J Comb Optim (2008) 15: 49–59
DOI 10.1007/s10878-007-9083-3

# A survey on nonadaptive group testing algorithms through the angle of decoding

**Hong-Bin Chen · Frank K. Hwang**

Published online: 29 June 2007


**Abstract** Group testing, sometimes called pooling design, has been applied to a variety of problems such as blood testing, multiple access communication, coding theory, among others. Recently, screening experiments in molecular biology has become the most important application. In this paper, we review several models in this application by focusing on decoding, namely, giving a comparative study of how the problem is solved in each of these models.

**Keywords** Group testing · Pooling designs · Nonadaptive algorithms

## 1 Introduction

In the classic group testing problem, we consider a set $N$ of $n$ items consisting of at most $d$ positive items with the other being negative items. Typically $d$ is much smaller than $n$. A group test, sometimes called a pool, can be applied to an arbitrary subset $S$ of items with two possible outcomes; a negative outcome implies all items in $S$ are negative, while a positive outcome implies otherwise, i.e., there exists at least one positive item in $S$, not knowing which or how many. Let $P$ denote the set of all positive items. The problem is to identify all items in $P$ with a small number of tests.

Group testing is a basic tool which can be applied to a variety of problems such as blood testing, multiple access communication, coding theory, among others (Du and Hwang 2000). Recently group testing procedures have been applied to computational molecular biology, for example, in screening clone library. The problem of clone library screening is to determine which *clones* (a DNA segment) in the clone library

H.-B. Chen (✉) · F.K. Hwang
Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan
e-mail: andan.am92g@nctu.edu.tw

F.K. Hwang
e-mail: fhwang@math.nctu.edu.tw

hybridize with a given probe in an efficient fashion. A clone is said to be positive if it hybridizes with the probe, and negative otherwise.

In some applications, beside positive and negative clones, there is a third category of clones called *inhibitors* whose effect is to neutralize positive clones. That means the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive clones in the pool. The inhibitor model was first introduced by Farach et al. (1997), and studied further in (De Bonis et al. 2005; De Bonis and Vaccaro 1998; Chang et al. 2007; Hwang and Liu 2003).

In some other applications the property of being positive or negative is defined on subsets of items, instead of on individual items. Such a model is usually referred to as the *complex* model, first introduced by Torney (1999). The complex group testing model is related to other problems such as graph testing, secure key distribution, among others (Alon et al. 2004; Chen et al. 2007; Macula and Popyack 2004; D'yachkov et al. 2002; Torney 1999).

Group testing algorithms can be roughly divided into two types, sequential and nonadaptive. A *sequential* algorithm conducts the tests one by one and the outcomes of all previous tests can be used to set up the later test. A *nonadaptive* algorithm specifies all tests in advance so that all tests can be conducted simultaneously; thus forbidding using the information of previous tests to design later tests. Sequential algorithms require fewer number of tests in general, because extra information help for more efficient test designs. Nonadaptive algorithms permit to conduct all tests simultaneously, thus saving time for testing. Typically, the main concern of group testing is to minimize the number of tests required to identify all positive items. Therefore, sequential algorithms have dominated the literature. But in the applications to molecular biology, it is another thing; while minimizing the number of tests is still important, two other goals emerge.

In the applications to molecular biology, an experiment corresponding to a group test could take several hours or even several days. Thus, it is impractical to perform the experiments sequentially and great importance is attached to *nonadaptive group testing algorithms*, also called *pooling designs* in the molecular biology literature, in which all experiments are performed simultaneously. Note that pooling designs lead to an attached decoding problem: How $P$ is identified from the outcomes of the pooling designs?

Another feature of biological experiments is that errors in the outcomes cannot be ignored. With experimental errors, test outcomes may consist of false negative outcomes and false positive outcomes. In practice, the decoding issue becomes even more difficult due to experimental errors. So the second goal is to control the experimental errors, which has rarely been studied in the classical group testing literature, so that even though errors occur the positive items can still be identified.

