, v) is assigned the element $p_v(a)$ in $GF(q)$. Then,

Lemma 4.3.1. *If $q \geq s \geq drk + z$ and $q^{k+1} \geq n$ where q is a prime power, then $A_{H_{\bar{r}}}(q, k, s)$ is an $s \times n$ q -ary $(H_{\bar{r}} : d; z)$ -disjunct matrix.*

Proof. Let $P_X(a)$ denote the set $\{p_v(a) : v \in X\} = \{\text{entries of } X \text{ in row } a\}$. Suppose to the contrary that for some X_0, X_1, \dots, X_d , there are no such z rows. Then there are at least $drk + 1$ values $a \in S$ such that $P_{X_i}(a) \subseteq P_{X_0}(a)$ for some i . Then there exists a fixed i' such that $P_{X_{i'}}(a) \subseteq P_{X_0}(a)$ for at least $rk + 1$ values $a \in S$. Thus for those $rk + 1$ values $a \in S$ and any $u \in X_{i'}$, $p_u(a) \in P_{X_0}(a)$, implying that there exists some $v \in X_0$ such that $p_u(a) = p_v(a)$ for at least $k + 1$ distinct $a \in S$. Thus $p_u = p_v$, showing $u = v \in X_0$. Hence $X_{i'} \subseteq X_0$, contradicting the assumption on $H_{\bar{r}}$. ■

Gao *et al.* (2006) converted the $A_{H_{\bar{r}}}(q, k, s)$ matrix to the binary matrix $B_{H_{\bar{r}}}(q, k, s)$ whose columns are labeled by \mathcal{N} . For any $a \in S$ and $F \in \{P_X(a) : X \in H_{\bar{r}}\}$, $B_{H_{\bar{r}}}(q, k, s)$ has a row labeled by $\langle a, F \rangle$ and has a 1-entry in cell $\langle a, F \rangle, v$ if $p_v(a) \in F$, and a 0-entry otherwise.

Lemma 4.3.2. *$B_{H_{\bar{r}}}(q, k, s)$ is $(H_{\bar{r}} : d; z)$ -disjunct.*

Proof. Consider any $d + 1$ edges X_0, X_1, \dots, X_d . Let a be a row in $A_{H_{\bar{r}}}(q, k, s)$ such that $P_{X_i}(a) \not\subseteq P_{X_0}(a)$ for $i = 1, \dots, d$. Then for any i , $p_{v_i}(a) \notin P_{X_0}(a)$ for some $v_i \in X_i$. Thus row $\langle a, P_{X_0}(a) \rangle$ of $B_{H_{\bar{r}}}(q, k, s)$ covers X_0 but not X_i for $i = 1, \dots, d$. Since $A_{H_{\bar{r}}}(q, k, s)$ has z such rows, the theorem follows. \blacksquare

Then properly choosing the parameters would imply

Theorem 4.3.3. *For any d, r, n and z , there exists an $(H_{\bar{r}} : d; z)$ -disjunct matrix $B_{H_{\bar{r}}}(q, k, s)$ with at most $q \cdot \binom{q+r-1}{r}$ rows, where*

$$q = z + (1 + o(1)) \frac{dr \lg n}{\lg(dr \lg n)}.$$

Moreover, for $n \geq 2^{\frac{16}{dr}}$, $q \leq z + \frac{2dr \lg n}{\lg(dr \lg n)}$.

Proof. For the existence of $A_{H_{\bar{r}}}(q, k, s)$, k and q should be chosen to satisfy $|\mathcal{N}| = n \leq q^{k+1}$ and $q \geq drk + z$. Then

$$\log_q n - 1 \leq k \leq \frac{q - z}{dr} \tag{4.3.1}$$

for the chosen k and q . There exists a positive integer k satisfies (4.3.1) if q satisfies $\log_q n \leq \frac{q-z}{dr}$. Therefore, it suffices to choose q satisfying

$$n^{dr} \leq q^{q-z}. \tag{4.3.2}$$

Let q_0 be the smallest number q satisfying (4.3.2). Then

$$q_0 \leq z + (1 + h(dr, n)) \frac{dr \lg n}{\lg(dr \lg n)}$$

where

$$h(l, n) = \frac{\lg \lg(l \lg n)}{\lg(l \lg n) - \lg \lg(l \lg n)} = o(1).$$

For $n \geq 2^{\frac{16}{dr}}$, $\lg(dr \lg n) \geq 4$, implying $(\lg(dr \lg n))^2 \leq 2^{\lg(dr \lg n)} = dr \lg n$. Hence $2 \lg \lg(dr \lg n) \leq \lg(dr \lg n)$, implying $h(dr, n) \leq 1$. Therefore,

$$q_0 \leq z + \frac{2dr \lg n}{\lg(dr \lg n)}$$

for $n \geq 2^{\frac{16}{dr}}$.

Next, $B_{H_{\bar{r}}}(q, k, t)$ has $\sum_{a \in S} |\{P_X(a) : X \in H_{\bar{r}}\}|$ rows. $|\{P_X(a) : X \in H_{\bar{r}}\}| \leq \sum_{i=1}^r \binom{q}{i} \leq \binom{q+r-1}{r}$. ■

Thus $B_{H_{\bar{r}}}(q, k, s)$ has at most $q^{\binom{q+r-1}{r}} \leq q^{r+1}$ rows for some s where q is as in Theorem 4.3.3. Hence,

Corollary 4.3.4. *When $z = O(1)$ and $n > 2^{\frac{16}{dr}}$,*

$$t(n, (H_{\bar{r}} : d; z)) = O\left(\left(\frac{2dr \lg n}{\lg(dr \lg n)}\right)^{r+1}\right).$$

Chen *et al.* (2007) [13] proposed another conversion to transform $A_{H_{\bar{r}}}(q, k, s)$ to an $(H_{\bar{r}} : d, z)$ -disjunct matrix. We generalize the conversion such that it is feasible not only for $A_{H_{\bar{r}}}(q, k, s)$ but also for any m -ary $(H_{\bar{r}} : d; z)$ -disjunct matrix.

