Adaptive Phenotype Testing for AND/OR Items

Francis Y.L. Chin Henry C.M. Leung S.M. Yiu

Department of Computer Science, The University of Hong Kong

{chin, cmleung2, smyiu}@cs.hku.hk

Abstract: The combinatorial group testing problem is concerned with the design of experiments so as to minimize the number of tests needed to find the sets of items responsible for a particular phenotype (an observable property). The traditional group testing problem only considers the OR problem, i.e. the phenotype appears as long as one of the responsible items exists. In this paper, we introduce the phenotype testing problem which is a generalization of the well-studied combinatorial group testing problem. In practice, there are more than one phenotype and the responsible items and their mechanism (AND or OR) for each phenotype are unknown where an AND mechanism means that the phenotype only appears if all of the responsible items exist.

This phenotype testing problem has an important application in biological research, known as phenotype knockout study. New algorithms for designing adaptive experiments for solving the phenotype testing problem with *n* items using O(log *n*) tests in the worst cases (the constant varies for different problem settings) are introduced. When the number of phenotypes is small, say at most 2, and the number of responsible items for each phenotype is at most 2, algorithms with near-optimal number of tests are presented.

1 Introduction

Phenotype knockout study [10, 13, 14] is a new and critical application in the study of biology. Phenotype refers to an observed physical characteristic of an organism (e.g. blue eyes, black hair, tumor). Many important phenotypes are induced by genes. Given a set of phenotypes and a set of genes, the problem is to determine which subset of genes induces which phenotype and the mechanism (AND or OR). Knowing which subset of genes and its mechanism for the phenotypes is useful in drug design such as tumor therapy. There are two simple mechanisms for a subset of genes to induce a phenotype, the *OR-mechanism* and the *AND-mechanism*. In the OR (AND)-mechanism, a subset of genes can induce a phenotype as long as one (all) of these genes is (are) active, in other words, the phenotype disappears if and only if all (any one of) these genes are (is) inactive. Besides AND and OR mechanisms, there are more complicated mechanisms between the responsible genes and phenotypes, which are not discussed in this paper [3].

To discover which subset of genes and its mechanism for each phenotype, biologists can knockout a gene [7,8] and observe if some particular phenotype still exists. However, such test is expensive and more importantly, it is very time consuming. Since the number of genes responsible for a phenotype is usually small, in practice, instead of checking one gene against each phenotype, we can check a subset of genes and several phenotypes together in one test. For example, for the OR-mechanism, if some phenotype disappears, we can conclude that none of the genes in the subset is responsible for their phenotype. Given n items and k phenotypes, the *phenotype testing problem* is to design how to group the items (genes) into subsets such that, based on the test results on these subsets, one can identify the *responsible items* (genes) for each phenotype and find out the OR- or AND-mechanism of these items (genes) to induce the phenotype. From now on, we shall use “genes” and “item” interchangeably as long as no confusion arises. The items that are responsible for the phenotype with the OR- (AND-) mechanism, called OR- (AND-) phenotype, are referred to as *OR-items* (*AND-items*). The objective is to have a design with as few subsets (or tests) as possible.

It turns out that this problem is related to the well-studied combinatorial group testing problem in computer science. In the *combinatorial group testing (CGT) problem* [1,2,5,9,11], we are given a set of items in which some of them are contaminated (or defective). We assume that a test can determine if a subset of items contains any contaminated ones. If the result is negative, all the items in the subset are not contaminated. An important objective of this problem is to design the grouping of the items into subsets in order to minimize the number of tests. The phenotype testing problem is a generalization of the CGT problem by considering more than one phenotype, each phenotype with different mechanisms and responsible items. One can easily see that the CGT problem is equivalent to the phenotype testing problem with only one OR-phenotype.

The CGT problem has been studied under two different scenarios, adaptive and non-adaptive. In the *adaptive* (or *sequential*) *scenario*, group tests are divided into stages, conducted one by one after the test results in previous stages are known. In the *non-adaptive scenario*, there is only one stage and all the group tests are performed together at the same time. It is assumed that after one simple stage of group tests, we should be able to discover all the responsible items in the non-adaptive scenario. Usually fewer tests are required under the adaptive scenario at the expense of more test stages. Non-adaptive tests will be conducted if time is more critical. In the phenotype testing problem, we can also consider these two scenarios. In fact, the non-adaptive phenotype testing problem with OR-mechanism has been studied in another application called *DNA library screening* [4,6,12] in bioinformatics, which is referred as *the pooling design* problem. In this paper, we will focus on the adaptive phenotype testing (APT) problem which minimizes the number of tests.

1.1 Our contributions:

For the adaptive case, existing solutions assume that there is only one phenotype. However, one test can produce test results for several phenotypes simultaneously and we shall show how to make use of this property to solve the phenotype testing problem. The difficulty of the phenotype testing problem arises from the fact that the subsets of items for testing might be different for each phenotype. For example, assume S_1 and S_2 are two disjoint sets of items, phenotypes P_1 is positive for S_1 and phenotypes P_2 is positive for S_2 respectively. For detecting the responsible items for P_1, splitting S_1 would be recommended, whereas splitting S_2 would be recommended for detecting the responsible items for P_2. Also, the mechanism of each phenotype is usually unknown. So the design of the tests for phenotype testing might be different from that for CGT when there are more than one phenotype.

