Annals of Combinatorics 3 (1999) 61-69

Annals of Combinatorics

Probabilistic Nonadaptive Group Testing in the Presence of Errors and DNA Library Screening*

Anthony J. Macula

Department of Mathematics, State University of New York, College at Geneseo, Geneseo, NY 14454, USA
macula@geneseo.edu

Received July 29, 1998

AMS Subject Classification: 05B20, 05D05, 62K99

Abstract. We use the subset containment relation to construct a probabilistic nonadaptive group testing design and decoding algorithm that, in the presence of testing errors, identifies many positives in a population. We give a lower bound for the expected portion of positives identified as a function of an upper bound on the number of testing errors.

Keywords: nonadaptive group testing, minimal families of k-sets, DNA library screening, testing with errors, representatives of subsets

1. Group Testing and DNA Library Screening

Suppose we have a finite ground set or *population* containing elements which can be uniquely characterized as positive or negative. We refer to the collection of *positive elements*, which is initially unknown, as the *positive subset* P. In the abstract *group testing problem*, P must be identified by performing 0, 1 tests on subsets or *pools* of the population. A pool is said to be positive (1) if the test result indicates that a member of P is in that pool; the pool is said to be negative (0) if the test result indicates otherwise. Note that the terms positive element, positive subset, and positive pool have different meanings.

Using probes to screen DNA libraries of clones fits the group testing paradigm in the following way: The population is the DNA library which consists of thousands of separate recombinant DNA clones each of which represents some contiguous piece of a contiguous superpiece of DNA. To help understand what a DNA library is, think of several copies of an identical but incredibly long word (i.e., a chromosome), each of which has been cut into thousands of contiguous pieces. Take each piece and copy that

* The algorithms contained herein are part of The State University of New York Research Foundation invention C1230-125, Probabilistic and Combinatorial Nonadaptive and Two-Stage Group Testing and DNA Library Screening by A. Macula and K. Anne.

letter string onto its own separate small piece of paper . The thousands of resulting pieces of paper (i.e., clones) essentially constitute a DNA library.

A unique, identifiable, predetermined, and contiguous DNA subpiece is called a *sequenced tagged site* (STS). A clone is called *positive* for an STS if it contains that STS (see Example 1.1). A pool is a subset of the clones that are mixed together and tested by exposing the entire group to a chemical probe. A pool is *labeled* positive for an STS if the probe chemically indicates its presence. In other words, if the tests are error-free, then a pool is labeled positive for an STS if and only if that pool contains at least one clone that contains that STS.

After the same DNA library has been repeatedly screened with different probes and the clones positive for each individual STS have been identified, then clones positive for more than one STS are used in the construction of gene maps precisely because the interval of DNA between two STSs is contained in each clone positive for both of those STSs (see Example 1.1). This is one reason why it is important to identify as many positives clones as possible—the more positive clones per STS, the higher the probability of identifying clones positive for multiple STSs.

Example 1.1. Let the DNA superpiece be AAAGCGTCTTAACCGATAGGCAACTTG. Suppose the library is $\{C_1, C_2, C_3, C_4, C_5\}$ where C_1 = AAAGCGTCTTAA, C_2 = GTCTTAACCGA , C_3 =CCGATAGGCAAC , C_4 = CTTAACCGATAGGC, and C_5 = AGGCAACTTG . Let STS_1 = AAA and STS_2 =TAA. Then C_1 is positive for STS_1 and C_1, C_2 and C_4 are positive for STS_2. Note that C_1 is positive for both STSs.

Primarily because the same DNA library is screened with many different probes, parallel rather than sequential screening methods are generally preferred. For other screening cost factors (see [8]). Consideration of analogous factors in other testing, screening, or coding situations predates the Human Genome Project and leads to the development of *nonadaptive group testing* (NGT) (see [5]). There are two traditional categories of NGT, probabilistic nonadaptive group testing (PNGT) and combinatorial nonadaptive group testing (CNGT). An essential difference between these categories is that in PNGT one considers the average cost, and in CNGT one considers the worst cost. In both NGT categories, one must decide exactly which pools to test *before any testing occurs.* An NGT algorithm is sometimes referred to as a *one-stage* algorithm. A *two-stage* algorithm is a *nearly* nonadaptive algorithm. In a *trivial two-stage algorithm,* all nontrivial pools occur in the first stage. After the first stage is complete, one has a set CP called the *candidate positives.* In the second stage, each candidate positive is individually tested to see if it is an actual positive. In [8], a lower bound for the expected number candidate positives, as a function of the number of first stage pools, is given.