Theorem 4.3.5. *If there exist a $t \times n$ m -ary $(H_{\bar{r}} : d; z)$ -disjunct matrix M and a $t' \times m$ $(d, r; z')$ -disjunct matrix M' , then there exists a $tt' \times n$ $(H_{\bar{r}} : d; zz')$ -disjunct matrix.*

Proof. The conversion is to label columns of M' by $0, \dots, q-1$ and replace each entry of $M = [M_{ji}]$ by a corresponding column of M' . Let M^* be the matrix obtained from the conversion. Consider $d+1$ edges X_0, X_1, \dots, X_d . In the matrix M , let l be a row in which $\{\text{entries of } X_0\} \not\supseteq \{\text{entries of } X_i\}$ for $i = 1, \dots, d$. Let $v_i \in X_i \setminus X_0$ such that $M_{lv_i} \notin \{\text{entries of } X_0\}$. Thus $\{M_{lv_i} : i = 1, \dots, d\} \cap \{\text{entries of } X_0\} = \emptyset$. Then after the conversion, there exist z' rows in M^* such that each row intersects columns corresponding to vertices of X_0 and none of columns corresponding to the v_i 's, i.e., $|\cap X_0 \setminus \bigcup_{i=1}^d \cap X_i| \geq z'$. Since in M there are at least z rows in each of which $\{\text{entries of } X_0\} \not\supseteq \{\text{entries of } X_i\}$ for $i = 1, \dots, d$, $|\cap X_0 \setminus \bigcup_{i=1}^d \cap X_i| \geq zz'$. ■

In particular, by Lemma 4.3.1 with $s = drk + z$, we have

Theorem 4.3.6. *If $n \leq q^{k+1}$ and $q \geq drk + z$, then $t(n, (H_{\bar{r}} : d; zz')) \leq (drk + z)t(q, (d, r; z'))$.*

The m -ary method is also employed in the constructions of $(d, r; z]$ -disjunct matrices where the m -ary matrices satisfy the following property.

Definition 2. An m -ary matrix $M = [M_{ji}]$ is $(d, r; z]$ -disjunct if for any two disjoint sets D and R of columns with $|D| = d$ and $|R| = r$, there exist at least z rows indexed j such that

$$\{m_{ji} : i \in D\} \cap \{m_{ji} : i \in R\} = \emptyset.$$

Let $t_m(n, (d, r; z])$ denote the minimum number of rows in an m -ary $(d, r; z]$ -disjunct matrix with n columns.

We relate this disjunctness property to the $(H : d; z)$ -disjunctness.

Lemma 4.3.7. $t_m(n, (H_{\bar{r}} : d; z)) \leq t_m(n, (d, r; z])$. *In particular, $t_m(n, (d, r; z]) = t_m(n, (H_r^* : d; z))$.*

Proof. Let M be an m -ary $(d, r; z]$ -disjunct matrix. Consider any $d + 1$ complexes $X_0, X_1, \dots, X_d \in H_{\bar{r}}$. Since no complex contains another, there exists $v_i \in X_i \setminus X_0$ for $i = 1 \cdots d$. Let D and R be two disjoint subsets of \mathcal{N} such that $|D| = d$, $|R| = r$, $\{v_1, \dots, v_d\} \subseteq D$ and $X_0 \subseteq R$. Then by the $(d, r; z]$ -disjunctness of M , there exist z rows in each of which $\{\text{entries of } D\} \cap \{\text{entries of } R\} = \emptyset$. Then in each of these z rows, entry of $v_i \notin \{\text{entries of } X_0\}$ and thus $\{\text{entries of } X_i\} \not\subseteq \{\text{entries of } X_0\}$ for $i = 1 \cdots d$. Hence, M is $(H_{\bar{r}} : d; z)$ -disjunct.

Next, suppose that M is an m -ary $(H_r^* : d; z)$ -disjunct matrix. Let D and R be any two disjoint sets of columns with $|D| = d$ and $|R| = r$. Suppose $D = \{v_1, \dots, v_d\}$. Let v be an element in R and X_i denote $(R \setminus \{v\}) \cup \{v_i\}$ for $i = 1 \cdots d$. Then by the $(H_r^* : d; z)$ -disjunctness of M , there exist z rows in each of which $\{\text{entries of } X_i\} \not\subseteq \{\text{entries of } R\}$ for $i = 1 \cdots d$, implying entry of $v_i \notin \{\text{entries of } R\}$ for $i = 1 \cdots d$. Thus in each of these z rows, $\{\text{entries of } D\} \cap \{\text{entries of } R\} = \emptyset$. ■

In fact, the complex set $H_{\bar{r}}$ in Gao *et al.*'s construction can be designated as $H_{\bar{r}}^*$. Then by Lemma 4.3.7 we have

Theorem 4.3.8. *When $z = O(1)$ and $n > 2^{\frac{16}{dr}}$,*

$$t(n, (d, r; z]) = O\left(\left(\frac{2dr \lg n}{\lg(dr \lg n)}\right)^{r+1}\right).$$

Stinson and Wei (2004) [49] proved the following result while the case $z = 1$ was proposed by D'yachkov *et al.* (2002) [27].

Lemma 4.3.9. *If there exist a $t \times n$ m -ary $(d, r; z]$ -disjunct matrix and a $t' \times m$ $(d, r; z']$ -disjunct matrix, then there exists a $tt' \times n$ $(d, r; zz']$ -disjunct matrix.*

Next, we shall consider the construction of m -ary $(d, r; z]$ -disjunct matrices. The incident matrices of some well-known m -ary codes potentially have a disjunct property. A *maximum-distance separable (MDS) code* with parameters (m, k, t) is an m -ary code of size m^k , length t and Hamming distance $t - k + 1$. Kautz and Singleton (1964) [35] first employed an MDS code to construct d -disjunct matrices. Sagalovich (1994) [45] observed the $(d, r; 1]$ -disjunctness property of its incident matrix.

Lemma 4.3.10. *If $t \geq dr(k - 1) + 1$ and $m^k \geq d + r$, then for any MDS code C with parameters (m, k, t) , the incident matrix of C is a $t \times m^k$ m -ary $(d, r; 1]$ -disjunct matrix.*

For any integer $k \geq 2$ and a prime power $q > k - 1$, there exists an MDS-code with parameters $(q, k, q + 1)$, which is a Reed-Solomon code. By the existence of such code, D'yachkov *et al.* (2002) [27] derived the following result.

