2-Stage Fault Tolerant Interval Group Testing

Ferdinando Cicalese[1] and José Augusto Amgarten Quitzau[1,2]

[1] AG Genominformatik, Technical Faculty, Bielefeld University, Germany
[2] International Graduate School in Bioinformatics and Genome Research, Center for Biotechnology, Bielefeld University, Germany

Abstract. We study the following fault tolerant variant of the interval group testing model: Given four positive integers n, p, s, e, determine the minimum number of questions needed to identify a (possibly empty) set $P \subseteq \{1, 2, \ldots, n\}$ ($|P| \leq p$), under the following constraints. Questions have the form "Is $I \cap P \neq \emptyset$?", where I can be any interval in $\{1, 2 \ldots, n\}$. Questions are to be organized in s batches of non-adaptive questions (stages), i.e, questions in a given batch can be formulated relying only on the information gathered with the answers to the questions in the previous batches. Up to e of the answers can be erroneous or lies.

The study of interval group testing is motivated by several applications. remarkably, to the problem of identifying splice sites in a genome. In particular, such application motivates to focus algorithms that are fault tolerant to some degree and organize the questions in few stages, i.e., on the cases when s is small, typically not larger than 2. To the best of our knowledge, we are the first to consider fault tolerant strategies for interval group testing.

We completely characterize the fully non-adaptive situation and provide tight bounds for the case of two batch strategies. Our bounds only differ by a factor of $\sqrt{11/10}$ for the case $p = 1$ and at most 2 in the general case.

1 Introduction

Problem Statement. In this paper we consider fault tolerant algorithms for interval group testing. An instance of the problem is given by four non-negative integers n, p, s, e and a subset $P \subseteq O = \{1, 2, \ldots, n\}$, such that $|P| \leq p$. The set O is the search space and P is the set of *positive* objects that have to be identified. Queries are binary test asking "Is $P \cap \{i, i+1, \ldots, j\} \neq \emptyset$?", for some $1 \leq i \leq j \leq n$. The target is to identify P by using the minimum possible number of queries. We assume that tests are arranged in *stages*: in each stage a certain number of tests is performed non-adaptively, while tests of a given stage can be determined depending on the outcomes of the tests in all previous stages. Finally, we assume that up to a finite number e of the answers might be erroneous or lies.

For each value of the parameters n, p, s, e we want to determine $\mathcal{N}(n, p, s, e)$, the worst-case number of tests that are necessary (and sufficient) to successfully identify all positives in a search space of cardinality n, under the hypothesis that

T. Tokuyama (Ed.): ISAAC 2007, LNCS 4835, pp. 858–868, 2007.

the number of positives is at most p, s-stage algorithms are used and up to e answers are lies.

Motivations and Related Research. Group testing is a basic paradigm in the theory of combinatorial search and is efficiently used in very diverse areas such as quality control, multiple access communication, computational molecular biology, data compression, and data streams algorithms among the others (see [5,6,9,14,17,3]). Group testing with interval tests also arises in variety of domains, e.g., detecting holes in a gas pipe [5,4], finding faulty links in an electrical or communication network, data gathering in sensor networks [10,11,12], just to mention a few.

Our main motivation for the study of interval group testing comes from its application to the problem of determining exon-intron boundaries within a gene [15,18]. In a very simplified model, a gene is a collection of disjoint substrings within a long string representing the DNA molecule. These substrings, called *exons*, are separated by substrings called *introns*. The boundary point between an exon and an intron is called a *splice site*, because introns are spliced out between transcription and translation. Determining the splice sites is an important task, e.g., when searching for mutations associated with a gene responsible for a disease.