So far, there has been a number of related surveys in this area (Balding et al. 1996; D'yachkov and Rykov 1983; Du and Hwang 2000; Ngo and Du 2000). To our best knowledge, however, none of which takes a look at group testing through the angle of decoding algorithm. The angle we use in this paper to cut through these models is the decoding algorithm. From this angle, we see the simplicity and integrity of the pooling design theory in the sense that all models share the same basic structure in their decoding algorithms. We also see how the differences in the models are reflected in the modifications of the basic structure.

The rest of the paper is organized as follows. Section 2 introduces several common classic models, from simple one to complicated ones, with the presence of inhibitors and experimental errors. Section 3 introduces the concept of group testing on complexes and extends the models mentioned in Sect. 2 to counterparts on the complex model.

## 2 Decoding algorithms for the classic model

A nonadaptive group testing algorithm is usually represented by the incidence matrix where rows are labeled by pools and columns by items. Represent a column $C$ by the set of row indices where $C$ has 1-entries. Then we can talk about the union and the intersection of columns. A pool with a negative/positive outcome is called a negative/positive pool. Denote $V$ as the outcome-set of indices of positive pools. Let $C^+$ denote a positive column and $C^-$ a negative column. Define $t_0^V(C) \equiv |C \setminus V|$ and $t_1^V(C) \equiv |C \cap V|$, i.e., the number of negative (positive) pools in which column $C$ appears, respectively.

### 2.1 The basic model

A matrix is said to be *d-disjunct* if for any $d+1$ columns $C_0, C_1, \ldots, C_d$,

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

That means there is at least one row where $C_0$ has a 1-entry and every $C_i$, $1 \leq i \leq d$, has a 0-entry.

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

1 **for** each item $C \in N$
2 **compute** $t_0^V(C)$
3 **if** $t_0^V(C) \leq e$
4 **then** $D \leftarrow D \cup C$
5 **return** $D$

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

**$d$-BASIC ALGORITHM**

0 **use** a $d$-disjunct matrix
1 $V \leftarrow$ the outcome-set

2 $D \leftarrow \emptyset$
3 **ITEM SELECTION**$(N, V, D, 0)$

**Theorem 2.1** *The* $d$**-BASIC ALGORITHM** *can identify the up-to-*$d$ *positive items.*

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

2.2 The error-tolerant model

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

A matrix is said to be $(d; z)$-*disjunct* if for any $d + 1$ columns $C_0, C_1, \ldots, C_d$,

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

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

$(d, e)$**-E ALGORITHM**
0 **use** a $(d; 2e + 1)$-disjunct matrix
1 $V \leftarrow$ the outcome-set
2 $D \leftarrow \emptyset$
3 **ITEM SELECTION**$(N, V, D, e)$

**Theorem 2.2** *The* $(d, e)$**-E ALGORITHM** *can identify the up-to-*$d$ *positive items with at most* $e$ *errors.*

*Proof* Assume the number of errors is at most $e$. For each negative item $C^-$, by the $(d; 2e + 1)$-disjunctness property, there exist at least $2e + 1$ rows intersecting $C^-$ but none of $P$. Therefore the pools corresponding to these rows must test negative. Even for the worst case that $e$ outcomes are erroneous, $C^-$ still appears in at least $e + 1$ negative pools, i.e., $t_0^V(C^-) \geq e + 1$. On the other hand, the outcomes of the pools containing the positive item $C^+$ should be positive except for the occurrence of errors. Hence $t_0^V(C^+) \leq e$. Thus, we can determine, via the function $t_0^V(C)$, whether $C$ is positive or not. □

2.3 The error-tolerant inhibitor (EI) model

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

$(d, h, e)$**-EI ALGORITHM**

**use** a $(d + h; 2e + 1)$-disjunct matrix
$V \leftarrow$ the outcome-set
$D \leftarrow \emptyset$
$O \leftarrow \emptyset$
**for** every item $C \in N$
    **compute** $t_1^V(C)$
    **if** $t_1^V(C) \leq e$
        **then** $O \leftarrow O \cup C$
**for** all $h$-subsets $S \subseteq O$
    $V \leftarrow V \cup (\bigcup_{\in S} C)$
    **ITEM SELECTION**$(N \setminus O, V, D, e)$