When screening DNA libraries, screening errors almost always occur during the testing procedure, and there are constraints on pool sizes. Practical algorithms must be able to identify a large portion of the positives when error rates can be as high as 10%, and it is reasonable to assume the error probability increases with pool size. This paper addresses PNGT by probabilistically analyzing a class of CNGT algorithms. In [2], a nice overview of nonadaptive pooling designs is given. Our mathod is not described there, but it has similarities to aspects of set packing and random r-set designs which are (also see [3]). We consider a class of CNGT error-free algorithms that always determine P when $|P| \leq 2$ (see [7,11,13]). Then we probabilistically analyze how well these same

algorithms identify P in the presence of errors when $|P| > 2$. For P with $|P| > 2$, we do this by computing an upper bound on $|CP|$ and the expected value of $|CP \cap P|$. Our method also gives us control over the pool sizes.

2. The Mathematical Objects in Our Algorithms

Throughout this paper, all simple lower case variables are nonnegative integers. Let $[n]$ denote $\{1, 2, \ldots, n\}$. Given set S, $|S|$ denotes its cardinality. We call a subset of $[n]$ with cardinality k a k-set. Let $\binom{[n]}{k}$ denote the k-sets of $[n]$. Let J be a subset of $[n]$ and let $\binom{J,[n]}{k}$ denote the k-sets of $[n]$ that have a nonempty intersection with J. Let $\binom{j,n}{k}$ denote the cardinality of $\binom{[j],[n]}{k}$. Then $\binom{j,n}{k} = \binom{n}{k} - \binom{n-j}{k}$ and $\binom{j,n}{k} = \binom{n}{k}$ whenever $k > n - j$.

Definition 2.1. *For $2 < k < n$, let the rows and columns of the 0, 1 matrix $\delta(j, n, 2, k)$ be respectively represented by the members of $\binom{[j],[n]}{2}$ and $\binom{[n]}{k}$ ordered lexicographically. For $T \in \binom{[j],[n]}{2}$ and $K \in \binom{[n]}{k}$, the matrix $\delta(j, n, 2, k)$ has a 1 in its (T, K)th entry if and only if $T \subset K$.*

We let $\mathbf{c}_K$ denote the column of $\delta(j, n, 2, k)$ corresponding to the k-set K. Let γ_{xy} denote the row of $\delta(j, n, 2, k)$ corresponding to $\{x, y\}$. Note $\gamma_{xy} = \gamma_{yx}$ (see Figures 1 and 2).

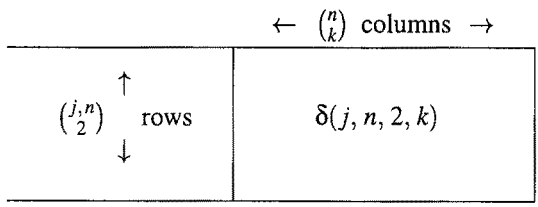

Figure 1.

$\delta(2,4,2,3)$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$
12	1	1	0	0
13	1	0	1	0
14	0	1	1	0
23	1	0	0	1
24	0	1	0	1

Figure 2.

We use the matrices $\delta(j, n, 2, k)$ to model a pooling strategy. We identify a population of cardinality $z \leq \binom{n}{k}$ with a random z-set of columns of $\delta(j, n, 2, k)$. This identification defines a submatrix, $\delta_z(j, n, 2, k)$, of $\delta(j, n, 2, k)$. A row γ_{xy} of $\delta_z(j, n, 2, k)$ determines a pool of the population in the obvious way. That is, $\mathbf{c}_K$ is in the pool determined by γ_{xy} if and only if $\{x, y\}$ is contained in K. We identify a row of $\delta_z(j, n, 2, k)$ with the pool of columns of $\delta_z(j, n, 2, k)$ that it determines.