In [18], a new experimental protocol is proposed that searches for the exons boundaries using group testing. This consists of selecting two positions in the cDNA, a copy of the original genomic DNA from which introns have been spliced out, and determine whether they are at the same distance as they were in the original genomic DNA string. If these distances do not coincide then at least one intron (and hence a splice site) must be present in the genomic DNA between the two selected positions. The formulation of splice sites identification as a group testing problem with interval queries is explicitly stated in [13,15,18]. The advantages of splice site detection by distance measurements over sequence-based methods using, e.g., Hidden Markov Models are that this method works without expensive sequencing of genomic DNA and it gives the results directly from experiments, without relying on inference rules. The work [18] and the book [15] report about the experimental evaluation, on real data, of the algorithm ExonPCR, that finds exon-intron boundaries within a gene. The authors of [18] give also a simple asymptotic analysis of their $\Theta(\log n)$-stage algorithm. The question was whether there exist less obvious but more efficient query strategies for Interval Group Testing, and more importantly, algorithms able to cope with the technical limitation of the experiments, and particularly with errors. We remark that non-adaptive strategies are desirable in this context, in order to avoid long waiting periods necessary to prepare each experiment. However a totally non-adaptive algorithm (with $s = 1$) needs unreasonably many queries. Thus, the necessity arises to trade more stages for fewer queries, but without exceeding with stages. In [1] the first rigorous algorithmic study of the problem was presented, and for the case $s \leq 2$ a precise evaluation of $\mathcal{N}(n, p, s, 0)$ was given. In [2] a sharper asymptotic estimation of $N(n, p, s, 0)$ was given that is optimal up to the constant of the main term in the case of large s.

The necessity of dealing with errors in the tests had been already stated in the seminal papers [15,18] and reaffirmed in the subsequent ones. However, to the best of our knowledge, ours are the first non trivial results on this interesting variant of the problem.

Our Results. We focus on strategies that use adaptiveness at most once, i.e., strategies with questions organized in one or two batches of non adaptive queries ($s \in \{1, 2\}$). In fact, according to [7] *... the technicians who implement the pooling strategies generally dislike even the* 3*-stage strategies that are often used [...]. The pools are either tested all at once or in a small number of stages (usually at most 2).* We exactly determine $N(n, p, 1, e)$ and provide very tight bounds for the $N(n, p, 2, e)$ that in the case $p = 1$ at most differ by a factor of $\sqrt{11/10}$, and at most by a factor 2 in all the other cases. We remark that these are the first non trivial results on fault tolerant interval group testing procedures and we stress the necessity to drive attention onto the fault tolerant variant of interval group testing.

2 Definitions and Notation

In this section we fix the notation used in the text. The set of objects where we try to find the positives is the set of the first n non-negative integers $[n] = \{1, 2, \ldots, n\}$. By abuse of notation we shall use square brackets to denote intervals of integers in $[n]$. Then, for each $1 \leq i \leq j \leq n$, we shall use $[i, j]$ to denote the set $\{i, i + 1, \ldots, j\}$. Given an interval $\pi = [i, j]$, we shall denote its size by $|\pi|$, i.e., $|\pi| = j - i + 1$. By definition each query asks about the intersection of a given interval with the set of positive elements. Therefore, we shall identify a query with the interval it specifies. We say that a query $Q \equiv [i, j]$ covers an element $k \in [n]$ if $k \in [i, j]$.

A query $Q \equiv [i, j]$ has two boundaries: the left, $(i-1, i)$, and the right, $(j, j+1)$. For the sake of definiteness, we assume that, for any a, a the query $[1, a]$ has left boundary $(0, 1)$, and the query $[a, n]$ has right boundary $(n, n+1)$. A multiset of queries $\mathcal{Q}$ defines a set of boundaries $\mathcal{B}(\mathcal{Q}) = \{(i_1, i_1 + 1), (i_2, i_2 + 1), \ldots\}$, where $i_k < i_{k+1}$. Every interval $[i_k + 1, i_{k+1}]$ is called a *piece*. Because every query has two distinct boundaries, but two queries may share some boundaries, we have $|\mathcal{B}(\mathcal{Q})| \leq 2\,|\mathcal{Q}|$. A boundary B of a piece P is said to be *turned to* the piece if there is a query Q such that $P \subset Q$ and B is also a boundary of Q. A piece is called a *2-piece* if both its boundaries are turned to it. A piece that has only one of it boundaries turned to it is called a *1-piece*. If none of the boundaries of a piece are turned to it, the piece is called a *0-piece*. Figure 1 illustrate the definitions given so far.