*Remark* D'yachkov et al. (2001) first gave a nonadaptive algorithm for the inhibitor model without erroneous outcomes. The basic idea is to restore all positive outcomes neutralized by inhibitors and their method exhaustively searches all $\binom{n}{h'}$, $h' \leq h$, $h'$-subsets of the $n$ items. Hwang and Liu (2003) gave a more efficient decoding algorithm with error-tolerance that can substantially reduces the number of searching operations down to $\binom{|O|}{h}$, where $O$ is a set containing all inhibitors but no positives. Computing $t_0^V(C)$ and $t_1^V(C)$ for each item $C$ as a prior operation, they partition the $n$ items into four sets so that all inhibitors are separated from all positives. Here we give a simplified version. The idea of this algorithm is to first collect all inhibitors into the set $O$, and then identify a column $C$ as positive if there exists one $S$ for which $t_0^V(C) \leq e$ under the outcome vector $V$ adjusted by $S$.

**Theorem 2.3** *The* $(d, h, e)$**-EI ALGORITHM** *can identify all positive items under the* $(d, h, e)$-**EI** *model.*

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

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

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

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

### 2.4 The general error-tolerant inhibitor (GEI) model

In the simplest inhibitor model, the model discussed in Sect. 2.3, the mere existence of a single inhibitor dictates the outcome to be negative regardless of the presence of positive items. This notion has been extended to the $k$-inhibitor model (De Bonis and Vaccaro 2003) which requires the existence of $k$ inhibitors to dictate a negative outcome. We could make even more complicated assumption that each set of $k$ inhibitors cancel the effect of a set of $g$ positive items, but practically, accurate information of $k$ and $g$ is usually not available. Thus in the general inhibitor model, we only assume the existence of some kind of canceling effect between the inhibitors and the positive items, but no further quantifiable information. Surprisingly, a decoding algorithm exists even under such ambiguity.

A result by Chang et al. (2007) implies that a $(d + h; 2e + 1)$-disjunct matrix identifies all positive items under the $(d, h, e)$-**GEI** model as well as the $(d, h, e)$-**EI** model. The main idea is similar to that in the $(d, h, e)$-**EI** model, that is, restoring all possible positive outcomes neutralized by inhibitors. Unfortunately, the same method on separating all inhibitors from all positives in advance does not work in this model. So, instead of searching all $h$-sets in $O$, we have to search all $h$-sets in $N$.

$(d, h, e)$-**GEI ALGORITHM**

0 **use** a $(d + h; 2e + 1)$-disjunct matrix
1 $V \leftarrow$ the outcome-set
2 $D \leftarrow \emptyset$
3 **for** all $h$-subsets $S \subset N$
4 $V \leftarrow V \cup (\bigcup_{C \in S} C)$
5 **ITEM SELECTION**$(N \setminus S, V, D, e)$

**Theorem 2.4** *The* $(d, h, e)$-**GEI ALGORITHM** *can identify all positive items under the* $(d, h, e)$-**GEI** *model.*

*Proof* The proof is similar to the proof of Theorem 2.3. Note that a positive item $C$ is identified when $S$, $C \notin S$, is a set containing all inhibitors. □

## 3 Decoding algorithms for the complex model

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

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

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

Let $H$ denote the given set of complexes, then $H$ can be viewed as a hypergraph with items as vertices and complexes as edges. Accordingly, the group testing problem on complexes is related to the graph testing problem on searching a hidden subgraph $P$ in a given graph $H$, which consists of the set of positive edges. Suppose $H$ is a rank-$r$ graph (each edge consists of at most $r$ vertices) and our only knowledge of $P$ is $|P| \leq d$. Let $\cap S$ denote the intersection of all columns in $S$. A binary matrix is said to be $(H_{\bar{r}} : d)$-*disjunct* if for any $d+1$ edges $e_0, e_1, \ldots, e_d$ in $H_{\bar{r}}$,