In this paper, we first provide several algorithms to solve the APT problem for general k and d. Then, for some special cases, algorithms with near-optimal number of tests are presented. The number of tests required by our algorithms is given in the Table 1.

2 Preliminaries

Given a set of n items (represented by a set of integers $U = \{1, 2, \ldots, n\}$) and a phenotype P_r with a set U_r of responsible items, we define P_r as a function $2^U \rightarrow \{0,1\}$ such that for any subset S of U, $P_r(S) = 1$ if and only if $U_r \subseteq S$, where U_r is a set of AND-items for P_r (AND-mechanism) or $U_r \cap S \neq \emptyset$ where U_r is a set of OR-

# of phenotypes (k)	Max. # of responsible items per phenotype (d)	Number of tests required
General solution $k = 1$	$d \geq 1$	$d\lceil \log_2 n\rceil + (d + 2)$
$k \geq 2$	$d \geq 1$	$4\ d\lceil \sqrt{k/2}\ \rceil \cdot \lceil \log_2 n\rceil$
Special cases $k = 1$	$d = 2$	$2\lceil \log_2 n\rceil$
$k = 2$	$d = 1$	$\lceil \log_2 n\rceil$
$k = 2$	$d = 2$	$2\lceil \log_2 n\rceil + 2\lceil \sqrt{\log_2 n - 1}\ \rceil$

Table 1. Summary of our results

items for P_r (OR-mechanism). In practice, the size of U_r is small and bounded by a constant d, i.e. $|U_r| \leq d$.

Given a set of n integers $U = \{1, 2, \ldots, n\}$ and a set of phenotype P_r, $r = 1, \ldots, k$, each with a hidden set U_r of responsible items where $|U_r| \leq d$, the *Adaptive Phenotype Testing (APT) problem* is to design an algorithm for constructing the minimum number of subsets $S_1, S_2, \ldots, S_q$ of U for testing sequentially such that we can deduce U_r and its mechanism from $P_r(S_1)$, $P_r(S_2)$, $\ldots$, $P_r(S_q)$, for all $r = 1, \ldots, k$, where the k test results of $\{P_r(S_i)|\ r = 1, \ldots, k\}$ can be provided in a single test on subset S_i. Note that the construction of S_i might depend on the test results on $S_1, S_2, \ldots, S_{i-1}$. □

The following theorem shows that the APT problem with only AND-phenotypes is equivalent to the APT problem with only OR-phenotypes.

Theorem 1: The APT problem with only AND-phenotypes and the APT problem with only OR-phenotypes are equivalent.

Proof: We reduce the APT problem with only AND-phenotypes P_r to the APT problem with only OR-phenotypes P_r as follows. Assume that for a sequence of tests, the subsets of items $S_1, S_2, \ldots, S_i, \ldots$ are constructed by an algorithm for solving the APT problem with AND-phenotypes P_r. We can apply the same algorithm to solve the APT problem with OR-phenotypes P_r' by replacing S_i by U/S_i, i.e. the complement of S_i, and the test results of $P_r(S_i)$ by 1- P_r'(U/S_i). It is easy to show that $P_r(S_i) = 1$ iff all the AND-items are in S_i; alternatively, P_r'$(U/S_i) = 0$ iff all the OR-items are in S_i. Thus, the algorithm which constructs $S_1, S_2, \ldots, S_i$ for identifying the AND-items would be able to construct the sequence $U/S_1, U/S_2, \ldots, U/S_i$, … for identifying the same OR-items. It is obvious that the reduction in the other direction is the same. □

3 APT algorithms for general *k* and *d*

In this section, we provide several algorithms to solve the APT problem for general $k \geq 1$ and $d \geq 1$. For only one phenotype with any number of responsible items ($k = 1$ and $d \geq 1$), we give an algorithm using at most $d \lceil \log_2 n\rceil + (d + 2)$ tests. Then, for more than one phenotype, with at most d responsible items, we provide an algorithm to solve the APT problem with $4\ d\lceil \sqrt{k/2}\ \rceil \cdot \lceil \log_2 n\rceil$ tests.

3.1 One phenotype with at most *d* responsible items ($k = 1$, $d \geq 1$)

Assume that we can solve one OR-phenotype ($k = 1$) problem with d OR-items in $t(n,d)$ tests [9]. We shall first use two tests to determine the mechanism of the phenotype and at the same time reduce the size of the problem. Then the responsible items can be identified by Theorem 1.

Theorem 2: Let $U = \{1,2, \ldots, n\}$. The APT problem with $k = 1$ unknown phenotype and at most d ($d \geq 1$) responsible items can be solved in $max_{s = 0,\ldots,\lceil \log 2n \rceil}\{t(n/2^s, d)+2s\}$ tests.

Proof: Without loss of generality, assume n is a power of 2. Partition U into two equal size subsets S_1 and S_2. Apply a test on S_1 and S_2 respectively. There are 4 outcomes.

If $P(S_1) = 1$ and $P(S_2) = 1$, this implies an OR-phenotype. This reduces to the APT problem of size n with OR-mechanism which can be solved using $t(n,d)$ tests.