We shall also use the definition of a *YES set*. Given a multiset of queries $\mathcal{Q}$, a *YES set* (for $\mathcal{Q}$) is a subset of $\mathcal{Q}$ such that there exists a set of positives P ($|P| \leq p$) such that answering YES to queries in the *YES set* and NO to the other queries, the answers are consistent with P, but for at most e lies. A *YES set* is basically a possible (legal) strategy for the adversary, given the set of questions $\mathcal{Q}$. A YES set is called *specific* if the intersection of all its queries

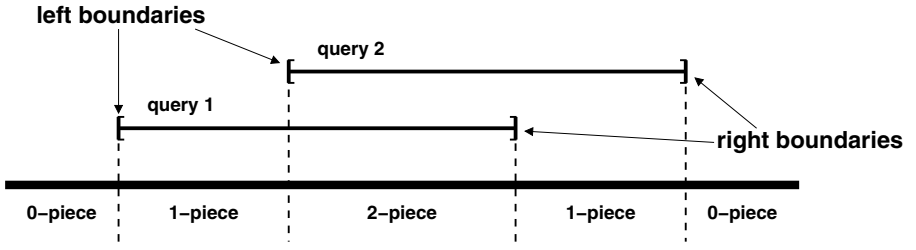

Fig. 1. A set of two interval queries, which partition the set of objects into 5 pieces. The thicker line represents the set of objects.

corresponds to a single piece, and the piece has at most one positive, otherwise it is called *unspecific*. More formally, a YES set $\mathcal{Y} \subset \mathcal{Q}$ is specific if and only if there is a piece π of $\mathcal{Q}$, with $|\pi \cap P| \leq 1$, such that $\bigcap_{Q \in \mathcal{Y}} Q = \pi$.

3 Non-adaptative Interval Group Testing with One Error

We start our analysis with the case of 1 stage strategies. In fact, the results in this section will be the basis for the analysis of the more practical two batch case. The following two theorems completely characterize 1-stage e-fault tolerant interval group testing.

Theorem 1. *For all $n \geq 1$ and $e \geq 0$, it holds that* $\mathcal{N}(n,1,1,e) = \left\lceil \frac{(2e+1)(n+1)}{2} \right\rceil$.

Proof. The lower bound directly follows from the following claim.
Claim. Every strategy that correctly identifies the (only) positive or reports $P = \emptyset$, uses a set of questions such that there are at least $2e+1$ questions' boundaries $(i, i+1)$ for each $i = 0, 1, \ldots, n$.

By contradiction, let us consider a strategy such that for some $i \in [n]$ there are $b \leq 2e$ questions with a boundary $(i, i+1)$. Let $\mathcal{Q}$ be the set of such questions and $\mathcal{Q}_1$ the set of all questions in $\mathcal{Q}$ which contain i. Assume, without loss of generality, that $|\mathcal{Q}_1| \geq |\mathcal{Q} \setminus \mathcal{Q}_1|$.

Let the adversary answer i) NO to all the questions having empty intersection with $\{i, i+1\}$, ii) YES to all questions including both i and $i+1$, iii) YES to exactly $\lceil |\frac{\mathcal{Q}}{2}| \rceil$ questions in $\mathcal{Q}_1$ and NO to the remaining ones in $\mathcal{Q}_1$, iv) answers YES to all the questions in $\mathcal{Q} \setminus \mathcal{Q}_1$.

A moment reflection shows that, due to the possibility of having up to e erroneous answers, the above set of answers is consistent with the both cases when $P = \{i\}$ and $P = \{i+1\}$[1].Hence, the given strategy cannot correctly discriminate among the above possibilities. The claim is proved.

Therefore, any strategy that is able to correctly identify P must use in total at least $(2e+1)(n+1)$ boundaries. Then, the desired results follows by observing that each question can cover at most 2 boundaries.

