JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 10, Number 2, 2003

Pp. 231–236

Error-Tolerant Pooling Designs with Inhibitors

F.K. HWANG and Y.C. LIU

ABSTRACT

Pooling designs are used in clone library screening to efficiently distinguish positive clones from negative clones. Mathematically, a pooling design is just a nonadaptive group testing scheme which has been extensively studied in the literature. In some applications, there is a third category of clones called "inhibitors" whose effect is to neutralize positives. Specifically, the presence of an inhibitor in a pool dictates a negative outcome even though positives are present. Sequential group testing schemes, which can be modified to three-stage schemes, have been proposed for the inhibitor model, but it is unknown whether a pooling design (a one-stage scheme) exists. Another open question raised in the literature is whether the inhibitor model can treat unreliable pool outcomes. In this paper, we answer both open problems by giving a pooling design, as well as a two-stage scheme, for the inhibitor model with unreliable outcomes. The number of pools required by our schemes are quite comparable to the three-stage scheme.

Key words: nonadaptive group testing, inhibitors, error-tolerant.

1. INTRODUCTION

SEQUENCING A SET OF CLONES OFTEN RELIES on identifying the clones which contain a given probe. For examples, in physical mapping, the probe can be a sequence-tagged site which is a unique subsequence in the target sequence. The identification of a clone containing this probe essentially locates the clone. In a DNA array, a probe is a given l-tuple and a positive identification confirms the existence of such an l-tuple in the target sequence. We will assume that the setting is in the first application, namely, to identify which clones in the given set contain the probe. A clone is called a *positive* if it contains the probe, and a *negative* if not. A *pool* is a subset of clones put together for a joint assay with two possible outcomes: a *negative pool* signifies that there's no positive in the pool, a *positive pool* signifies otherwise, namely, that there is at least one positive in the pool. A *pooling design* is a 0-1 matrix where the columns are the set of clones, the rows are the set of pools, and a 1-entry in cell (i, j) signifies that clone j is in pool i.

In some biological applications, there is a third category of clones called *inhibitors* whose presence in a pool dictates a negative outcome, regardless of the presence of a positive in the pool. While the pooling design corresponds to the classical nonadaptive group testing problem (Du and Hwang, 2000), the presence of inhibitors presents a new group testing model not considered in the group testing literature. Farach *et al.* (1997) first introduced this model. Let n denote the total number of clones including at most d positives and at most r inhibitors. They gave a randomized algorithm to identify all positives in

Department of Applied Mathematics, National Chiao Tung University, Hsìnchu 300, Taiwan.

$O((d+r)\log n)$ tests, assuming $d+r \ll n$. Bonis and Vaccaro (1998) gave a deterministic algorithm in $O((r^2+d)\log n)$ tests. However, both algorithms are sequential in natural; specifically, tests cannot be performed in parallel (as some tests depend on the results of other tests). It is possible to convert the De Bonis and Vaccaro algorithm into a three-stage algorithm (tests in a given stage can be performed in parallel) by increasing the number of tests to $O((r^2+d^2)\log n)$. But it remains an open question whether there exists a pooling design (1-stage) for the inhibitor model. Further, experimental or recording errors may be made. De Bonis and Vaccaro raised the open problem of treating errors in the inhibitor model. In this paper, we answer both open problems by proving the existence of an error-tolerant pooling design with $O((d^2+r^2+e^2)\log n)$ tests when there are at most e unreliable outcomes. We also give a two-stage algorithm in $O((d^2+r^2+e^2)\log n)$ tests; each stage can have at most e errors.

2. PRELIMINARY

Since a pooling design is a 0-1 matrix, whether we can identify positives from negatives and inhibitors by decoding the pooling outcomes apparently depends on the structure of this matrix. Indeed, when there are only positives and negatives, the case corresponding to the nonadaptive group testing problem, we can identify positives from negatives by a 0-1 matrix with some particular structure.

Consider a $t \times n$ 0-1 matrix M where R_i and C_j denote row i and column j. A 1-entry in cell (i, j) is *covered* by a set X of columns if at least one column in X has a 1-entry in row i. A set A of columns is covered by a set B of columns if every 1-entry in A is covered by B. M is called *d-disjunct* if no union, or Boolean sum, of up to d columns covers any other column.

Note that a set of d columns can be viewed as a *candidate set* of positives; while the union of this candidate set, also a binary vector, then corresponds to the set of pools which give positive outcomes. Since the d-disjunct property guarantees that all negatives have at least a 1-entry not covered by the set of positive pools, we can then identify those covered by the set of positive pools as positives. It is well known (Hwang and Sós, 1987) that an upper bound of the number of pools in a d-disjunct matrix is $O(d^2 \log n)$. Figure 1 gives an example of a 2-disjunct matrix.