Theorem 4.3.11. *If q is a prime power and $q \geq dr(k - 1) + 1$, then*

$$t(q^k, (d, r; z]) \leq (dr(k - 1) + 1)t(q, (d, r; z]).$$

Proof. The matrix Q obtained from removing $q - dr(k - 1)$ rows from the incident matrix of a $(q, k, q + 1)$ MDS code is the incident matrix of a $(q, k, dr(k - 1) + 1)$ MDS code. Thus by Lemma 4.3.10, Q is a $(dr(k - 1) + 1) \times q^k$ q -ary $(d, r; 1]$ -disjunct matrix. By the existence of such matrix and Theorem 4.3.9, the theorem follows. ■

An $(n, m, \{d, r\})$ - z -separating hash family is a set of functions \mathcal{F} , such that $|Y| = n, |X| = m, f : Y \rightarrow X$ for each $f \in \mathcal{F}$, and for any $D, R \subseteq Y$ such that $|D| = d, |R| = r$ and $D \cap R = \emptyset$, there exist at least z functions $f \in \mathcal{F}$ such that

$$\{f(y) : y \in D\} \cap \{f(y) : y \in R\} = \emptyset.$$

An (n, m, w) - z -perfect hash family is a stronger family of functions where for any $|W| = w$, there exist at least z functions such that for any $y \neq y' \in W$,

$$f(y) \neq f(y').$$

Then it is obvious that an $(n, m, d + r)$ - z -perfect hash family is an $(n, m, \{d, r\})$ - z -separating hash family which is equivalent to an m -ary $(d, r; z]$ -disjunct matrix with n columns.

Stinson and Wei (2004) [49] observed the following result from a result on separating hash family (Stinson *et al.* 2000 [50]).

Lemma 4.3.12. *For any positive integers m, d and r , there exists an infinite class of $t \times n$ m -ary $(d, r; 1]$ -disjunct matrices where $t = O((dr)^{\lg^*(n)} \lg n)$.*

Note that the function \lg^* is defined by $\lg^*(n) = \lg^*(\lceil \lg n \rceil) + 1$ for $n > 1$ and $\lg^*(1) = 1$. In fact, it grows very slowly; for example, $\lg^*(n) \leq 6$ for $n \leq 2^{65536}$.

Further, they used a result on perfect hash family with $z = 1$ (Wang and Xing, 2001 [53]) to obtain

Lemma 4.3.13. *For any positive integers $m \geq d + r$, there exists an explicit construction for an infinite class of $t \times n$ m -ary $(d, r; 1]$ -disjunct matrices with $t = O(c(m) \log n)$ for some function c of m .*

Then plugging the m -ary matrix in Lemma 4.3.13 and the matrix in Corollary 4.2.7 into Theorem 4.3.9 with $m = d + r$ implies

Theorem 4.3.14. $t(n, (d, r; z]) \leq O(z \binom{d+r}{r} c(d+r) \log n)$ for some function c of $d + r$.

4.4 Constructing by Controlling Row-covering

Chen *et al.* (2008) [14] provided an upper bound by another approach. A z -cover of a hypergraph $G = (V, \mathcal{F})$ is a multi-subset $\mathcal{C} \subseteq V$ of vertices such that $|\mathcal{C} \cap F| \geq z$ for every edge $F \in \mathcal{F}$. Let $t_z(G)$ denote the minimum size among all z -covers of G . Since a z -cover can be obtained by copying a 1-cover z times,

$$t_z(G) \leq z t_1(G).$$

Let G^* be the hypergraph with vertex set $\binom{[n]}{w}$ and edge set $F^* = \{E_{D,R} : D \cap R = \emptyset, |D| = d, |R| = r\}$ where $E_{D,R} = \{S \in \binom{[n]}{w} : R \subseteq S \text{ and } D \cap S = \emptyset\}$. Let $M_{G^* \mathcal{C}}$ be the matrix with rows indexed by a z -cover \mathcal{C} of G^* and columns indexed by $[n]$, and the matrix has a 1-entry in cell (W, a) if $a \in W$, and a 0-entry otherwise. Chen *et al.* (2008) [14] observed that $M_{G^* \mathcal{C}}$ is $(d, r; z]$ -disjunct. To obtain an upper bound of $t_1(G^*)$, they quoted a lemma of Lovász (1975) [39] on hypergraph. For a hypergraph $G = (V, \mathcal{F})$, greedily choosing vertices sequentially such that every chosen vertex belongs to the maximum number of edges which are not covered yet yields a 1-cover of G of size less than

$$\frac{|V|}{\min_{F \in \mathcal{F}} |F|} (1 + \ln \Delta)$$

where Δ is the maximum degree of a vertex, thus implying an upper bound of $t_1(G)$. Therefore,

Theorem 4.4.1. *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*

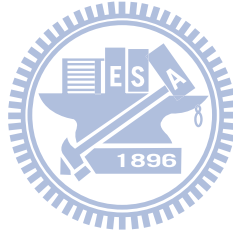
$$t < z \frac{\binom{n}{r} \binom{n-r}{d}}{\binom{w}{r} \binom{n-w}{d}} \left[1 + \ln \binom{w}{r} \binom{n-w}{d} \right]. \quad (4.4.1)$$

Proof. $G^* = (\binom{[n]}{w}, \mathcal{F}^*)$ is uniform and regular. Thus $\frac{|\binom{[n]}{w}|}{\min_{F \in \mathcal{F}^*} |F|} = \frac{|\mathcal{F}^*|}{\Delta}$ where $|\mathcal{F}^*| = \binom{n}{r} \binom{n-r}{d}$ and $\Delta = \binom{w}{r} \binom{n-w}{d}$. ■

By properly choosing w to minimize (4.4.1), they proved

Theorem 4.4.2. *For any positive integers d, r, z and n with $d + r \leq n$,*

$$t(n, (d, r; z]) < z \left(\frac{d+r}{r}\right)^r \left(\frac{d+r}{d}\right)^d [1 + (d+r)(1 + \ln(\frac{n}{d+r} + 1))].$$



Chapter 5

Reconstruction of Hidden Graphs

As introduced previously, in the graph reconstruction problem, a hidden graph G is known belonging to a given family \mathcal{G} of labeled graphs on the set $\mathcal{N} = [n]$, and the main task is to reconstruct G by asking queries as few as possible, where a query is of the form,

“Does S induce at least one edge of G ?”

for $S \subseteq \mathcal{N}$. This query is denoted by $Q(S)$ and $Q(S) = 1$, representing “yes”, or 0, representing “no”.