[1] In particular, for the cases, $i = 0$ (respectively $i = n$) the ambiguity is whether P contains no elements or the element is 1 (resp. n).

We now turn to the upper bound. Direct inspection shows that for $n \leq 3$ there exists an easy strategy with the desired number of questions.

For each $k \geq 2$, let $\mathcal{A}_{2k+1} = \{[1,2],[2,4],[4,6],\ldots,[2k-2,2k],[2k,2k+1]\}$ and $\mathcal{A}^1_{2k} = \{[2,2k-1],[3,2k-2],\ldots,[k,k+1]\}$, $\mathcal{A}^2_{2k} = \{[1,k],[k+1,2k]\}$, and $\mathcal{A}^3_{2k} = \{[1,k]\}$.

Then, for $n \geq 4$, the following strategy attains the desired bound.

If n is odd, the strategy consists of asking $2e+1$ times the questions in $\mathcal{A}_n$. These amount to $(2e+1)\lceil (n+1)/2 \rceil = \lceil (2e+1)(n+1)/2 \rceil$ questions which clearly cover $2e+1$ times each boundary $(i, i+1)$, for each $i = 0, 1, \ldots, n$.

If n is even, let $k = n/2$. Now, the strategy consists of asking $2e+1$ times the questions in $\mathcal{A}^1_n$, plus $e+1$ times the questions in $\mathcal{A}^2_n$, plus e times the questions in $\mathcal{A}^3_n$. In total, in this case, the strategy uses $(2e+1)(k-1) + 2(e+1) + e = (2e+1)k + e + 1 = \lceil (2e+1)(2k+1)/2 \rceil = \lceil (2e+1)(n+1)/2 \rceil$, as desired.

For the case of more positives we have the following generalization.

Theorem 2. *For all integers $n \geq 1, p \geq 2, e \geq 0$, it holds that $\mathcal{N}(n,p,1,e) = (2e+1)n$*

Proof. The upper bound is trivially obtained by a strategy made of $(2e+1)$ copies of the singleton questions $\{1\},\{2\},\ldots,\{n\}$.

The lower bound is obtained proceeding in a way analogous to the argument used in the previous theorem. Here, we argue that every strategy that correctly identifies P must ask, for each $i = 1, 2, \ldots, n-1$, at least $2e+1$ questions with boundary $(i, i+1)$ and including i, and at least $2e+1$ questions with boundary $(i, i+1)$ and including $i+1$. Moreover, it must ask at least $2e+1$ questions with boundary $(0,1)$ and $2e+1$ questions with boundary $(n, n+1)$. For otherwise, assume that there exists $i \in \{1,2,\ldots,n-1\}$, such that one of the above $4e+2$ boundaries $(i, i+1)$ is missing. Proceeding as in the proof of the previous theorem, it is possible to define an answering strategy for the adversary that balances the answers on the two sides of the boundary in so that with the information provided by the answers and given the possible number of lies, it is not possible to discriminate between the case $P = \{i\}$ and the case $P = \{i, i+1\}$, or between the case $P = \{i+1\}$ and the case $P = \{i, i+1\}$. Alternatively, if some of the above boundaries $(0,1)$ (resp. $(n, n+1)$) are missing, the adversary can answer in such a way that it is not possible to discriminate between the case $P = \emptyset$ and $P = \{1\}$ (resp. $P = \{n\}$).

4 Bounds for Two-Stage Strategies with One Positive

The aim of this section is to prove asymptotically tight upper and lower bounds on the query number of 2-stage interval group testing algorithms when up to one of the answers is a lie. We shall first analyze the case when P contains *at most one positives.*

We start with some notations and facts which will be used for the proof of the lower bound.

Let $\mathcal{Q}$ be a set of interval questions. For any piece π, cut by $\mathcal{Q}$, we denote by $N(\pi)$ the set of query intervals in $\mathcal{Q}$ containing π.