As we mentioned in Section 1, the inhibitors present a new group testing model not considered before. We prove in Section 4 that a $(d+r+2e)$-disjunct matrix can be used as the pooling design for the inhibitor model with e errors. We also give a two-stage method in Section 3. It contains an $(r+2e)$-disjunct and a $(d+2e)$-disjunct matrix in the first and second stage, respectively. Note that the test numbers of both our methods have the same complexity as the three-stage method of De Bonis and Vaccaro and is only slightly more than their sequential method.

Before introducing our methods, some lemmas about the structure of disjunct matrices are needed. A column C_i in a 0-1 matrix is *isolated* if there exists one row that contains C_i only. A 1-entry has a

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	outcome
R_1	1	**1**	1	1									1
R_2	1				**1**	1	1						1
R_3	<u>1</u>							<u>1</u>	<u>1</u>	<u>1</u>			0
R_4		**1**					1		1		1		1
R_5			1		**1**					1	1		1
R_6				<u>1</u>		<u>1</u>		<u>1</u>			<u>1</u>		0
R_7		**1**				1				1		1	1
R_8				1	**1**				1			1	1
R_9			<u>1</u>				<u>1</u>	<u>1</u>				<u>1</u>	0

FIG. 1. This is a 2-disjunct matrix. Any union of up to two columns doesn't cover any other column. Then suppose C_2 and C_5 are the positives; all negatives have at least a 1-entry (those appear with an underline) not covered by the union of C_2 and C_5.

1-outcome (0-outcome) if it lies in a positive (negative) pool. A column has m 1-outcomes (0-outcomes) if m of its one-entries has 1-outcomes (0-outcomes).

Note that the deletion of the isolated column and rows that contain it in a disjunct matrix does not destroy the disjunctness. Further, it's easy to determine whether an isolated clone is a positive. We assume that a disjunct matrix has no isolated column. D'yachkov and Rykov (1982) proved the following.

Lemma 2.1. *Every column should have column weight at least $m + 1$ in a m-disjunct matrix.*

Lemma 2.2. *Let K denote a set of k row indices. There exists a set of at most k columns whose union covers K.*

Proof. For each row index in K, select any columns with a one-entry in that row. Then the set of at most k chosen columns covers K. ■

Suppose there are at most d positives, r inhibitors, and e errors. Then we have the following.

Lemma 2.3. *A positive should have at least $(m - r - e + 1)$ 1-outcomes in a m-disjunct matrix.*

Proof. Suppose there is a positive C which has at most $m - r - e$ 1-outcomes. By Lemma 2.2, there exists a set of at most $m - r - e$ columns that are different from C because there's no isolated column, covering the row indices of these 1-outcomes. Further, all its other (at least $r+e+1$) one-entries are covered by the r inhibitors and the up to e errors. Since there exists a set E of e columns, $C \notin E$ when there's no isolated column, covering the row indices of the up to e errors. Hence, at most $(m - r - e) + (r + e) = m$ columns cover C, contradicting the m-disjunct property. ■

Lemma 2.4. *A negative should have at least $(m - d - e + 1)$ 0-outcomes in a m-disjunct matrix.*

Proof. Suppose to the contrary that a negative c has only $(m - d - e)$ 0-outcomes. Let E be defined as in Lemma 2.3. Then all the 1-entries of c with 1-outcomes are covered by at most $d + e$ columns, and all with 0-outcomes by at most $(m - d - e)$ columns, leading to the conclusion that c is covered by m columns, a contradiction to the m-disjunctness. ■

Lemma 2.5. *An inhibitor should have at most e 1-outcomes in any disjunct matrix.*

Proof. An inhibitor should have no 1-outcome if there is no error. Those e errors can result in at most e 1-outcomes for an inhibitor. ■

3. THE 2-STAGE METHOD

In this section, we will first state our algorithm, then prove its correctness.

The first stage:
Pooling: Use an $(r + 2e)$-disjunct matrix.
Decoding: Collect all clones which have at most e 1-outcomes as a set AI.
Take the complement of AI as PN.
The second stage:
Pooling: Use a $(d + 2e)$-disjunct matrix on clones in the set PN.
Decoding: Collect all clones which have at most e 0-outcomes as a set P.
Output the set P as our positives.

Lemma 3.1. *The set AI contains all inhibitors and no positives.*

Proof. By Lemma 2.3, a positive should have at least $(e+1)$ 1-outcomes in an $(r+2e)$-disjunct matrix; hence, no positive will appear in AI. Further, since all clones with at most e 1-outcomes are chosen into AI, by Lemma 2.5, all inhibitors are contained in AI. ■

Theorem 3.2. *The set P contains all positives and nothing else.*

Proof. By Lemma 2.4, a negative should have at least $(e+1)$ 0-outcomes; hence, those with at most e 0-outcomes can not be a negative. They are all positives. ■

Actually, what we do in the first stage is leave all inhibitors in the set AI and collect all positives into the set PN. But there can be negatives in PN; hence, we need a second stage. Since there is no inhibitor in PN, we can then do the classical group testing in the second stage to find all positives.