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

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

A binary matrix is said to be $(d, r]$-*disjunct* (different from $(d, r)$-disjunct), first studied by Mitchell and Piper (1988), if for any $d+r$ columns $C_1, C_2, \ldots, C_{d+r}$,

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

That means for any $d+r$ columns there exists a row in which each of the $r$ designated columns has a 1-entry and each of the other $d$ columns has a 0-entry. Such a property was further studied in (Chen et al. 2007; Kim and Lebedev 2004; Stinson and Wei 2004; Stinson et al. 2000), sometimes under the name of cover-free families or superimposed codes with application to the secure key distribution problem.

Chen et al. (2007) established a connection between the complex model and the secure key distribution problem, and showed the relation that

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

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

### 3.1 The error-tolerant complex (EC) model

In the **EC** model, we consider the problem on the complex model with at most $e$ erroneous outcomes. For a subset $X$ of items, define $t_0^V(X) \equiv |\cap X \setminus V|$ and $t_1^V(X) \equiv$

$|(\cap X) \cap V|$, i.e., the number of negative (positive) pools in which every item in $X$ appears, respectively. Notice that $\cap S$ is the set of pools in which every item in $S$ appears.

Stinson and Wei (2004) first gave an error-tolerant version of the $(d, r]$-disjunct matrix. A binary matrix is said to be $(d, r; z]$-disjunct if for any $d + r$ columns $C_1, C_2, \ldots, C_{d+r}$,

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

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

**for** each complex $X \in H$
  **compute** $t_0^V(X)$
  **if** $t_0^V(X) \leq e$
    **then** $D \leftarrow D \cup X$
**return** $D$

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

**use** a $(d, r; 2e + 1]$-disjunct matrix
$V \leftarrow$ the outcome-set
$D \leftarrow \emptyset$
**COMPLEX SELECTION**$(H, V, D, e)$

**Theorem 3.1** *The* $(d, r, e)$**-EC ALGORITHM** *can identify all positive complexes under the* $(d, r, e)$-**EC** *model.*

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

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

### 3.2 The error-tolerant inhibitor complex (EIC) model

In this subsection, we will introduce a synthetic model on complexes with the presence of inhibitors and erroneous outcomes. We use the parameters $(d, h, r, e)$ to denote the assumption that among the complexes which are subsets of $n$ items, there are at most $d$ positive complexes each consisting of at most $r$ items, and there are at most $h$ inhibitors and at most $e$ erroneous outcomes.

Consider the simplest inhibitor model where the mere existence of a single inhibitor dictates the outcome to be negative, regardless of the presence of positive complexes. The first decoding algorithm we provide here is similar to that in Sect. 2.3 except replacing items by complexes.

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

0 **use** a $(d + h, r; 2e + 1]$-disjunct matrix
1 $V \leftarrow$ the outcome-set
2 $D \leftarrow \emptyset$
3 $O \leftarrow \emptyset$
4 **for** every complex $X \in H$
5 **compute** $t_1^V(X)$
6 **if** $t_1^V(X) \leq e$
7 **then** $O \leftarrow O \cup X$
8 **for** all $h$-subsets $S \subseteq O$
9 $V \leftarrow V \cup (\bigcup_{X \in S}(\cap X))$
10 **COMPLEX SELECTION**$(H \setminus O, V, D, e)$

**Theorem 3.2** *The* $(d, h, r, e)$-**EIC ALGORITHM I** *can identify all positive complexes under the* $(d, h, r, e)$-**EIC** *model.*

*Proof* Similar to the proof of Theorem 2.3 except replacing items by complexes. □

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

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

0 **use** a $(d + h, r; 2e + 1]$-disjunct matrix
1 $V \leftarrow$ the outcome-set
2 $D \leftarrow \emptyset$
3 **for** all $h$-subsets $S \subseteq N$
4 $V \leftarrow V \cup (\bigcup_{X \in S} X)$
5 **COMPLEX SELECTION**$(H \setminus S, V, D, e)$