If $P(S_1) = 0$ and $P(S_2) = 0$, this implies an AND-phenotype. This reduces to the APT problem of size n with AND-mechanism which can be solved using $t(n,d)$ tests (converting AND-items to OR-items by applying Theorem 1).

If $P(S_1) = 1$ and $P(S_2) = 0$, or alternatively $P(S_1) = 0$ and $P(S_2) = 1$, this reduces to the original problem of half of the size, which can be solved recursively.

The number of required tests would be the maximum of these cases as given in the statement of the theorem.□

Corollary: The APT problem with $k = 1$ and $d \geq 1$ can be solved with $d\lceil \log_2 n \rceil + (d + 2)$ tests.

Proof: Hunag's Generalized Binary Splitting Algorithm [9] can solve the APT problem with one OR-phenotype ($k = 1$) with at most d OR-items in $t(n,d) = \lceil \log_2 \binom{n}{d} \rceil + d \leq d\lceil \log_2 n \rceil + d$ tests. Since $t(n, d) + 2 = d\lceil \log_2 n \rceil + d + 2 \geq d\lceil \log_2 n \rceil + d + 2 + (2 - d)s = t(n/2^s, d) + 2s$ when $d \geq 2$ for all positive integer s, the number of tests needed is $t(n, d) + 2 = d\lceil \log_2 n \rceil + (d + 2)$. □

3.2 Multiple phenotype with at most *d* responsible items ($k \geq 2$ and $d \geq 1$)

When there are more than one phenotype, the algorithm mentioned in Section 3.1 can be repeated k times and the APT problem can be solved with $kd\lceil \log_2 n \rceil + k(d + 2)$ tests. However, the test on a phenotype may also provide information of another phenotype as long as both phenotypes need the test results on the same subset or disjoint subset of items. In order to reduce the number of tests, we should design the subsets to be tested by different phenotypes such that each test can provide information to determine the responsible items for several phenotypes.

Consider solving the APT problem ($k = 1$ and $d = 2$) for phenotype P_p with at most two responsible items x_1 and y_1 using at most $3\lceil \log_2 n \rceil - 1$ tests as follows. First, we represent the n items by n distinct length-$\lceil \log_2 n \rceil$ binary numbers, e.g. item $b = b[1]b[2]\ldots b[\lceil \log_2 n \rceil]$. The idea is to deduce the binary representation of the responsible items by the phenotype test. If the tested subset is formed according to binary numbers of the items, e.g. the subset containing all items with 1 at their i-th digit, then a positive phenotype test on this subset would indicate that one of the responsible items has 1 in the i-th digit of its binary representation. Assume we perform 2 tests $P_p(S_{i,0})$ and $P_p(S_{i,1})$ such that $b \in S_{i,0}$ iff $b[i] = 0$ and $b \in S_{i,1}$ iff $[i] = 1$ for every digit i of the length-$\lceil \log_2 n \rceil$ binary numbers. Note that the same two tests would also give the test results of other phenotypes in $S_{i,0}$ and $S_{i,1}$ at the same time. If $P_p(S_{i,0}) = P_p(S_{i,1}) = 0$ or 1, we can conclude that some responsible items are in $S_{i,0}$ and some in $S_{i,1}$, i.e. these two sets of responsible items have different i-th digit, and P_p is AND- or OR-phenotype. Otherwise, $P_p(S_{i,0}) \neq P_p(S_{i,1})$, the set $S_{i,\alpha}$ with $P_p(S_{i,\alpha}) = 1$ contains all responsible items for P_p, i.e. the i-th digit of all responsible items is α which is either 0 or 1. If there is only one responsible item for P_p, the binary string $b = b[1]b[2]\ldots b[\lceil \log_2 n \rceil]$ such that $P_p(S_{j,b[j]}) = 1$ represents the responsible item of P_p.

The algorithm starts with $i = 1$, we perform a test on $P_p(S_{1,0})$ and $P_p(S_{1,1})$. If $P_p(S_{1,0}) \neq P_p(S_{1,1})$, i.e. the first digit of all responsible items for P_p are the same. After having determined the first digit, we can continue the test on $P_p(S_{2,0})$ and $P_p(S_{2,1})$ and so on. Assume the j-th digit is the first digit with $P_p(S_{j,0}) = P_p(S_{j,1})$, i.e. the first $j - 1$ digits of all responsible items are the same but different at the j-th digit. There are two possible length-j binary numbers $b[1]b[2]\ldots b[j - 1]0$ and $b[1]b[2]\ldots b[j - 1]1$ representing the prefixes of the binary representations of all responsible items of P_p. For each $i > 0$, there are at most d length-i binary numbers

$b[1]b[2]\ldots b[i]$ representing the prefixes of the binary representations of all responsible items of P_p. In order to determine the $(i + 1)$-th digits of the responsible items of P_p, two tests on items with the prefixes of their binary representations $b[1]b[2]\ldots b[i]0$ and $b[1]b[2]\ldots b[i]1$ have to be performed. A total of $2d$ tests might be needed for each digit and $2d\lceil \log_2 n\rceil$ in total for a particular phenotype. Thus, $2dk\lceil \log_2 n\rceil$ tests are needed for k phenotypes.