\mathcal{G} usually provides some information to the setting of queries. In this chapter, we study the graph reconstruction problem where the structure of the hidden graph is known.

Notations. Subsequently, for a graph G , $G[S]$ denotes the induced subgraph of graph G with vertex set S .

5.1 Preparation and Subroutines

A simple graph is a graph where each edge contains exactly two vertices and a vertex v is said to be adjacent to u if they induce an edge. We focus

our attention on the reconstructions of four class of hidden simple graphs of known structure: Hamiltonian cycle, matchings, stars, and cliques. A *Hamiltonian cycle* on \mathcal{N} is a cycle passing through every vertex in \mathcal{N} exactly once and thus \mathcal{G} could be the set of all

$$\frac{(n-1)!}{2}$$

Hamiltonian cycles on \mathcal{N} and of course, the hidden graph G is one of them. A *matching* on \mathcal{N} is a set of disjoint edges while a *perfect matching* is a matching where every vertex in \mathcal{N} belongs to (is incident to) one edge. Thus the number of perfect matchings on \mathcal{N} is

$$\frac{n!}{2^{\frac{n}{2}}(n/2)!}$$

A *star* is a graph where all its edges have a common vertex called center. A star of k edges can be defined by choosing a vertex as the center and other k vertices that are adjacent to the center. Therefore, the number of stars on \mathcal{N} is upper bounded by

$$\sum_{k=2}^{n-1} n \binom{n-1}{k} + \frac{n(n-1)}{2} + 1 = n(2^{n-1} - 1) - \frac{n(n-1)}{2} + 1.$$

A *clique* on \mathcal{N} is of the form $\binom{S}{2}$ for some $S \subseteq \mathcal{N}$ of size at least two and there are $2^n - n - 1$ different cliques on \mathcal{N} .

In the following, we will introduce some useful tools and algorithms that will be used as subroutines in the main algorithms.

An affine plane of order p is a balanced incomplete block design with p^2 points and $p^2 + p$ blocks of size p such that each pair of points appear together in exactly one block. It is well-known that an affine plane of order p exists whenever p is a prime power (see Anderson, 1990 [3]).

The affine plane method was first proposed in (Tettelin *et al.* [51], 1996; Grebinski and Kucherov, 1998 [30]) and then employed by Hwang and Lin (2003) [32] which is to take an affine plane with the point set containing \mathcal{N}

and then reconstruct each subgraph induced by a block. The advantages of using this method are that each block has size p which could be much smaller than n and that all graphs induced by blocks can be dealt simultaneously. The problem is how small a prime power p such that $p^2 \geq n$ could be. Nagura (1952) [43] proved that for $n > 24$, there is always a prime between n and $1.2n$. Hence,

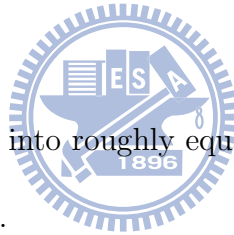
Lemma 5.1.1. *For $n > 24^2$, there exists a prime p such that $n \leq p^2 \leq 1.44n < 2n$.*

Angluin and Chen (2006) [4] gave an elegant algorithm (see Algorithm 1) to find a vertex contained in at least one edge of a hidden graph on n vertices using at most $\lg n$ queries.

Algorithm 1 FIND-ONE-VERTEX

```

1:  $S \leftarrow \mathcal{N}$ 
2: if  $Q(S) = 0$  then
3:   Return  $\emptyset$ .
4: end if
5:  $A \leftarrow \mathcal{N}$ .
6: while  $|A| > 1$  do
7:   Arbitrarily partition  $A$  into roughly equal-sized  $A_0$  and  $A_1$ .
8:   if  $Q(S \setminus A_0) = 1$  then
9:      $S \leftarrow S \setminus A_0, A \leftarrow A_1$ .
10:  else
11:     $A \leftarrow A_0$ .
12:  end if
13: end while
14: Return the element in  $A$ .
```



Notice that the algorithm preserves the invariance that $Q(S) = 1$ and $Q(S \setminus A) = 0$ if the input hidden graph contains at least one edge. This shows that A contains a vertex on an edge of the hidden graph; indeed, $|A|$ is monotonically decreasing and the halving of A 's cardinality in each iteration results in $\lg n$ queries. Furthermore, once the algorithm terminates,

$A = \{v\}$ for some vertex v and $S \setminus \{v\}$ contains a vertex adjacent to v ; hence, a neighbor of v can be found by using binary splitting algorithm on $S \setminus \{v\}$ with v added to each test. Therefore, reconstructing an edge can be accomplished in $2 \lg n$ queries.

A matching is maximal if it is not contained in a matching of larger size. A general approach that we propose to reconstruct a hidden graph is to find a maximal matching of the hidden graph at the beginning of the progress of reconstructing the whole graph (Chang *et al.*, 2010 [11]). The advantage of this approach is that a reconstructed maximal matching of a hidden graph would reveal a partial structure of the hidden graph, thus providing a direction to complete the reconstruction of the remaining graph.

Algorithm 2 FIND-MAXIMAL-MATCHING

```

1:  $M \leftarrow \emptyset, S \leftarrow \mathcal{N}, U \leftarrow \emptyset, U' \leftarrow \emptyset.$ 
2: while  $Q(S) = 1$  do
3:   Reconstruct an edge in  $G[S]$ , say  $\{u, f(u)\}$ .  $M \leftarrow M \cup \{\{u, f(u)\}\},$ 
      $S \leftarrow S \setminus \{u, f(u)\}, U \leftarrow U \cup \{u\}, U' \leftarrow U' \cup \{f(u)\}.$ 
4: end while
5: Return  $(M, U, U', f).$ 

```

Algorithm 2 reconstructs edges one by one. The two vertices in an edge are removed from S as soon as it is reconstructed and thus the reconstructed edges share no vertex, implying the returned set M is a matching. Indeed, M is a maximal matching because searching an edge induced by S continues until it induces no edge which implies that no larger matching contains M . Overall, Algorithm 2 reconstructs a maximal matching of the hidden graph in $2m' \lg n + 1$ queries, where m' is the size of the maximal matching. In addition, the algorithm returns two sets U and U' to collect the vertices in the reconstructed edges and also returns a function f that pairs the vertices between U and U' to record the edges in M . We call $U \cup U'$ the *saturating set* of M for U, U' and M returned by the algorithm.