4. THE 1-STAGE METHOD

Pooling: Use a $(d+r+2e)$-disjunct matrix.
Decoding:
Step I: Partition the clones into 4 sets:
P consists of those with at most $r+e$ 0-outcomes
N consists of those with at least $e+1$ but at most $d+e$ 1-outcomes
O consists of those with at most e 1-outcomes
R consists of the rest
Step II: If $R \neq \varnothing$
Denote the outcome vector as V.
For all possible r-subsets of O
a. denote the union of the r-subset as V'.
b. let V union V' be U.
c. if clone c in R has at most e 0-outcomes under U, then put it into P.
Step III: Output P as our set of positives.

Lemma 4.1. *After Step I, N is contained in the set of all negatives, and P is contained in the set of all positives. O contains all inhibitors and no positives.*

Proof. By Lemma 2.3, a clone with at most $d+e$ 1-outcomes cannot be positive. An inhibitor has at most e 1-outcomes. So an inhibitor cannot appear in N. Hence, there are only negatives in N.

By Lemma 2.4, a clone with at most $r+e$ 0-outcomes cannot be negative. Further, the column weight is at least $(d+r+2e+1)$ in a $(d+r+2e)$-disjunct matrix; these clones then have at least $(d+e+1) > e$ 1-outcomes. By Lemma 2.5, inhibitors will not appear in P either.

By Lemma 2.4, a clone with at most e 1-outcomes cannot be a positive. An inhibitor cannot have more than e 1-outcomes. Hence, O contains all inhibitors but no positives. ■

Lemma 4.2. *A clone c in R is positive $\Leftrightarrow$ there exists at least one r-subset of O such that c has at most e 0-outcomes under U.*

Proof. ($\Rightarrow$) Since O contains all inhibitors (whose number is at most r), some chosen r-subset in Step II should contain all inhibitors. Then the vector V' formed by such an r-subset corrects the false negative outcomes caused by the inhibitors. If c is positive, then every pool containing c has positive outcome except at most e pools with unreliable outcomes. Thus c is in P.

($\Leftarrow$) Suppose a clone c in R is negative. We can view U as the union of $d+r$ clones. Then $|c \setminus U| > e$ for otherwise c can be covered by $U \cup E$, where E is a set of e columns, a contradiction to the $(d+r+2e)$-disjunctness since $|U \cup E| \leq d+r+e$. ■

Corollary 4.3. *The set P contains all positives and nothing else.*

In this one-stage method, we first partition all clones into four parts. After Step I, part P contains only positives, N contains only negatives, and O contains all inhibitors and no positives. This is guaranteed by Lemmas 2.3, 2.4, and 2.5. Clones in part R maybe either positives or negatives; hence, we need Step II. Then what we do in Step II is to recover the false negative outcomes caused by inhibitors by trying all possible r-subsets in O. By Lemma 4.1, at least one of these r-subset will contain the true inhibitor-subset. Then each positive in R will be covered by this U and thus collected into P. On the other hand, due to the sufficiency part in Lemma 4.2, we can always keep negatives in R from P. Since there may be more than one r-subset that can recover the false negative outcomes correctly, we cannot recognize which one contains the true inhibitors-subset.

Then, we estimate the time complexity of this method. Most disjunct matrices are constructed by using some combinatorial designs (see Du and Hwang [2000] for a general introduction) which treats the clone symmetrically. In particular, most such designs have a constant column weight k; i.e., each column has exactly k 1-entries. The partition into P, N, O, and R requires one to count for each column how many of its 1-entries have 0-outcomes and how many have 1-outcomes. Thus, the partition takes $O(kn)$ time.

Next, we estimate the time complexity of Step II. We first give a case for which Step II is not needed.

Corollary 4.4. *$R = \varnothing$ after Step I if the $(d + r + 2e)$-disjunct matrix has constant column weight $(d + r + 2e + 1)$.*

Proof. By Lemma 2.4, we put a clone into P in Step I when it has at most $r + e$ 0-outcomes. That is, it is a sufficient condition for a positive. However, in a matrix with constant column weight $(d + r + 2e + 1)$, a clone with at most $r + e$ 0-outcomes implies it has at least $(d + e + 1)$ 1-outcomes, and by Lemma 2.3, this is the necessary condition for a positive. That is, all positives are collected into P. A similar argument works for negatives. ■

Assume $R \neq \varnothing$; then the time complexity of Step II is $O(\binom{|O|}{r}k|R|)$. To estimate $|O|$ and $|R|$, we need to be more specific about the $(d + r + 2e)$-disjunct matrix. A popular construction is to have constant column weight $k \geq d + r + 2e + 1$, while each pair of columns intersect the same row (having a 1-entry) at most once (called the *1-intersection property*). Then $p = \frac{k}{t}$ is approximately the probability of the appearance of a 1-entry in the matrix. We will give a rough analysis by assuming that the distribution of 1-entries in the rows is random.