However, many of these tests can be shared. For example, when the responsible items of two phenotypes with the same mechanism have the same prefix $b[1]b[2]\ldots b[i]$, the test on the $(i + 1)$-th positions of these two items can be performed at the same time. Similarly, when there are two disjoint sets of phenotypes S_α and S_β with the same mechanism each has a responsible item with prefix $b[1]b[2]\ldots b[i]$ and $b'[1]b'[1]\ldots b'[i]$ respectively, the test on the $(i + 1)$-th positions of these items can be performed at the same time. However, two disjoint set with different prefixes of the same phenotype cannot be tested together as we cannot associate the test results to which set or both sets of items. Thus, we can construct a graph G with each vertex represents a subset of prefixes of some responsible items of different phenotypes, on which a single test can be performed without mixing up their results. There is an edge between two vertices u and v if and only if there is a phenotype P_p having some responsible items with prefixes in u and v at the same time, i.e. the tests on these two subset of prefixes cannot be performed together. Given two vertices u and v with no edge in between, the test results of the phenotypes having responsible item with prefix in u will not be affected by those items with prefix in v of different phenotypes, and vice versa. Thus, the test on the responsible items with prefixes in these two vertices u and v can be performed at the same time. Vertices without edges connecting them can be merged together until a clique is formed. The maximum number of prefixes that cannot be performed at the same time is the same as the size of the largest clique in the graph G.

Lemma 1: Given x phenotypes each with at most d responsible items with the same mechanism. Let each vertex in G represent the prefixes of some responsible items and there is an edge between two vertices u and v if and only if a phenotype P_p has two items with prefixes in u and v at the same time, the size of the largest clique formed by G is at most $d\sqrt{x}$ **.**

Proof: Given a phenotype P_p with at most d responsible items, there are at most $\binom{d}{2}$ edges corresponding to P_p. Since there are x phenotypes, there are at most $x\binom{d}{2}$ edges in graph G. Since a clique of size c has $\binom{c}{2}$ edges and $\binom{c}{2} \leq k'\binom{d}{2}$, which implies $c \leq d\sqrt{x}$. □

Theorem 3: The number of tests needed for solving the APT problem with k phenotypes each with at most d responsible items is at most $4\, d\lceil \sqrt{k/2}\,\rceil \cdot \lceil \log_2 n\rceil$.

Proof: Consider there is x phenotypes with OR-mechanism and $k - x$ phenotypes with AND-mechanism. By Lemma 1, at most $2\, d\sqrt{x} + 2\, d\sqrt{k-x}$ tests are needed for determining each digit of the responsible items. The maximum number of tests happens when $x = k/2$. Thus at most $4\, d\sqrt{k/2}$ tests are needed for each digit and at most $4\, d\lceil \sqrt{k/2}\,\rceil \cdot \lceil \log_2 n\rceil$ tests are needed for solving the APT problem. □

4 APT algorithms for special *k* and *d*

For some particular values of k and d, we are able to derive better algorithms to solve the APT problem. For $k = 1$, $d = 2$, we can use $2\lceil \log_2 n\rceil$ tests to solve the problem with 4 fewer tests than the general solution given in Section 3.1. For $k = 2$ and $d = 1$, only $\lceil \log_2 n\rceil$ tests are needed. And for $k = 2$ and $d = 2$, we can have a solution which uses at most $2\lceil \log_2 n\rceil + 2\lceil \sqrt{\log_2 n - 1}\,\rceil$ tests compared to the solution in Section 3.2 which uses at most $4(\lceil \log_2 n\rceil - 1) + 2$ tests.

4.1 One phenotype (k = 1 and d = 2)

Assume that we only have one phenotype ($k = 1$) with unknown mechanism and at most 2 responsible items, we will show that the APT problem can be solved by $2\lceil \log_2 n \rceil$ tests which matches the lower bound $\lceil \log_2(\binom{n}{1}+\binom{n}{2}+\binom{n}{2}) \rceil = \lceil 2\log_2 n \rceil$ when n is power of 2. In Lemma 2, we first show that using a binary search technique, we can locate the responsible items if the mechanism of the phenotype is known and the responsible items are in two disjoint known subsets.

Lemma 2: Given a phenotype with two responsible items of *known* mechanism, let $G = \{1,2, \ldots, n\}$ be divided into two disjoint subsets S_1 and S_2 with each subset containing one responsible item. Locating the responsible item in S_1 or S_2 takes $\lceil \log_2|S_1| \rceil$ and $\lceil \log_2|S_2| \rceil$tests respectively.
Proof: Without loss of generality, we show how to locate the responsible item x in S_1. If the phenotype is an AND-phenotype, we know that the other responsible item y is in S_2. So, we divide S_1 into two subsets S_{11} and S_{12} of equal size, and test ($S_{11} \cup S_2$), if the result is positive, then x is in S_{11}, otherwise, x is in S_{12}. Each round, we can remove half of the items of S_1 from consideration. So, $\lceil \log_2|S_1| \rceil$ tests are sufficient. If the phenotype is an OR-phenotype, again we divide S_1 into two equal subsets S_{11} and S_{12}, we only need to test S_{11} to see if it contains x. So, $\lceil \log_2|S_1| \rceil$ tests are sufficient. Similarly $\lceil \log_2|S_2| \rceil$ tests are sufficient for determine the responsible item in S_2. □