Fix j. Since the ratio of the weight of any row to the total number of columns in $\delta(j, n, 2, k)$ is $\frac{k(k-1)}{n(n-1)}$, then using a binomial approximation to the hypergeometric distribution for the number of 1's selected from a given row of $\delta(j, n, 2, k)$ when z columns are randomly chosen, we can assume $\frac{k(k-1)}{n(n-1)}z$ approximates the number of 1's in a row of a submatrix $\delta_z(j, n, 2, k)$, because in most cases, $\frac{k(k-1)}{n(n-1)} \leq .05$ and $z \geq 5000$. This observation gives us considerable control over the pool sizes. As noted Section 1, this is an important practical consideration.

3. The Nonadaptive Algorithm with No Testing Errors

Given a population of cardinality $z \leq \binom{n}{k}$, let P represent the positives and suppose $|P| = p$. Let $j, k < n$. Identify the population with a random z-set of columns of $\delta(j, n, 2, k)$. Then P is randomly associated with a subfamily of columns of $\delta_z(j, n, 2, k)$, which in turn are represented by a p-family $\{K_1, \ldots, K_p\}$ of k-sest of $[n]$. By testing each row of $\delta_z(j, n, 2, k)$, we define an *output vector* $\mathbf{o}$ by setting $\mathbf{o}_{xy}$ equal to 1 if the test result of pool γ_{xy} is positive and 0 if not. Suppose the test results are free of any errors and we have tested each row of $\delta_z(j, n, 2, k)$ and the output vector $\mathbf{o}$ which contains no errors. We use $\mathbf{o}$ to probabilistically identify P. Since $\gamma_{xy} = \gamma_{yx}$, then $\mathbf{o}_{xy}$ and $\mathbf{o}_{yx}$ denote the same entry in $\mathbf{o}$.

Definition 3.1. *Let $\{K_1, \ldots, K_p\}$ be a randomly selected p-family of k-sets from $[n]$. Let J be a subset of $[n]$ with $|J| = j$. If for K_i, there is an element x_i in K_i, but not in any other $K_{i'}$ with $i' \neq i$, then we call x_i a representative of K_i in $\{K_1, \ldots, K_p\}$. Let $\phi_i(J, n, p, k)$ be the probability that K_i has a representative contained in J. Since $\phi_i(J_1, n, p, k) = \phi_{i'}(J_2, n, p, k)$ when $|J_1| = |J_2| = j$, we simply let $\phi(j, n, p, k)$ be the probability that K_i has a representative contained in J.*

Algorithm 1. For a population of cardinality z identified with the columns of $\delta_z(j, n, 2, k)$, test the pools identified with the rows of $\delta_z(j, n, 2, k)$ and consider the output vector $\mathbf{o}$. Search for all x in $[j]$ with the property that $|\{y : \mathbf{o}_{xy} = 1\}| = k - 1$. For each such x, from the k-sets $\{x\} \cup \{y : \mathbf{o}_{xy} = 1\}$. Take these k-set to represent the positive elements.

Note that, when using $\delta_z(j, n, 2, k)$, Algorithm 1 will never identify more than j positive objects.

Theorem 3.2. *Suppose the tests are error-free. For a population with cardinality z and $|P| = p$, the expected number of positives identified using $\delta_z(j, n, 2, k)$ in Algorithm 1 is $p \cdot \delta(j, n, p, k)$.*

Proof. Suppose P is randomly associated with the p-family $\{K_1, \ldots, K_p\}$ of k-sets of $[n]$. It is easy to see that x_i is a representative of K_i in $\{K_1, \ldots, K_p\}$ if and only if

$\{\{x_i, y\} : \{x_i, y\} \subset K_i \text{ for some } 1 \le i \le p\} = \{\{x_i, y\} : \{x_i, y\} \subset K_i\}$ and that this equality occurs if and only if $|\{y : \mathbf{o}_{x_i y} = 1\}| = k - 1$. So the probability of identifying K_1 using $\delta_z(j, n, 2, k)$ in Algorithm 1 is $\phi(j, n, p, k)$. Thus, *all of the positives with representatives* can be extracted from $\mathbf{o}$ by searching for all $x \in [j]$ with the property that $|\{y : \mathbf{o}_{xy} = 1\}| = k - 1$. Hence, the expected number of positives identified using $\delta_z(j, n, 2, k)$ in Algorithm 1 is equal to the expected number of members of $\{K_1, \ldots, K_p\}$ with representatives.