A nontrivial path is a path containing at least one edge. We provide an

algorithm (see Algorithm 3) to reconstruct any hidden graph on n vertices that contains only nontrivial paths in $2m \lg n + m + 5$ queries where m is the number of edges in the hidden graph (Chang *et al.*, 2010 [11]).

Algorithm 3 FIND-ALL-PATHS

Let G be a hidden graph on a set \mathcal{N} of n vertices and contain only nontrivial paths.

- 1: $E \leftarrow \emptyset$.
 - 2: Apply FIND-MAXIMAL-MATCHING on G . Assume (M, U, U', f) is returned.
 - 3: $E \leftarrow E \cup M$, $I \leftarrow \mathcal{N} \setminus (U \cup U')$.
 - 4: Apply FIND-MAXIMAL-MATCHING on $G[U]$ and $G[U']$. Assume (M_1, A, A', f_1) and (M_2, B, B', f_2) are returned, respectively.
 - 5: $E \leftarrow E \cup M_1 \cup M_2$, $I_1 \leftarrow U \setminus (A \cup A')$, $I_2 \leftarrow U' \setminus (B \cup B')$.
 - 6: **for** $u \in I_1$ **do**
 - 7: Apply a binary splitting algorithm on $I_2 \setminus \{f(u)\}$ with u added to each test. Assume v (if any) is obtained from the search.
 $E \leftarrow E \cup \{\{u, v\}\}$, $I_1 \leftarrow I_1 \setminus \{u\}$, $I_2 \leftarrow I_2 \setminus \{v\}$.
 - 8: **end for**
 - 9: **while** $Q(I_1 \cup I) = 1$ **do**
 - 10: Reconstruct an edge in $G[I_1 \cup I]$, say $\{u, i\}$ where $u \in I_1$ and $i \in I$.
 $E \leftarrow E \cup \{\{u, i\}\}$, $I_1 \leftarrow I_1 \setminus \{u\}$.
 - 11: **end while**
 - 12: Reconstruct edges between I_2 and I by the same way as lines 9-11.
 - 13: Return E .
-

Figure 5.1 demonstrates an example of Algorithm 3: (a) The bold edges form a maximal matching and an independent set I is produced. (b) Line 4 reconstructs edges in $G[U]$ and $G[U']$. Then finally two independent sets $I_1 = \{b, i, k\}$ and $I_2 = \{a, c, j\}$ are obtained (line 5). (c) Lines 6-8 reconstruct edges between I_1 and I_2 . By applying a binary splitting algorithm to $I_2 \setminus \{f(b)\} = \{c, j\}$ with b added to each test, edge $\{b, c\}$ is reconstructed. Finally, $I_1 = \{i, k\}$ and $I_2 = \{a, j\}$. (d) Lines 9-12 reconstruct edges between I and $I_1 \cup I_2$.

Lemma 5.1.2. *Algorithm 3 reconstructs any graph G on n vertices containing only nontrivial paths in $2m \lg n + m + 5$ queries where m is the number*

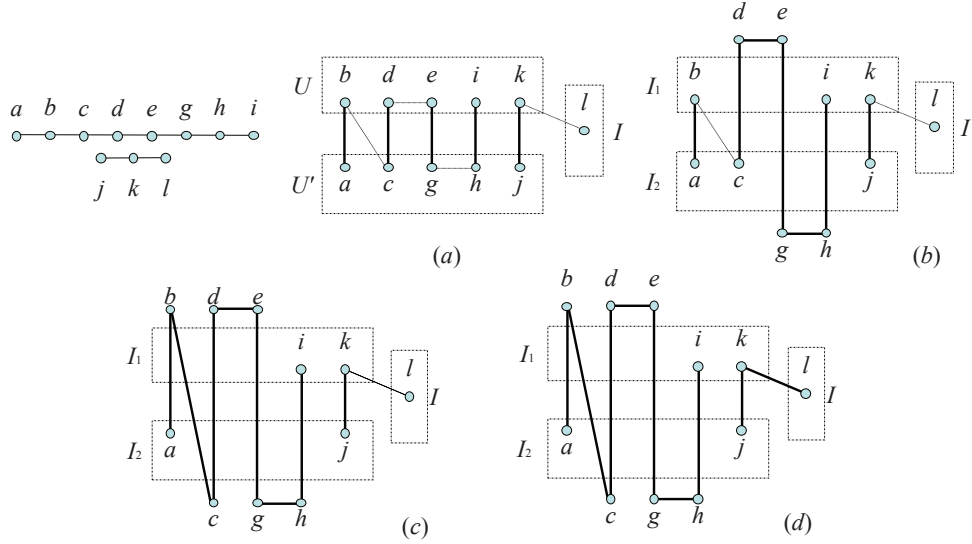


Figure 5.1: An example of FIND-ALL-PATHS algorithm

of edges of G .

Proof. The algorithm first starts at reconstructing a maximal matching M of G whose saturating set is assumed to be $U \cup U'$. Since G only contains nontrivial paths, the structure of the remaining graph consists of two matchings $E[U]$ and $E[U']$ and some edges between U and U' and $I = \mathcal{N} \setminus (U \cup U')$ which is an independent set since M is maximal. Next, FIND-MAXIMAL-MATCHING would reconstruct the matching M_1 induced by U and M_2 induced by U' whose saturating sets are $A \cup A'$ and $B \cup B'$, respectively. Then the incident edges of all vertices in $A \cup A' \cup B \cup B'$ are reconstructed so it remains to reconstruct edges between three independent sets I , $I_1 = U \setminus (A \cup A')$, and $I_2 = U' \setminus (B \cup B')$. Note that constructing those three matchings takes

$$2(|M| + |M_1| + |M_2|) \lg n + 3$$

queries.

Next, an edge between I_1 and I_2 is not reconstructed only if it is not in M and every vertex in I_1 is adjacent to at most one vertex in I_2 . Therefore, line

7 exactly accomplishes the reconstruction of edges between I_1 and I_2 that are not in M . Since $|I_1| \leq |M|$ and the splitting algorithm takes at most $\lg n$ queries, there are at most

$$m_1 \lg n + |M|$$

queries spent in this portion where m_1 is the number of edges reconstructed here.