Let $\pi_1, \ldots, \pi_\ell$ be the pieces determined by the intervals of $\mathcal{Q}$. Given the YES set Y, we define the *weight* it assigns to the piece π_i's according to the following scheme:

- A piece gets weight 1/2 if it can contain a positive and there will not be a lie in the next stage.
- A piece gets weight 3/2 if it can contain a positive and there might be still a lie in the next stage.

Here, "can" means that this possibility is consistent with the YES set.

We denote with $w^Y(\mathcal{Q})$ the weighted sum of the lengths of the pieces cut by $\mathcal{Q}$ weighted according to the weighted associated to Y. In formulas, if w_j is the weight given to the piece π_j, we have $w^Y(\mathcal{Q}) = \sum_{j=1}^{\ell} |\pi_j| w_j$.

Assume now that $\mathcal{Q}$ is the set of interval questions asked in the first stage of a two stage group testing algorithm which finds more than one positive. Using Theorems 1 and 2 it follows that if Y is the set of intervals in $\mathcal{Q}$ that answer YES, the number of queries to be asked in the second stage in order to find all the positives is *at least* $w^Y(\mathcal{Q})$. Since each piece π_j that may have a positive must be solved as an independent interval group testing problem with universe of size $|\pi_j|$ at the second stage, and w_j associates the correct lower bound factor given by Theorems 1 and 2 in the case of one error.
In order to prove the promised bound we will show that for each possible set of interval questions $\mathcal{A}_1$ there exists a yes set Y such that $w^Y(\mathcal{A}_1) \geq n/|\mathcal{A}_1|$.

The following proposition allows us to limit the analysis for the lower bound to a subset of all possible first stages.

Proposition 1. *[1] Let $\mathcal{Q}$ be a set of interval questions producing a partition of the search space in which there are pieces a and b such that $N^{\mathcal{Q}}(a) = N^{\mathcal{Q}}(b)$. Then, there exists a set of interval question $\mathcal{Q}'$ of the same cardinality of $\mathcal{Q}$ such that the following two conditions hold: (i) for each two pieces a' and b' in the partition produced by $\mathcal{Q}'$ it holds $N^{\mathcal{Q}'}(a') \neq N^{\mathcal{Q}'}(b')$; (ii) for each* YES *set Y' for $\mathcal{Q}'$ there exists a* YES *set for $\mathcal{Q}$ such that $w^{Y'}(\mathcal{Q}') = w^Y(\mathcal{Q})$.*

After these preliminaries we can start the proof of the lower bound. Let $\mathcal{Q}$ be the set of questions asked in the first stage by a two stage interval group testing algorithm. Let $q = |\mathcal{Q}|$ In virtue of Proposition 1 we can assume that for each two pieces π_1 and π_2 determined by $\mathcal{Q}$ it holds that $N(\pi_1) \neq N(\pi_2)$. We also have that the total number ℓ of pieces is at most $2q$, since the number of pieces covered by query intervals is at most $2q-1$ (by induction) and by Proposition 1, at most one piece π_o is outside all query intervals ($N(\pi_o) = \emptyset$).

The next technical lemma was proved in [1]. It uses an averaging argument to prove the existence of an adversary strategy that can force a certain number of questions in the second stage.

Lemma 1. *Consider a multiset of k (not necessarily distinct!)* YES *sets, and for each $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, \ell$, let w_{ij} be the weight of the jth piece*

in the YES *vector associated to the ith* YES *set. If there exist* $r > 0$ *such that for all* $j = 1, 2, \ldots, \ell$, *it holds that* $\sum_{i=1}^{k} w_{ij} \geq r$, *then an adversary can force at least* $\frac{r}{k}n$ *queries in the second stage.*

We adapt the bounds for the 2-stage strategy for 2 positives, given by Cicalese et al. [1], to the case where a 2-stage strategy for at most one positive may contain at most one error.

Lemma 2

$$\mathcal{N}(n, 1, 2, 1) \geq \sqrt{5n} - O(1).$$

Proof. We show that we may achieve $r \geq 2.5$ using at most $2t_1 + 2$ *YES set*'s, where t_1 is the number of queries in the first stage. Then, by Lemma 1, the number o queries we need is at least $\min\left(t_1 + \frac{5n}{4t_1+4}\right) = \sqrt{5n} - O(1)$.