Based on Lemma 2, we can solve the problem of a single phenotype of unknown mechanism with at most two responsible items x and y ($x = y$ when there is only one responsible item) using a recursive algorithm as follows. We divide G into two disjoint sets S_1 and S_2, and test S_1 and S_2 separately. There are four cases:
(a) $P_1(S_1) = 1$ and $P_1(S_2) = 0$
It implies that both responsible items x and y are in S_1, so we can recursively work on S_1 only.
(b) $P_1(S_1) = 0$ and $P_1(S_2) = 1$
Similar to Case (a), it implies that both responsible items x and y are in S_2.
(c) Both $P_1(S_1)$ and $P_1(S_2)$ equal 0
It implies an AND-phenotype with exactly two responsible items, w.l.o.g $x \in S_1$ and $y \in S_2$. By Lemma 2, we can locate x in S_1 and y in S_2 using $\lceil \log_2(n/2) \rceil = \lceil \log_2 n \rceil - 1$ additional tests.
(d) Both $P_1(S_1)$ and $P_1(S_2)$ equal 1
Similar to Case (c), it implies an OR-phenotype with exactly two responsible items, w.l.o.g $x \in S_1$ and $y \in S_2$. By Lemma 2, we can locate x in S_1 and y in S_2 using $\lceil \log_2(n/2) \rceil = \lceil \log_2 n \rceil - 1$ additional tests.
So, the algorithm uses only $2\lceil \log_2 n \rceil$ tests to solve the APT problem.

4.2 Two phenotypes with at most one responsible item (k = 2 and d = 1)

This is a very easy case and is not covered in the previous sections. Since there is only one responsible item for each phenotype, the mechanisms of the phenotype can be ignored as both OR- and AND-mechanisms are the same. The responsible items can be determined by performing a binary search on both phenotypes at the same time until the two responsible items are found in different subsets, i.e. the test results are different for the two phenotypes. Then a binary search on these two subsets can be performed simultaneously because the test result of a phenotype does not affect the test result of another phenotype. The total number of tests needed is only $\lceil \log_2 n \rceil$.

4.3 Two Phenotypes with at most two responsible items (k = 2 and d = 2)

The same binary search technique described in the Section 4.1 cannot be applied when $d = 2$ since it is no longer possible to ignore the mechanisms of the phenotypes. The same procedure does not work when the two phenotypes have different mechanisms. To illustrate the problem, assume phenotype P_1 is an AND-phenotype and phenotype P_2 is an OR-phenotype, S_1 and S_2 are two disjoint subsets of G, each contains two responsible items, one from each phenotype. Using the binary search technique as described in Lemma 2, we can divide S_1 into S_{11} and S_{12}. However, to search the responsible item for P_1 in S_1, we need to test ($S_{11} \cup S_2$) as P_1 is an AND-phenotype. On the other hand, to determine the responsible item for the OR-phenotype P_2, we need to test S_{11} only. So, we cannot get the test results of phenotypes on different sets with a single test. In this section, we will describe a recursive algorithm for solving APT problem with two phenotypes of unknown mechanism, each with at most 2 responsible items using at most $2\lceil \log_2 n \rceil + 2\lceil \sqrt{\log_2 n - 1} \rceil$ tests.

We divide G into two disjoint subsets of equal size S_1 and S_2 and test S_1 and S_2. For each phenotype P, there are two possible outcomes (1) $P(S_1) \neq P(S_2)$, the phenotype appears in exactly one subset, i.e. one subset does not contain any responsible item related to the phenotype, or (2) $P(S_1) = P(S_2) = 1$ (or 0), the phenotype appears (disappears) in both subsets S_1 and S_2 and must be OR-mechanism (AND-mechanism) with exactly two responsible items, i.e. each subset contains one responsible item for the phenotype. Depending on the outcomes of the two phenotypes, we have the following three cases. When the outcomes for both phenotypes are (1), it is case (a). When the outcomes for both phenotypes are (2), it is case (c). Case (b) is when the outcomes for two phenotypes are different, one is (1) and the other is (2). Cases (a) and (b) are relatively easier and the phenotypes can be determined using at most $2\lceil \log_2 n \rceil + 1$ tests for each case. The details of handling these cases can be found in the appendix. Here, we mainly focus on Case (c).

Let x_1, y_1 be the responsible items for P_1 and x_2, y_2 be the responsible items for P_2. Note that when P_1 (P_2) has only one responsible item, $x_1 = y_1$ ($x_2 = y_2$). Without loss of generality, let $\{x_1, x_2\} \subseteq S_1$, $\{y_1, y_2\} \subseteq S_2$ and the mechanisms of phenotype P_1 and P_2 are known. If the two phenotypes are of the same mechanism, similar to the case for $k = 2$ and $d = 1$, we can perform a binary search on S_1 for x_1 and x_2 together using ($\lceil \log_2 n \rceil - 1$) tests and then on S_2 for y_1 and y_2 using another ($\lceil \log_2 n \rceil - 1$) tests. Thus, $2\lceil \log_2 n \rceil$ tests in total will be needed.