**Theorem 3.3** *The* $(d, h, r, e)$-**EIC ALGORITHM II** *can also identify all positive complexes under the* $(d, h, r, e)$-**EIC** *model.*

*Proof* The proof is similar to that of Theorem 3.2 except that the restoring operation runs through all $h$-subsets $S \subseteq N$. □

### 3.3 The general error-tolerant inhibitor complex (GEIC) model

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

**Theorem 3.4** *The* $(d, h, r, e)$**-EIC ALGORITHM II** *identifies all positive complexes under the* $(d, h, r, e)$-**GEIC** *model as well.*

*Proof* Note that the proof of Theorem 3.2 does not depend on quantifiable information about the canceling effect. With a slight modification of the proof of Theorem 3.2, one can conclude that the $(d, h, r, e)$**-EIC ALGORITHM II** also identifies all positive complexes under the $(d, h, r, e)$-**GEIC** model. □

## References

Alon N, Beigel R, Kasif S, Rudich S, Sudakov B (2004) Learning a hidden matching. SIAM J Comput 33:487–501

Balding DJ, Bruno WJ, Knill E, Torney DC (1996) A comparative survey of nonadaptive pooling designs. In: Genetic mapping and DNA sequencing. IMA volumes in mathematics and its applications. Springer, Berlin, pp 133–154

Chang FH, Chang HL, Hwang FK (2007) Pooling designs for clone library screening in the inhibitor complex model, to appear

Chen HB, Du DZ, Hwang FK (2007) An unexpected meeting of four seemingly unrelated problems: graph testing, DNA complex screening, superimposed codes and secure key distribution. J Comb Opt, to appear

Chen HB, Fu HL, Hwang FK (2007) An upper bound of the number of tests in pooling designs for the error-tolerant complex model. Opt Lett, to appear

De Bonis A, Vaccaro U (1998) Improved algorithms for group testing with inhibitors. Inform Process Lett 67:57–64

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

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

Du DZ, Hwang FK (2000) Combinatorial group testing and its applications, 2nd ed. World Scientific, Singapore

D'yachkov AG, Rykov VV (1983) A survey of superimposed code theory. Probl Control Inf Theory 12:229–242

D'yachkov AG, Macula AJ, Torney DC, Vilenkin PA (2001) Two models of nonadaptive group testing for designing screening experiments. In: Attkinson AC, Hackl P, Muller WG (eds) Proceedings of the 6th international workshop in model oriented design and analysis. Physica, Berlin, pp 63–75

D'yachkov AG, Vilenkin PA, Macula AJ, Torney DC (2002) Families of finite sets in which no intersection of $\ell$ sets is covered by the union of $s$ others. J Comb Theory Ser A 99:195–218

Farach M, Kannan S, Knill E, Muthukrishnan S (1997) Group testing problem with sequences in experimental molecular biology. In: Proceedings of the compression and complexity of sequences, pp 357–367

Hwang FK, Liu YC (2003) Error-tolerant pooling designs with inhibitors. J Comput Biol 10:231–236

Kim HK, Lebedev V (2004) On optimal superimposed codes. J Comb Des 12:79–91

Macula AJ, Popyack LJ (2004) A group testing method for finding patterns in data. Discret Appl Math 144:149–157

Mitchell CJ, Piper FC (1988) Key storage in secure networks. Discret Appl Math 21:215–228

Ngo HQ, Du DZ (2000) A survey on combinatorial group testing algorithms with applications to DNA library screening. In: DIMACS Ser Discret Math Theor Comput Sci, vol 55,. American Mathematical Society, Providence, pp 171–182

Stinson DR, Wei R (2004) Generalized cover-free families. Discret Math 279:463–477

Stinson DR, Wei R, Zhu L (2000) Some new bounds for cover-free families. J Comb Theory Ser A 90:224–234

Torney DC (1999) Sets pooling designs. Ann Comb 3:95–101