To achieve $r \geq 2.5$, we create a *specific* YES set for each piece defined by the t_1 queries in the first stage. Recall that there are at most $2t_1$ distinct (according to question containment) pieces. This already guarantees $r \geq 1.5$. Moreover, each pair of adjacent pieces fall in one of the following cases, depending on how many queries separate them:

Case 1. Consider the case where two pieces are separated by the boundary of exactly one query. Let $(i, i+1)$ be such boundary. The YES created for the piece containing i assigns weight $1/2$ to the piece containing $i + 1$, since there is the chance that exactly the query having the boundary $(i, i + 1)$ was an error.

Therefore, by symmetry, each piece in a pair of neighbors separated by a single boundary automatically gets an extra weight $\frac{1}{2}$.

Case 2. When the pieces are separated by the boundary $(i, i + 1)$ of exactly two queries, the YES set created for one of them indicates precisely that piece as the one containing a positive. In these cases, we don't get the extra weight of $\frac{1}{2}$ for the neighbor. However, we can use the fact that, since there is no piece between these two boundaries, the number of pieces is at most $2t_1 - 1$, and so is the number of YES sets used so far. Therefore we may create an *unspecific* YES set involving both pieces. This is a YES set that answer yes to all queries including both pieces and answers the two questions with boundaries $(i, i + 1)$ inconsistently, i.e., one indicating the piece containing i and one indicating the piece containing $i + 1$. This gives us the desired extra weight $\frac{1}{2}$ to each piece.

Case 3. Using the same argument as in the previous case, if a pair of pieces is separated by more than 2 boundaries, then the number of pieces is at most $2t_1 - 2$. We may use two of this extra pieces to create a new specific YES set for each piece in the pair. At the end, each of the pieces gets an extra weight of $\frac{3}{2}$.

Therefore, we are able to extend the previously suggested multiset of YES sets in such a way that each piece gets extra weight $\frac{1}{2}$ from each of its neighbors. As a result, all the pieces, but the ones on the extremities, surely have sum of weights at least 2.5. For pieces on the extremities, creating two extra consistent YES sets, one for each, gives desired total weight. At the end, we have a multiset with the desired properties.

Lemma 3

$$\mathcal{N}(n, 1, 2, 1) \leq \sqrt{5.5n}.$$

Proof. We show a 2-stage query scheme which is able to find a positive in a set of n elements using at most $\sqrt{5.5n}$. The first stage consist in queries divided in two groups, as shown in Fig. 2:

Group A: Consists of t_A overlapping queries that divide the set of objects in $2t_A$ pieces of the same size.

Group B: Consists of rt_A overlapping queries, for $0 < r < 1$.

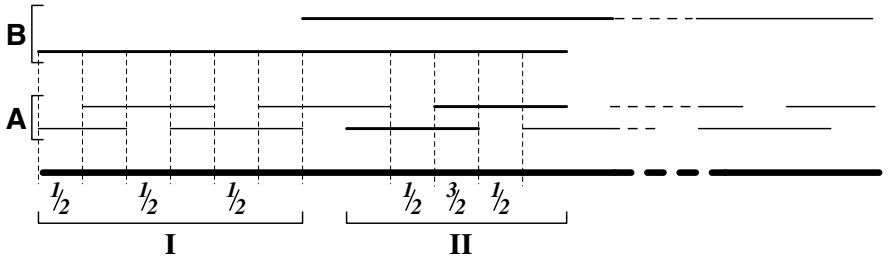

Fig. 2. Query scheme used in the first stage of a 2-stage strategy. The thicker line represents the set of objects, whereas all the other represent the a- and b-queries. In the figure we may see the two patterns that compete for the worst case. Darker lines indicate queries that answer YES.