In the situation where the two phenotypes are of different mechanisms, assume P_1 is OR-mechanism and P_2 is AND-mechanism. Initially, $\{x_1, x_2\} \subseteq S_1$ and $\{y_1, y_2\} \subseteq S_2$. Partition S_1 into two subsets of equal size S_{11} and S_{12} and S_2 into S_{21} and S_{22}. Two tests are performed on $S_{11} \cup S_{21}$ and $S_{11} \cup S_{22}$ respectively. Depending on the test outcomes, the sizes of sets containing the responsible items will be reduced. If the test outcome of the OR-mechanism phenotype P_1 on $S_{11} \cup S_{21}$ and $S_{11} \cup S_{22}$ is:

(0,1): then $x_1 \in S_{12}$ and $y_1 \in S_{22}$
(1,0): then $x_1 \in S_{12}$ and $y_1 \in S_{21}$
(1,1): then $x_1 \in S_{11}$ and $y_1 \in S_2$
(0,0): this case is not possible

If the test outcome of the AND-mechanism phenotype P_2 on $S_{11} \cup S_{21}$ and $S_{11} \cup S_{22}$ is:

(0,1): then $x_2 \in S_{11}$ and $y_2 \in S_{22}$
(1,0): then $x_2 \in S_{11}$ and $y_2 \in S_{21}$
(1,1): this case is not possible
(0,0): then $x_2 \in S_{12}$ and $y_2 \in S_2$

As you can see, cases that halve the sizes of the sets containing the responsible items or produce disjoint sets containing x_1 and x_2 (y_1 and y_2) are not problematic (see Lemma 3 in the Appendix) and the responsible items for the phenotypes can be determined with $2\lceil \log n \rceil$ tests. In fact, the outcome (1,1) for the OR-mechanism phenotype P_1 with outcome (0,1) or (1,0) for the AND-mechanism phenotype P_2 will result in $\{x_1, x_2\} \subseteq S_{11}$,

$y_2 \in S' = S_{21}$ or S_{22} and $y_1 \in S_2$ where $S' \subseteq S_2$. While the outcome (0,0) for the AND-mechanism phenotype P_2 with outcome (0,1) or (1,0) for the OR-mechanism phenotype P_1 will result in $\{x_1, x_2\} \subseteq S_{12}$, $y_1 \in S' = S_{21}$ or S_{22} and $y_2 \in S_2$ where $S' \subseteq S_2$. A sub-problem P that solving the APT problem with three subsets: $\{x_1, x_2\} \subseteq T$, $y_1 \in T_O$ and $y_2 \in T_A$ where $T_A \subseteq T_O$ or $T_O \subseteq T_A$. Assume $T_A \subseteq T_O$, we can partition subset T (T_A) into two equal-size disjoint subsets T_1 and T_2 (T_{A1} and T_{A2}) and perform 2 tests ($T_1 \cup T_{A1}$ and $T_1 \cup T_{A2}$) with the different cases ($P_1(T_1 \cup T_{A1})$, $P_2(T_1 \cup T_{A1})$), ($P_1(T_1 \cup T_{A2})$, $P_2(T_1 \cup T_{A2})$):

Case i:

(0,0), (1,0): $\{x_1, x_2\} \subseteq T_2$, $y_2 \in T_A$, $y_1 \in T_{A2}$ to be solved by recursion

Case ii:

(1,0), (1,1): $\{x_1, x_2\} \subseteq T_1$, $y_2 \in T_{A2}$, $y_1 \in T_O$

(1,1), (1,0): $\{x_1, x_2\} \subseteq T_1$, $y_2 \in T_{A1}$, $y_1 \in T_O$

We can recursively divided T_1 and T_{A1} to determine x_1, x_2 and y_2 using $2\lceil \log_2 n \rceil$ tests. However, it takes $\lceil \log_2 n \rceil - 1$ extra test to determine y_1, so a total of $3\lceil \log_2 n \rceil - 1$ might be needed. In order to determine y_1 more efficiently, we perform a test on $T_O - T_{A1}$ every s steps (halving of T_1). If $P_1(T_O - T_{A1}) = 1$, $y_2 \in T_{A1}$, $y_1 \in T_O - T_{A1}$ the problem can be solved by Lemma 3 using $2\lceil \log_2 n \rceil + \lceil \sqrt{\log_2 n} \rceil + 1$ tests in total. If $P_1(T_O - T_{A1}) = 0$, the problem can be solved by recursion with the size of T_O at most $\lceil n / 2^s \rceil$. In the worst case, $\lceil \log_2 n - 1 \rceil / s$ tests on $T_O - T_{A1}$ are needed and an extra s tests are needed for determining y_1 in $T_O - T_{A1}$ of size $2^s - 1$ using binary searching. Thus, $\lceil \log_2 n - 1 \rceil / s + s$ extra tests are needed for determining y_1 with the minimum value when $s = \lceil \sqrt{\log_2 n - 1} \rceil$. The total number of tests required is $2\lceil \log_2 n \rceil + 2\lceil \sqrt{\log_2 n - 1} \rceil$.

Case iii:

Other cases: $\{x_1, x_2\}$, y_1 and y_2 are in disjoint subsets and the problem can be solved easily by Lemma 3 in the Appendix and the phenotypes can be determined with $2\lceil \log n \rceil$ tests.

Thus, the APT problem can be solved using at most $2\lceil \log_2 n \rceil + 2\lceil \sqrt{\log_2 n - 1} \rceil$ tests.