Finally, it remains to reconstruct hidden edges between I and $I_1 \cup I_2$. For the edges between I and I_1 , as shown in lines 9-11, the algorithm recursively reconstructs an edge in $G[I_1 \cup I]$, say $\{u, i\}$ where $u \in I_1$ and $i \in I$ and removes u from I_1 until $I \cup I_1$ induces no edge. Note here that u can be removed because both its incident edges are reconstructed after the reconstruction of $\{u, i\}$ and indeed removing u is to make sure that edges in $I \cup I_1$ are unreconstructed before each iteration. Similarly, the edges between I and I_2 can be reconstructed by the same way. Note that the number of queries spent here is at most

$$2m_2 \lg n + 2$$

where m_2 is the number of edges between I and $I_1 \cup I_2$.

It is easily observed that each edge is reconstructed once and hence the overall cost of this algorithm is upper bounded by $2m \lg n + m + 5$. Therefore, the lemma follows. ■

5.2 Reconstructions of Simple Graphs

Assume that \mathcal{G} consists of all Hamiltonian cycles on \mathcal{N} . Since there are $\frac{(n-1)!}{2}$ of them, the theoretic information lower bound is $\lg \frac{(n-1)!}{2} \leq n \lg n$. Grebinski and Kucherov (1998) [30] gave a sequential algorithm to reconstruct a Hamiltonian cycle with $2n \lg n$ queries. Chang *et al.* (2010) [11] improved their result to $(1 + o(1))(n \lg n)$ by employing the affine plane method together with the algorithm FIND-ALL-PATHS.

Theorem 5.2.1. *For any hidden Hamiltonian cycle G of order $n > 24^2$, G can be reconstructed in $n \lg n + 15n$ queries.*

Proof. By Lemma 5.1.1, there is a prime p such that $n \leq p^2 \leq 2n$. Add $p^2 - n$ dummy vertices to obtain an affine plane and take them away when testing the blocks. A block is said to be positive if its testing result is positive. It is obvious that each positive block induces a graph containing only nontrivial paths; hence, a Hamiltonian cycle can be reconstructed by applying FIND-ALL-PATHS to these blocks (see an example in Table 5.1). Since there are $p^2 + p$ blocks and every edge appears exactly in a block, there are totally at most $(p^2 + p) + 2m \lg p + m + 5(p^2 + p)$ queries where $m = n$. Hence, a Hamiltonian cycle can be reconstructed in $12n + 6\sqrt{2n} + n \lg 2n + n < n \lg n + 15n$ queries for $n > 24^2$. ■

Next, we consider that \mathcal{G} is the set of all matchings on \mathcal{N} . The reconstruction of matchings has been studied in (Alon and Asodi, 2005; Bouvel *et al.*, 2005). The number of perfect matchings on n (even) vertices is $\frac{n!}{2^{\frac{n}{2}}(n/2)!}$,

providing an information lower bound $\lg \frac{n!}{2^{\frac{n}{2}}(n/2)!} = (1 + o(1))(\frac{n}{2} \lg n)$ on the reconstruction of matchings. Bouvel *et al.* (2005) [8] gave sequential algorithms to reconstruct a matching of unknown size and a perfect matching on n vertices in $(1 + o(1))(n \lg n)$ and $(1 + o(1))(\frac{n}{2} \lg n)$ queries, respectively. Recently, Chang *et al.* (2010) [11] took advantage of the affine plane method to reconstruct a matching of unknown size in at most $(1 + o(1))(\frac{n}{2} \lg n)$ queries.

Theorem 5.2.2. *For $n > 24^2$, reconstructing a matching on n vertices can be done in $m \lg n + 4n$ queries, where $m \leq \frac{n}{2}$ is the number of edges of the matching.*

Proof. Similar to the proof of Theorem 5.2.1, the affine plane method produces $p^2 + p$ blocks such that each pair of vertices belongs to exactly one of them where $n \leq p^2 \leq 2n$. Since each block induces a graph containing just a matching, FIND-MAXIMAL-MATCHING would reconstruct each graph induced by a positive block (see an example in Table 5.1). Hence, overall

process takes at most $(p^2 + p) + 2m \lg p < 2n + \sqrt{2n} + m \lg 2n$ queries to reconstruct a matching on n vertices. ■

Example 4. Examples of small order illustrating Theorem 5.2.1 and Theorem 5.2.2 are given in the following. Let $\mathcal{N} = [7]$. Then $p = 3$ is the smallest prime power such that its square is at least 7. $\{ \{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{1, 4, 7\}, \{2, 5, 8\}, \{3, 6, 9\}, \{1, 5, 9\}, \{2, 6, 7\}, \{3, 4, 8\}, \{3, 5, 7\}, \{1, 6, 8\}, \{2, 4, 9\} \}$ is an affine plane of order 3. In Table 5.1, the hidden graph G_1 is a Hamiltonian cycle and the hidden graph G_2 is a matching. For G_1 , $\{1, 2, 3\}, \{4, 5, 6\}, \{1, 4, 7\}, \{2, 6, 7\}$ and $\{3, 4, 8\}$ are positive blocks and then FIND-ALL-PATHS is applied to each of them. For G_2 , $\{1, 2, 3\}, \{4, 5, 6\}$ and $\{3, 5, 7\}$ are positive blocks and then FIND-MAXIMAL-MATCHING is applied to each of them (see the corresponding cell in Table 5.1). Note that a cell in Table 5.1 is empty means the corresponding block is not positive, i.e., the graph induced by it contains no edge.

Notice that the dummy vertices 8 and 9 are removed when the blocks are tested. Based on the property of affine plane, the edge set of the hidden graph is decomposed into the edge sets of graphs induced by positive blocks, and therefore the whole graph is reconstructed by collecting edges induced by positive blocks.

Next, we consider that \mathcal{G} is the set of all stars on \mathcal{N} . Thus $|\mathcal{G}| = n(2^{n-1} - 1) - \frac{n(n-1)}{2} + 1$. Accordingly, the information lower bound is $(1 + o(1))n$ which is the number of queries required to reconstruct a hidden star. Bouvel *et al.* (2005) [8] gave a sequential algorithm using queries achieving the lower bound $\Omega(n)$. In fact, their algorithm requires $2n$ queries in the worst case. Chang *et al.* (2010) [11] proved that the lower bound $(1 + o(1))n$ can be achieved by a sequential algorithm.