An inspection of the possible YES sets gives two situations as candidates for the worst case:

I. When a single b-query answers YES correctly, one of the a-queries surely lies. In any case, since at most one error is allowed, the positive must be in one of the single-covered pieces. As a consequence, all such pieces covered by the non overlapping part of the b-query need to be checked in the second stage. Since the non-ovelapping part of the b-querie has size $\frac{n}{2rt_A}$, half of this piece is covered by single a-queries, and an error free strategy may be used in the second stage, the number of queries needed in the following stage is $\frac{n}{8rt_A}$.

II. When two overlapping a-queries answer YES, together with the corresponding b-queries, we must look for a positive in the piece corresponding to the overlapping part as if there was no error. We also need to consider the hypothesis that one of the a-queries gave the wrong answer. Therefore the two pieces corresponding to the non-overlapping parts must also be investigated. In the last case, we may take advantage of the fact that they are only possible in the presence of one error, and use the error-free strategy on the two pieces of size $\frac{n}{2t_A}$. This gives us a total of $\frac{5n}{4t_A}$ questions in the second stage.

The total number of queries used by this strategy is given by
$\min\left(t_A(1+r) + \max\left(\frac{5n}{4t_A}, \frac{n}{8rt_A}\right)\right)$.

By choosing $r = 0.1$, we equalize both worst case candidates and get $\min\left(1.1t_A + \frac{5n}{4t_A}\right) = \sqrt{5.5n}$.

4.1 More Positives

The following theorem summarizes our finding on two stage interval group testing with at most one error in the tests.

Theorem 3

$$\sqrt{6n(p-1)} - O(1) \leq N(n,p,2,1) \leq 2\sqrt{6n(p-1)}.$$

Proof. We start with the lower bound. Assume that the adversary accepts not to lie in the first phase. Moreover, she/he agrees to put the positives into the $p-1$ largest pieces defined by the first stage of queries.

Notice that this information, exchanged between the questioner and the adversary, can only make the situation better for the questioner.

Let q be the number of questions in the first phase. These questions divide the search space into at most $2q+1$ pieces. Hence, the largest $p-1$ of these pieces have total size at least $(p-1)n/(2q+1)$.

Since each of these pieces might contain up to 2 positives, by Theorem 2 the questioner has to ask at least 3 questions per element in each of these pieces.

So we have that the number of questions asked by an algorithm that uses q queries in the first stage is at least $q + 3(p-1)n/(2q+1)$.

Thus, minimizing over all possible values of q we have the desired bound.

In order to prove the upper bound we consider the following strategy, where q is a parameter to be decide later. In the first stage, we divide the search space into q non-overlapping intervals of equal size. We call them segments. Then we ask twice one question coinciding with each segment.

Let $\mathcal{A}$ be the set of segments such that the two corresponding questions are answered YES. Let $\mathcal{B}$ the set of segments whose corresponding questions are answered NO. Finally, let $\mathcal{C}$ the set of segments such that one of the corresponding questions is answered YES and one is answered NO.

Since we are assuming at most one error, trivially, no question is necessary in the second stage in each segment in $\mathcal{B}$.

We also have $|\mathcal{C}| \leq 1$.

We can now have the following cases.

Case 1. $|\mathcal{A}| \leq p-1$, $|\mathcal{C}| = 0$. Then, since each segment π might contain more than 1 positive, and the adversary might still lie, by Theorem 2, $3|\pi|$ questions have to be asked in π in the second stage. Since all segments are of the same size, in total we have $2q + 3|\mathcal{A}|n/q$ questions are asked in this case.

Case 2. $|\mathcal{A}| \leq p-1$, $|\mathcal{C}| = 1$. Again, each segment π might contain more than 1 positive. However, in this case the adversary has clearly already used a lie. Then, by Theorem 2, for each segment $\pi \in \mathcal{A}$, $|\pi|$ questions have to be asked in π in the second stage. Moreover, for the only segment γ in $\mathcal{C}$, either $|\gamma|/2$ or $3|\gamma|/2$

questions have to be asked, according as $\mathcal{A}$ contains $p-1$ or less segments. In fact, in the first case, γ can contain at most one positive, and Theorem 1 applies. Whilst in the second case, γ might contain more than one positive and then Theorem 1 applies. Since all segments are of the same size, in total we have $2q + 3(p-1)n/2q + n/2q$ questions in the first case and $2q + 3(|\mathcal{A}| + 1)n/2q$ in the second case ($|\mathcal{A}| \leq p-2$).