Fix j and let $\omega = (K_1, \ldots, K_p)$ be a randomly chosen ordered p-family of distinct k-sets in $[n]$. Clearly the expected number of k-sets with representatives in an ordered p-family of distinct k-sets is equal to the expected number of k-sets with representatives in an unordered p-family. Let X_i be the random variable that sends ω to 1 if and only if K_i has a representative in $[j]$. Then the expected value of X_i is $\phi(j, n, p, k)$. If we let $X = \sum_{i=1}^{p} X_i$, then X gives the number of coordinates of ω with a representative. The desired result follows from the additivity of expectation. ∎

Example 3.3. The simplicity of the decoding procedure can be demonstrated by displaying a modified output vector. Consider using $\delta_z(4, 7, 2, 3)$ when there are four positives which are represented by $P = \{\{1, 2, 3\}, \{3, 4, 5\}, \{3, 5, 7\}, \{5, 6, 7\}\}$. Here, 1 and 2 represent $\{1, 2, 3\}$, 4 and 6 are the sole representatives of $\{3, 4, 5\}$ and $\{5, 6, 7\}$ respectively, and $\{3, 5, 7\}$ doesn't have a representative. Because $\mathbf{o}_{xy} = \mathbf{o}_{yx}$, the output vector $\mathbf{o} = (\mathbf{o}_{12}, \mathbf{o}_{13}, \mathbf{o}_{14}, \mathbf{o}_{15}, \mathbf{o}_{16}, \mathbf{o}_{17}, \mathbf{o}_{23}, \mathbf{o}_{24}, \mathbf{o}_{25}, \mathbf{o}_{26}, \mathbf{o}_{27}, \mathbf{o}_{34}, \mathbf{o}_{35}, \mathbf{o}_{36}, \mathbf{o}_{37}, \mathbf{o}_{45}, \mathbf{o}_{46}, \mathbf{o}_{47}) = 110000100001101100$ can be displayed as

1						2					
$\mathbf{o}_{12}$	$\mathbf{o}_{13}$	$\mathbf{o}_{14}$	$\mathbf{o}_{15}$	$\mathbf{o}_{16}$	$\mathbf{o}_{17}$	$\mathbf{o}_{21}$	$\mathbf{o}_{23}$	$\mathbf{o}_{24}$	$\mathbf{o}_{25}$	$\mathbf{o}_{26}$	$\mathbf{o}_{27}$
1	1	0	0	0	0	1	1	0	0	0	0
3						4					
$\mathbf{o}_{31}$	$\mathbf{o}_{32}$	$\mathbf{o}_{34}$	$\mathbf{o}_{35}$	$\mathbf{o}_{36}$	$\mathbf{o}_{37}$	$\mathbf{o}_{41}$	$\mathbf{o}_{42}$	$\mathbf{o}_{43}$	$\mathbf{o}_{45}$	$\mathbf{o}_{46}$	$\mathbf{o}_{47}$
1	1	1	1	0	1	0	0	1	1	0	0

Now the sets $\{x\} \cup \{y : \mathbf{o}_{xy} = 1\}$ with $|\{y : \mathbf{o}_{xy} = 1\}| = k - 1$ are easy to locate. In this case, $j = 4, n = 7$, and $k = 3$, so we divide the entries of the augmented output vector into four groups of six (one for each element of $[j]$) and we look among those groupings for those with two 1s. In our example, the first, second, and fourth groups of six indicate that $\{1, 2, 3\}$, $\{2, 1, 3\}$ and $\{4, 3, 5\}$ are positive. Hence, two of the four positives are identified. Since $\{1, 2, 3\}$ has two representatives, it is identified twice. The positive $\{3, 5, 7\}$ is not identified because it does not have a representative, and the positive $\{5, 6, 7\}$ is not identified because its representative is not in [4]. For general $\delta_z(j, n, 2, k)$, we can display the output vector in a similar fashion by repeating the value of $\mathbf{o}_{xy}$ when $\mathbf{o}_{yx}$ is required. Then the positives can be identified by looking among the k-groupings indexed by $[j]$ for those with exactly $k - 1$ 1s. Note that each k-grouping has at least $k - 1$ 1s and that regardless of the values of p and k, the decoding complexity of Algorithm 1 is $\binom{j}{2}n$ because $\mathbf{o}_{xy} = \mathbf{o}_{yx}$. Most other nonadaptive algorithms have a decoding complexity equal to the size of the population. This heuristic will be useful when we consider how testing errors affect the identification of the positives.