Theorem 5.2.3. *A star on n vertices can be reconstructed in $n + 2 \lg n$ queries.*

Proof. The first step is to find the center of the star, and then to find all its neighbors by querying each vertex with the center. An edge of the star

can be reconstructed in $2 \lg n$ queries and one of the two vertices in the edge must be the center. The center can be determined by simply testing one of these two vertices together with all other vertices. Clearly, the whole process takes at most $2 \lg n + n$ queries. ■

Finally, suppose that \mathcal{G} is the set of all cliques on \mathcal{N} . Then the information lower bound is $\lg 2^n = n$. Bouvel *et al.* (2005) [8] provided a sequential algorithm to reconstruct a hidden clique in $2n$ queries. Chang *et al.* (2010) [11] slightly improved their result by giving an algorithm to construct a clique in $n + \lg n$ queries.

Theorem 5.2.4. *A clique on n vertices can be reconstructed in $n + \lg n$ queries.*

Proof. A vertex v on the clique can be found in $\lg n$ queries by applying FIND-ONE-VERTEX. Then the clique can be reconstructed by querying each vertex with x . Hence the whole process takes at most $n + \lg n$ queries. ■



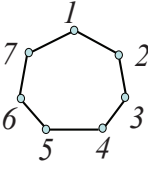
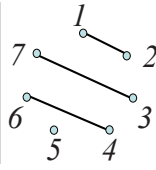
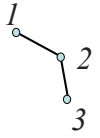
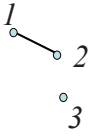
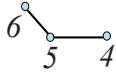

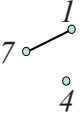
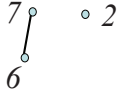

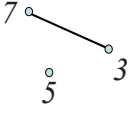
Blocks	G_1 	G_2 
{1, 2, 3}		
{4, 5, 6}		
{1, 4, 7}		
{2, 6, 7}		
{3, 4, 8}		
{3, 5, 7}		
others		

Table 5.1: Examples of small order for Theorem 5.2.1 and Theorem 5.2.2

Chapter 6

Conclusion and Remarks

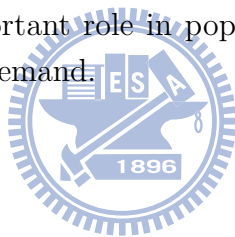
Research in this thesis can be cataloged into four categories: identification problems in k -inhibitor complex model and general inhibitor complex model, classification problems in 1-inhibitor complex model and k -inhibitor clone model, lower bounds and constructions of related disjunct matrices, and graph reconstruction problems on hidden graphs of known structure.

In the study of group testing, we introduced a new pooling design environment by allowing the coexistence of inhibitors and complexes which, separately, have been well studied in the literature. For identification problems, we give a nonadaptive pooling design, with error-tolerance ability, to the most general model in such an environment with no need to know the exact relation between inhibitors and positive complexes. We present a novel concept “inclusiveness” which leads to a significant improvement on the decoding procedure. Indeed, identifying all positive complexes can be done by comparing the values of complexes plugging into certain cutoff functions after the testing outcomes are produced.

On the other hand, in the k -inhibitor model, instead of treating the inhibitors as annoying elements, we face them as substances with certain features and attempt to identify them. We prove that all complexes under 1-inhibitor complex model with error tolerance e can be identified by using $O(ec(d, h, r) \log n)$ tests nonadaptively and $O(ec(d, h, r)hr \log n|H|)$ decoding time for some function c on d, h , and r , where a $(d + h, 2r; z]$ -disjunct

matrix is sufficient; no complex nonadaptive design is required. This is also a notable solution for identification problem under the 1-inhibitor model. Unlike other identification results, in this design, the strategy of identifying all inhibitors first is put into execution, leading to a great improvement on decoding complexity. Furthermore, this design indeed comes from merging a design for identifying inhibitors in the inhibitor model and a design for identifying positive items in a non-inhibitor model, suggesting a way to strengthen a design.

The problems we consider in this thesis all originated from applications that were observed in recent literatures and our results suggest an efficient nonadaptive strategy so that the time required to perform experiments and analyze outcomes can be substantially reduced. We believe that the new properties that we propose in this study can be applied to other practical models with decent testing performance and decoding procedure. We also call attention to the study of classification problem. This problem is not only of theoretical interest but significant in applications. We believe the inhibitory substances can play an important role in population and the setting-up of inhibitor libraries is also in demand.



References

- [1] N. Alon and V. Asodi, Learning a hidden subgraph, *SIAM J. Discrete Math.* 18 (2005) 697-712.
- [2] N. Alon, R. Beigel, S. Kasif, S. Rudich, and B. Sudakov, Learning a hidden matching, *SIAM J. Comput.* 33 (2004) 487-501.
- [3] I. Anderson, Combinatorial Designs: Construction Methods, *Ellis Horwood* (1990).
- [4] D. Angluin and J. Chen, Learning a hidden hypergraph, *J. Mach. Learn. Res.* 7 (2006) 2215-2236.
- [5] D. Angluin and J. Chen, Learning a hidden graph using $O(\log n)$ queries per edge, *J. Compu. Sys. Sci.* 74 (2008) 546-556.
- [6] M. Aigner, Combinatorial Search, *John Wiley and Sons* (1988).
- [7] 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, 22-30.
- [8] M. Bouvel, V. Grebinski, and G. Kucherov, Combinatorial search on graphs motivated by bioinformatics applications: a brief survey, in: Graph-Theoretic Concepts in Computer Science (WG), *Lec. Notes Comput. Sci.* 3787 (2005) 16-27.