5 Conclusions

In this paper, we introduced the phenotype testing problem which is an important generalization of the well-known combinatorial group testing problem. We have obtained several interesting results for the adaptive version of the problem for handling any number of phenotypes (k) and any number of responsible items (d) in each phenotype. For some special cases with k and d smaller than 2, algorithms using near-optimal number of tests are also presented. In this paper, we only consider two common mechanisms, namely And-version and OR-version, on how the subset of items relates to a phenotype. More complicated mechanisms, such as mixing AND and OR in the same subset of inducing items, should be modeled and considered. Also, even for the OR-version and AND-version of the problems, the lower bound and upper bound are still not closed yet. Finding a better algorithm which uses fewer tests or finding a better lower bound would be desirable.

References

1. M.A. Bishop, A.J. Macula, T.E. Renz, and V.V. Ufimtsev, "Hypothesis group testing for disjoint pairs", J. Comb. Optim. 15:7-16 (2008).
2. A. Bonis, L. Gasieniec, U. Vaccaro, “Optimal Two-Stage Algorithms for Group Testing Problems”, SIAM J. on Computing 34(5): 1253-1270 (2005).
3. F. Chin, H. Leung, S.M. Yiu, “Non-Adaptive Complex Group Testing with Multiple Positive Sets”, In proceeding of 8th Annual Conference on Theory and Applications of Models of Computation (TAMC), to

appear in 2011.

4. P. Deng, F.K. Hwang, Weili Wu, David MacCallum, Feng Wang, and Taieb Znati, "Improved construction for pooling design", J. Comb. Optim. 15:123-126 (2008).
5. D.Z. Du, F. Hwang, "Combinatorial group testing and its applications", 2nd edition, World Scientific, Singapore (2000).
6. D.Z. Du, F.K. Hwang, Weili Wu, and Taieb Znati, "New construction for transversal design", Journal of Computational Biology 13(4): 990-995 (2006).
7. S.M. Elbashir et al., “Duplexes of 21-Nucleotide RNAs Mediate RNA Interference in Cultured Mammalian Cells”, Nature 411: 494-498 (2001).
8. A. Fire et al., “Potent and Specific Genetic Interference by Double-Stranded RNA in Caenorhabditis Elegans”, Nature 391: 806-811 (1998).
9. F.K. Hwang, “A method for detecting all defective members in a poputation by group testing”, J. Amer. Statist. Assoc. 67: 605-608 (1972).
10. N. Jendreyko et al., “Phenotypic Knockout of VEGF-R2 and Tie-2 with an Intradiabody Reduces Tumor Growth and Angiogenesis *in vivo*”, PNAS 102(23): 8293-298 (2005).
11. C.H. Li, “A Sequential Method for Screening Experimental Variables”, Journal of the American Statistical Association 57(298): 455-477 (1962).
12. H.Q. Ngo and D.Z. Du, “A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening”, in D-Z. Du, P.M. Pardalos, P.M., and Wang J. (eds.), Discrete Mathematical Problems with Medical Applications, DIMACS Series, 55, American Mathematical Society, Providence, RI (2000).
13. F. Ruberti et al., “Phenotypic Knockout of Nerve Growth Factor in Adult Transgenic Mice Reveals Severe Deficits in Basal Forebrain Cholinergic Neurons, Cell Death in the Spleen, and Skeletal Muscle Dystrophy”, J. Neurosci. 20(7): 2589-601 (2000).
14. A.G. Yang et al., “Phenotypic Knockout of HIV Type 1 Chemokine Coreceptor CCR-5 by Intrakines as Potential Therapeutic Approach for HIV-1 Infection”, Proc. Natl. Acad. Sci. 94: 11567-572 (1997).

Appendix

A.1 Two Phenotypes with at most two responsible items (*k* = 2 and *d* = 2): Cases (a) and (b)

Recall that to handle two phenotypes with at most two responsible items, we divide G into two disjoint subsets of equal size S_1 and S_2 and test S_1 and S_2. For each phenotype P, there are two possible outcomes (1) $P(S_1) \neq P(S_2)$ and (2) $P(S_1) = P(S_2) = 1$ (or 0). Depending on the outcomes of the two phenotypes, we have three cases.

Case (a): When the outcomes for both phenotypes are (1).

Case (b): When the outcomes for the phenotypes are different, one is (1) and the other is (2).

Case (c): When the outcomes for both phenotypes are (2).

In this appendix, we show how to handle Cases (a) and (b). Let x_1, y_1 be the responsible items for P_1 and x_2, y_2 be the responsible items for P_2. Note that when P_1 (P_2) has only one responsible item, $x_1 = y_1$ ($x_2 = y_2$).

Case (a) All responsible items (at most 2) of each phenotype are in a single subset

(i) If the responsible items for both phenotypes, i.e. $\{x_1, y_1, x_2, y_2\}$, are in a single subset, we can ignore all items in another subset and the problem can be solved recursively with its size reduced by half.

(ii) If the responsible items for the two phenotypes are in different subsets S_1 and S_2, say $\{x_1, y_1\} \subseteq S_1$ and $\{x_2, y_2\} \subseteq S_2$, this reduces to two APT problems with $k = 1$ and $d = 2$, each can be solved using $2\lceil \log n \rceil$ tests using Algorithm 1. We can apply one single test on S_1 and S_2 for two phenotypes simultaneously as the test result for S_1 will not affect the test result for S_2 and vice versa. For example, if we need to test S' in S_1 and S'' in S_2, we can combine these two tests into one and test $S' \cup S''$ on both phenotypes instead.