For simplicity and practicality rather than necessity, we approximate the value of $\phi(j, n, p, k)$.

Proposition 3.4. *Let $k < j < n$, then*

$$\phi(j, n, p, k) \approx \binom{n}{k}^{-p} \sum_{i=1}^{k} (-1)^{i+1} \binom{j}{i} \binom{n-i}{k-i} \binom{n-i}{k}^{p-1}.$$

Proof. Let $\lambda = (K_1, \ldots, K_p)$ be a randomly chosen ordered p-family of k-sets (with repetition allowed) in $[n]$ and let E_1 be the event that K_1 has a representative in $[j]$. We can compute $\phi'(j, n, p, k) = \text{prob}(E_1)$ by enumerating the number of families λ in E_1. For each $x \in [j]$, let $\Omega(x) = \{\lambda : x \text{ is a representative of } K_1\}$. Then $E_1 = \cup_{x \in [j]} \Omega(x)$. By inclusion-exclusion, we have that

$$|E_1| = \sum_{i=1}^{k} (-1)^{i+1} \binom{j}{i} \binom{n-i}{k-i} \binom{n-i}{k}^{p-1}.$$

Since $\phi'(j, n, p, k) \approx \phi(j, n, p, k)$, the result follows. ∎

4. Applying the Main Result

To apply this result to a population of cardinality z that contains at most p positives, we must choose j, n, and k so that $\binom{n}{k} \geq z$, $p \cdot \phi(j, n, p, k)$ is at the desired level, and $\frac{k(k-1)}{n(n-1)} z$ is an upper bound for our pool size. It is easy to see that, for fixed values of n and p, the value of k that maximizes $\phi(n, n, p, k)$ is the same value of k that maximizes $\phi(j, n, p, k)$ when $j < n$ is also fixed. For almost all values of n, k, and p, we have the following sequence of successive approximations to $\phi(n, n, p, k)$:

$$\phi(n, n, p, k) \approx \sum_{i=1}^{k} (-1)^{i+1} \binom{k}{i} \left(\frac{\binom{n-i}{k}}{\binom{n}{k}} \right)^{p-1} \approx \sum_{i=1}^{k} (-1)^{i+1} \binom{k}{i} \left(\frac{n-i}{n} \right)^{k(p-1)}$$

$$\approx \sum_{i=1}^{k} (-1)^{i+1} \binom{k}{i} e^{\frac{-ki(p-1)}{n}} = 1 - \left(1 - e^{\frac{-k(p-1)}{n}}\right)^{k}.$$

The latter estimate is more lucid. More importantly, for fixed parameter values of n and p, $1 - \left(1 - e^{\frac{-k(p-1)}{n}}\right)^{k}$ attains its maximum value when $k = \frac{n \ln(2)}{p-1}$. Thus, given a desired value ϕ_0 of $\phi(j, n, p, k)$ when p is fixed, initially we choose $n = n_0$ and $k = k_0$ in $\delta_z(j, n, 2, k)$ with $\frac{n_0}{k_0}$ close to $\frac{p-1}{\ln(2)}$, $\binom{n_0}{k_0} \geq z$, $\phi(n_0, n_0, p, k_0) \geq \phi_0$ and n_0 as small as possible. From these initial choices, adjustments n_1, k_1 to n_0, k_0 respectively are made keeping n_1 as small as possible, $\binom{n_1}{k_1} \geq z$, $\phi(n_1, n_1, p, k_1) \geq \phi_0$, and making $\frac{k_1(k_1-1)}{n_1(n_1-1)} z$ less than the desired pool size. It is easy to see that for fixed values of k and p, $\phi(n, n, p, k)$ is an increasing function of n, and for fixed values of n and p, $\phi(n, n, p, k)$ decreases as k moves away from $\frac{n \ln(2)}{p-1}$. Finally, $j_0 \leq n_1$ is selected so that $\phi(j_0, n_1, p, k_1) \geq \phi_0$.