We first estimate $|R|$. A positive C is in R if it has at least $r + e + 1$ 0-outcomes, which is impossible if the $(d + r + 2e)$-disjunct matrix has the 1-intersection property since the r inhibitors can cause at most r 0-outcomes, leaving a total of at least $e + 1$ 0-outcomes. Even all e errors occur in these rows, there is still 1 0-outcome unexplained.

The above argument is very conservative in explaining the 0-outcomes. So even if a $(d+r+2e)$-disjunct matrix does not have the 1-intersection property, we expect $|R|$ to be small.

For $|R| \neq 0$, we compute the probability that a negative c is in O, i.e., the probability that it has at most e 1-outcomes.

Without errors, c gets a 0-outcome at a row if either an inhibitor is present or none of the positive is present in that row. The probability $P(c^-)$ of this is

$$P(c^-) = 1 - (1 - p)^r[1 - (1 - p)^d].$$

Hence the probability that c has z 0-outcomes is

$$f(z) = \binom{k}{z} P(c^-)^z(1 - P(c^-))^{k-z}.$$

However, there are e' errors present where $e' \leq e$. Suppose i of them make the z 0-outcomes of c become 1-outcomes, and j of them make the $k - z$ 1-outcomes become 0-outcomes. Let c_1 denote the number of

1-outcomes. Then

$$P(c_1 \le e) = \sum_{z=0}^{k} \sum_{i=0}^{e'} \sum_{j=k-z+i-e}^{e'-i} f(z) \frac{\binom{z}{i}\binom{k-z}{j}\binom{t-k}{e'-x}}{\binom{t}{e'}},$$

because we need at least $k-z+i-e$ 1-outcomes become 0-outcomes such that c has less than e 1-outcomes. Therefore,

$$E(|O|) = r + (n-d-r) \cdot P(c_1 \le e).$$

For $e = 0$, then $c \in O$ if and only if $z = k$. The probability is $P(c^-)^k$.

5. CONCLUSION

In this paper, we offer an error-tolerant 1-stage method to solve the inhibitor model. This solves the existence problem for a pooling design with inhibitors and errors. As in the classical group testing problem, we use disjunct matrices as our pooling designs. But the decoding algorithm is slightly more complicated than that in the classical group testing problem. Fortunately, by Corollary 4.4 and the analysis of $|R|$ and $|O|$, we can still decode quickly if the disjunct matrices we use as pooling designs have a certain constant column weight, and many well-studied (Du and Hwang, 2000) methods for constructing disjunct matrices actually do yield constant column weight.

After the submission of the paper, the authors became aware of a recent conference paper by D'yachkov *et al.* (2001) which gave a 1-stage method for the inhibitor model without error. Besides the errorless assumption, their method exhaustively checks all $\binom{n}{r'}$, $r' \le r$, r'-subsets of the n clones for each clone, and hence could be too computation intensive for practical use for large n and moderate r.

ACKNOWLEDGMENT

Research partially supported by the Republic of China NSC grant 90-2115-M-009-029.

REFERENCES

Bonis, A.D., and Vaccaro, U. 1998. Improved algorithms for group testing with inhibitors. *Information Processing Letters* 67, 57–64.

Du, D., and Hwang, F.K. 2000. *Combinatorial group testing and its applications*, 2nd ed., World Scientific, Singapore.

D'yachkov, A.G., Macula, A.J., Torney, D.C., and Vilenkin, P.A. 2001. Two models of nonadaptive group testing for designing screening experiments. *Proc. 6th Int. Workshop on Model-Oriented Design and Analysis* 63–75.

D'yachkov, A.G., and Rykov, V.V. 1982. Bounds of the length of disjunct codes. *Problems Control Inform. Thy.* 11, 7–13.

Farach, M., Kannan, S., Knill, E., and Muthukrishnan, S. 1997. Group testing problems with sequences in experimental molecular biology. *Proc. Compression and Complexity of Sequences*, 357–367. IEEE.

Hwang, F.K., and Sós, V.T. 1987. Non-adaptive hypergeometric group testing. *Studia Scient. Math. Hungarica* 22, 257–263.

Address correspondence to:
Y.C. Liu
Department of Applied Mathematics
National Chiao Tung University
Hsìnchu 300, Taiwan

E-mail: ycliu@usc.edu