It is not hard to see that the worst situation for the questioner is given by *Case 1* with $|\mathcal{A}| = p-1$.

Thus, the above strategy uses in total at most $2q + 3(p-1)n/q$ questions. Minimizing with respect to q we have the desired bound.

References

1. Cicalese, F., Damaschke, P., Vaccaro, U.: Optimal group testing strategies with interval queries and their application to splice site detection. Int. Journal of Bioinformatics Research and Applications
2. Cicalese, F., Damaschke, P., Tansini, L., Werth, S.: Overlaps Help: Improved Bounds for Group Testing with Interval Queries. Discrete Applied Mathematics (to appear)
3. Cormode, G., Muthukrishnan, S.: What's hot and what's not: Tracking most frequent items dynamically. In: ACM Principles of Database Systems (2003)
4. Cox, L.A., Sun, X., Qiu, Y.: Optimal and Heuristic Search for a Hidden Object in one Dimension. In: Proc. of IEEE Conf. on System, Man, and Cybernetics, pp. 1252–1256 (1994)
5. Du, D.Z., Hwang, F.K.: Combinatorial Group Testing and its Applications. World Scientific, Singapore (2000)
6. Farach, M., Kannan, S., Knill, E.H., Muthukrishnan, S.: Group testing with sequences in experimental molecular biology. In: Carpentieri, B., De Santis, A., Vaccaro, U., Storer, J. (eds.) Proc. of Compression and Complexity of Sequences 1997, pp. 357–367. IEEE CS Press, Los Alamitos (1997)
7. Knill, E.: Lower Bounds for Identifying Subset Members with Subset Queries. In: Proceedings of Symposium on Discrete Algorithms 1995 (SODA 1995), pp. 369–377 (1995)
8. Gelfand, M., Mironov, A., Pevzner, P.A., Roytberg, M., Sze, S.H.: PROCRUSTES: Similarity-based gene recognition via spliced alignment, `http://www-hto.usc.edu/software/procrustes/`
9. Hong, E.H., Ladner, R.E.: Group testing for image compression. IEEE Transactions on Image Processing 11(8), 901–911 (2002)
10. Hong, Y.W., Scaglione, A.: On multiple access for distributed dependent sources: A content-based group testing approach. In: IEEE Information Theory Workshop ITW (2004)
11. Hong, Y.W., Scaglione, A.: Group testing for sensor networks: the value of asking the right answers. In: Asilomar Conference (2004)
12. Hong, Y.W., Scaglione, A.: Generalized group testing for retrieving distributed information. In: ICASSP (2005)
13. Karp, R.: ISIT 1998 Plenary Lecture Report: Variations on the theme of Twenty Questions. IEEE Information Theory Society Newsletter 49(1) (1999)

14. Ngo, H.Q., Du, D.Z.: A survey on combinatorial group testing algorithms with applications to DNA library screening. Discrete Mathematical Problems with Medical Applications. DIMACS Ser. Discrete Math. Theoret. Comput. Sci., Amer. Math. Soc. 55, 171–182 (2000)
15. Pevzner, P.A.: Computational Molecular Biology, An Algorithmic Approach. MIT Press, Cambridge (2000)
16. Sobel, M., Groll, P.A.: Group testing to eliminate efficiently all defectives in a binomial sample. The Bell Systems Technical Journal 38, 1179–1253 (1959)
17. Wolf, J.: Born again group testing: Multiaccess communications. IEEE Trans. Information Theory IT-31, 185–191 (1985)
18. Xu, G., Sze, S.H., Liu, C.P., Pevzner, P.A., Arnheim, N.: Gene hunting without sequencing genomic clones: Finding exon boundaries in cDNAs. Genomics 47, 171–179 (1998)