Example 4.1. Suppose a population has $z = 10^5$ objects, $p = 5$, the pool size must be less than 2500, and we want to identify 90% *of the positives on average,* i.e., $\phi_0 = 0.9$. Since $\frac{p-1}{\ln(2)} \approx 5.8$, we initially choose $n_0 = 29$ and $k_0 = 5$. This gives $\binom{n_0}{k_0} \geq z$, $\phi(n_0, n_0, p, k_0) = .97$ and $\frac{k_0(k_0-1)}{n_0(n_0-1)}z = 2463$. Selecting $j_0 = 22$ gives $\phi(j_0, n_0, p, k_0) = .902$. Since $\delta_z(j_0, n_0, 2, k_0)$ has 385 rows, we need 385 pools. If the pool size needs to be approximately 1500, then we choose $n_1 = 37$, $k_1 = 5$, and $j_0 = 24$.This gives $\binom{n_1}{k_1} \geq z$, $\phi(j_0, n_1, p, k_1) = .905$ and $\frac{k_1(k_1-1)}{n_1(n_1-1)}z = 1501$. Since $\delta_z(j_0, n_1, 2, k_1)$ has 588 rows, we need 588 pools.

5. The Nonadaptive Algorithm with Testing Errors

In this section, we assume errors may occur only in the testing and not in the formation of the pools. As in Section 3, given a population of cardinality $z \leq \binom{n}{k}$, let P represent the positives and suppose $|P| = p$. Let $k < n$. Identify the population with a random z-set of columns of $\delta_z(n, n, 2, k)$. Then P is represented by a p-family $\{K_1, \ldots, K_p\}$ of k-sets of $[n]$. Test each row of $\delta_z(n, n, 2, k)$. If pool γ_{xy} contains a positive element but is mislabeled by the test result as a negative pool or viceversa, then we say that $\mathbf{o}_{xy}$ is an *outcome error.*

When using $\delta_z(n, n, 2, k)$, how do outcome errors affect the efficacy of Algorithm 1? If outcome errors occur, then we can no longer be sure that all of the objects identified by Algorithm 1 will be positive. Instead, the objects identified in the decoding part of the algorithm will be *candidate positives.* Let CP be the set of candidate positives. It is easy to see that $|CP| \leq n$. Recall the augmented output vector in Example 3.3. Algorithm 1 identifies the sets $\{x\} \cup \{y : \mathbf{o}_{xy} = 1\}$ with $|\{y : \mathbf{o}_{xy} = 1\}| = k - 1$ as CP. These sets are identified by looking for the k-groupings that have exactly $k - 1$ 1's. Thus, it is straightforward to see that the way in which outcome errors affect $|CP \cap P|$ is by changing the number of 1's in a k-grouping in the modified output vector whose index value is a representative of a positive element. Thus, if we to quantify the number of k-groupings unaffected by a fixed number of outcome errors, then we can compute the expected value of $|CP \cap P|$.

Definition 5.1. *Let $k < n$. When using $\delta_z(n, n, 2, k)$ in Algorithm 1 for a population with cardinality z with $|P| = p$, if e outcome errors occur, let $\gamma(n, p, k, e)$ denote the expected value of $|CP \cap P|$.*

Theorem 5.2.

$$\gamma(n, p, k, e) \approx p \cdot \binom{n}{2}^{-e} \sum_{y=2}^{\min(n,2e)} \sum_{i=0}^{y} (-1)^i \binom{n}{y} \binom{y}{i} \binom{y-i}{2}^{e} \phi(n-y, n, p, k).$$