- [9] F. H. Chang, H. Chang, and F. K. Hwang, Pooling designs for clone library screening in the inhibitor complex model, *J. Comb. Optim.* (2010) DOI 10.1007/s10878-009-9279-9.
- [10] H. Chang, H. B. Chen, and H. L. Fu, Identification and classification problems on pooling designs for inhibitor model, *J. Comput. Biol.* (2010) to appear.
- [11] H. Chang, H. B. Chen, H. L. Fu, and C. H. Shi, Reconstruction of hidden graphs and threshold group testing, *J. Comb. Optim.* (2010) DOI 10.1007/s10878-010-9291-0.
- [12] H. B. Chen, Combinatorial nonadaptive group testing with biological applications, *Ph.D. Thesis, National Chiao Tung University* (2006).
- [13] 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. Comb. Optim.* 14 (2007) 121-129.
- [14] 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, *Optim. Lett.* 2 (2008) 425-431.
- [15] A. De Bonis, New combinatorial structures with applications to efficient group testing with inhibitors, *J. Comb. Optim.* 15 (2008) 77-94.
- [16] A. De Bonis, L. Gasieniec, and U. Vaccaro, Optimal two-stage algorithms for group testing problems, *SIAM J. Comput.* 34 (2005) 1253-1270.
- [17] A. De Bonis and U. Vaccaro, Improved algorithms for group testing with inhibitors, *Inform. Process. Lett.* 67 (1998) 57-64.

- [18] A. De Bonis and U. Vaccaro, Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels, *Theoret. Comput. Sci.* 306 (2003) 223-243.
- [19] R. Dorfman, The detection of defective members of large populations, *Ann. Math. Statist.* 14 (1943) 436-440.
- [20] D. Z. Du and F. K. Hwang, Combinatorial group testing and its applications, *World Scientific*, Singapore (1993).
- [21] D. Z. Du and F. K. Hwang, Combinatorial group testing and its applications (2nd Edition), *World Scientific* Singapore (2000).
- [22] D. Z. Du and F. K. Hwang, Identifying d positive clones in the presence of inhibitors, *Int. J. Bioinformatics Research and Applications* 1 (2005) 162-168.
- [23] D. Z. Du and F. K. Hwang, Pooling Designs and Nonadaptive Group Testing - Important Tools for DNA Sequencing, *World Scientific* (2006).
- [24] D. Z. Du, F. K. Hwang, W. Wu, and T. Znati, New construction for transversal design, *J. Comput. Biol.* 13 (2006) 990-995.
- [25] A. G. D'yachkov, A. J. Macula, D. C. Torney, and P. A. Vilenkin, Two models of nonadaptive group testing for designing screening experiments, 63-75. In Attkinson, A.C., Hackl, P., and Muller, W.G., eds., *Proc. 6th Inter. Workshop in Model Oriented Design and Analysis* Physica-Verlog (2001).
- [26] A. G. D'yachkov and V. V. Rykov, A survey of superimposed code theory, *Probl. Control Inform. Theory* 12 (1983) 229-242.
- [27] 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 A* 99 (2002) 195-218.

- [28] M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan, Group testing with sequences in experimental molecular biology, *Proc. Compression and Complexity of Sequences* (1997) 357-367.
- [29] H. Gao, F. K. Hwang, M. Thai, W. Wu, and T. Znati, Construction of $d(H)$ -disjunct matrix for group testing in hypergraphs, *J. Comb. Optim.* 12 (2006) 297-301.
- [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. Comput. Biol.* 10 (2003) 231-236.
- [33] F. K. Hwang and F. H. Chang, The identification of positive clones in a general inhibitor model, *J. Comput. System Sci.* 73 (2007) 1090-1094.
- [34] F. K. Hwang and V. T. Sós, Nonadaptive hypergeometric group testing, *Studia Sci. Math. Hungar.* 22 (1987) 257-263.
- [35] W.H. Kautz and R.C. Singleton, Nonrandom binary superimposed codes, *IEEE Trans. Inform. Theory* 10 (1964) 363-377.
- [36] M. Lappe and L. Holm, Unraveling protein interaction networks with near-optimal efficiency, *Nature Biotechnology* 22 (2003) 98-103.
- [37] C. H. Li, A sequential method for screening experimental variable, *J. Amer. Statist. Assoc.* 57 (1962) 455-477.
- [38] Y. Li, M. Thai, Z. Liu, and W. Wu, Protein-to-protein interactions and group testing in bipartite graphs, *International J. of Bioinformatics Research and Applications* 1 (2005) 414-419.

- [39] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Math.* 13 (1975) 383-390.
- [40] C. J. Mitchell and F. C. Piper, Key storage in secure networks, *Discrete Appl. Math.* 21 (1988) 215-228.
- [41] A. J. Macula, V. V. Rykov, and S. Yekhanin, Trivial two-stage group testing for complexes using almost disjoint matrices, *Discrete Appl. Math.* 137 (2004) 97-107.
- [42] A. J. Macula, D. C. Torney, and P. A. Villenkin, Two-stage group testing for complexes in the presence of errors, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.* 55 (2000) 145-157.
- [43] J. Nagura, On the interval containing at least one prime number, *Proceedings of the Japan Academy, Series A* 28 (1952) 177-181.
- [44] R. M. Phatarfod and A. Sudbury, The use of a square array scheme in blood testing, *Stat. Med.* 13 (1994) 2337-2343.
- [45] Y. L. Sagalovich, On separating systems, *Problemy Peredachi Informat-sii* 30 (1994) 14-35 (in Russian).
- [46] 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.
- [47] A. Sorokin, A. Lapidus, V. Capuano, N. Galleron, P. Pujic, and S. D. Ehrlich, A new approach using multiplex long accurate PCR and yeast artificial chromosomes for bacterial chromosome mapping and sequencing, *Genome Res.* 6 (1996) 448-453.
- [48] D. R. Stinson, On some methods for unconditionally secure key distribution and broadcast encryption, *Design Code Cryptogr.* 12 (1997) 215-343.

- [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 A* 90 (2000) 224-234.
- [51] H. Tettelin, D. Radune, S. Kasif, H. Khouri, and S. L. Salzberg, Optimized multiplex PCR: Efficiently closing a whole-genome shotgun sequencing project, *Genomics* 62 (1996) 500-507.
- [52] D. C. Torney, Sets pooling designs, *Ann. Comb.* 3 (1999) 95-101.
- [53] H. Wang and C. Xing, Explicit constructions of perfect hash families from algebraic curves over finite fields, *J. Combin. Theory A* 93 (2001) 112-124.
- [54] M. Xie, K. Tatsuoka, J. Sacks, and S. Young, Group testing with blockers and synergism, *J. Amer. Statist. Assoc.* 96 (2001) 92-102.