Proof. Suppose P is represented by $\{K_1, \ldots, K_p\}$. Since each pool in $\delta_z(n, n, 2, k)$ is represented by a 2-set in $\binom{[n]}{2}$, if e random outcome errors occur, then the number of unaffected k-groupings will be equal to the number of elements in $[n]$ that are not in the union of the e 2-sets that correspond to the pools in which the outcome errors occurred. In other words, if J is the set of elements of $[n]$ not in the union of the e 2-sets that

correspond to the pools in which the outcome errors occurred, then the probability that K_1 will still be identified by Algorithm 1 is $\phi(j, n, p, k)$ when $|J| = j$. Thus if $Y(e)$ is the random variable that gives the cardinality of the union of e 2-sets of $[n]$ (with repetition allowed), then the expected value of $|CP \cap P|$ is approximately equal to the expected value $p \cdot \phi(n - Y(e), n, p, k)$. By an inclusion-exclusion argument, we have for $2 \leq y \leq 2e$ that

$$\text{Prob}(Y(e) = y) = \binom{n}{2}^{-e} \binom{n}{y} \sum_{i=0}^{\min(n,2e)} (-1)^i \binom{y}{i} \binom{y-i}{2}^e.$$

From this, the result follows. ∎

Example 5.3. Suppose a population has $z = 10^5$ objects, $p = 5$, and the pool size must be less than 2500. Assume the outcome error rates are 1% , 3%, and 5%, and let e_1, e_2, and e_3 denote the actual number of errors in each case, respectively. As in Example 4.1, we choose $n_0 = 29$ and $k_0 = 5$. This gives $\binom{n_0}{k_0} \geq z$, and $\frac{k_0(k_0-1)}{n_0(n_0-1)} z = 2463$. If we use $\delta_z(n_0, n_0, 2, k_0)$ and assume an outcome error rate of 1%, then $e_1 = \lceil .01\binom{n_0}{2} \rceil$ and $\gamma(n_0, p, k_0, e_1) = 4.39$. Since $\delta_z(n_0, n_0, 2, k_0)$ has 406 rows, using those 406 pools in Algorithm 1 will give $|CP| \leq 29$ and the expected value of $|CP \cap P| = 4.39$. Assuming an outcome error rates of 3% and 5%, then $e_2 = \lceil .03\binom{n_0}{2} \rceil$, $e_3 = \lceil .05\binom{n_0}{2} \rceil$ and $\gamma(n_0, p, k_0, e_2) = 3.26$, $\gamma(n_0, p, k_0, e_3) = 2.16$ respectively.

Acknowledgment. The author thanks John Spouge of NIH for his helpful suggestions concerning some of the technical aspects of this paper.

References

1. D.J. Balding and D.C. Torney, Optimal pooling designs with error detection, J. Combin. Theory, Ser. A **74** (1996).
2. D.J. Balding et al., A comparative survey of non-adaptive pooling designs, In: Genetic Mapping and DNA Sequencing, IMA Volumes in Mathematics and its Applications, Springer Verlag, 1995, pp. 133–155.
3. W.J. Bruno et al., Design of efficient pooling experiments, Genomics **26** (1995) 21–30.
4. R. Dorfman, The detection of defective members of a large population, Ann. Math. Stat. **14** (1943) 436–440.
5. D-Z. Du and F.K. Hwang, Combinatorial Group Testing and Its Applications, World Scientific, Singapore, 1993.
6. A. D'yachkov, A. Macula, and V. Rykov, On optimal parameters of a class of superimposed codes and designs, submitted, 1997.
7. A. D'yachkov and V. Rykov, Superimposed distance codes , Problems Contr. and Inf. Theory **18** (1989) 237–250.
8. Farach et al., Group testing problems with sequences experimental molecular biology, In: Proceedings of Compression and Complexity of Sequences, 1997, B. Carpentieri et al., Eds., IEEE Press, 1994, pp. 357–367.
9. E. Knill, Lower bounds for identifying subset members with subset queries, In: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, Association for Computing Machinery and Society for Industrial and Applied Mathematics, 1995, pp. 369–377.

10. E. Knill et al., Non-adaptive group testing in the presence of errors, Discrete Appl. Math. **88** (1998) 261–290.
11. A. Macula, A simple construction of d-disjunct matrices with certain constant weights, Discrete Math. **162** (1996) 311–312.
12. A. Macula, Nonadaptive group testing with error-correcting d^e–disjunct matrices, Discrete Appl. Math. **80** (1997) 217–282.
13. A. Macula, Probabilistic nonadaptive and two-stage group testing with relatively small pools and DNA library screening, J. Combin. Optimization, to appear.