Case (b): $\{x_1, x_2, y_2\} \subseteq S_1$ and $y_1 \in S_2$ and the mechanism of phenotype P_1 is known

This is a more complicated case. Subset S_1 contains one item for phenotype P_1 and the two items for phenotype P_2. We keep partitioning S_1 into two disjoint subsets, S_{11} and S_{12}, of equal size and test one of the subsets, say S_{11}. W.l.og. assume P_1 is OR-mechanism[1], based on the test results on S_{11}, we have the following cases for $(P_1(S_{11}), P_2(S_{11}))$.

(1,0): Perform an extra test on S_{12} and consider the possible cases for $P_2(S_{12})$

(i) $P_2(S_{12}) = 0$: $\{x_1, x_2\} \subseteq S_{11}$, $y_2 \in S_{12}$, $y_1 \in S_2$ and P_2 is AND-mechanism (to be solved by Lemma 3 given below.)

(ii) $P_2(S_{12}) = 1$: $x_1 \in S_{11}$, $\{x_2, y_2\} \subseteq S_{12}$ and $y_1 \in S_2$ (to be solved by Lemma 4 given below.)

(0,1):Similar to the case (1,0), perform an extra test on S_{12} and consider the cases for $P_2(S_{12})$

(i) $P_2(S_{12}) = 0$: $x_1 \in S_{12}$, $\{x_2, y_2\} \subseteq S_{11}$ and $y_1 \in S_2$ (to be solved by Lemma 4.)

(ii) $P_2(S_{12}) = 1$: $\{x_1, x_2\} \subseteq S_{12}$, $y_2 \in S_{11}$, $y_1 \in S_2$ and P_2 is OR-mechanism (to be solved by lemma 3.)

(1,1): There are two possible cases

(i) $\{x_1, x_2, y_2\} \subseteq S_{11}$ and $y_1 \in S_2$

(ii) $\{x_1, x_2\} \subseteq S_{11}$, $y_2 \in S_{12}$, $y_1 \in S_2$ and P_2 is OR-mechanism

Case (i) is the same as case (b) while case (ii) is a special case of case (b) with $x_2 = y_2$ if set S_{12} is ignored. Thus, the problem can be solved recursively on S_{11} and S_2 with $|S_{11}| = |S_1| / 2$. Note that when $x_2 = y_2$ or in case (ii), we will not have the case $(P_1(S_{11}), P_2(S_{11})) = (1,0)$ or $(0,1)$ and can determine x_1 and x_2 using $\lceil \log_2|S_{11}| \rceil$ tests. With an extra test on S_{12}, we can determine whether it is $x_2 = y_2$ or case (ii). If $x_2 = y_2$, we can determine $y_1 \in S_2$ by binary search using $\lceil \log_2|S_2| \rceil$ tests, otherwise, we can determine $y_2 \in S_{12}$ and $y_1 \in S_2$ by binary search on S_{12} and S_2 simultaneously using $\max\{\lceil \log_2|S_{12}| \rceil, \lceil \log_2|S_2| \rceil\}$ tests. Thus $2\lceil \log_2 n \rceil + 1$ tests are sufficient.

(0,0): Similar to case (1,1), except that P_2 is AND-mechanism

With Lemma 3 and Lemma 4, we can show that for case (b), at most $2\lceil \log_2 n \rceil + 1$ tests are sufficient.

Lemma 3: Let G be divided into three disjoint subsets S_1, S_2 and S_3. If $y_1 \in S_1$, $y_2 \in S_2$ and $\{x_1, x_2\} \subseteq S_3$, determining all the responsible items in S_1, S_2 and S_3 takes at most $\lceil \log_2(\max\{|S_1|,|S_2|\}) \rceil + \lceil \log_2|S_3| \rceil$ tests when the mechanisms of P_1 and P_2 are known.

Proof: Our approach is to determine x_1 and x_2 in S_3 first and determine y_1 and y_2 by performing binary search for both phenotypes on subsets S_1 and S_2 simultaneously. When both phenotypes are OR-mechanism, similar to the problem for $k = 2$ and $d = 1$, we determine x_1 and x_2 by binary search for both phenotypes on subset S_3 which requires $\lceil \log_2|S_3| \rceil$ tests. We can then determine y_1 and y_2 by binary search for both phenotypes on subsets S_1 and S_2 simultaneously using $\lceil \log_2(\max\{|S_1|,|S_2|\}) \rceil$ tests. Similarly, for both phenotypes are AND-mechanism, we perform binary search for both phenotypes as OR-mechanism except that each test set in S_3 includes $S_1 \cup S_2$.

[1] If P_1 is AND-mechanism, the tests on S_{11} and S_{12} will include S_2 which contains the other responsible items for P_1.

If one phenotype, say P_1, is OR-mechanism and another phenotype, say P_2, is AND-mechanism, we can divide S_3 into two equal subsets S_{31} and S_{32} and test ($S_{31} \cup S_2$), if the results are (i) positive (or negative) for both phenotypes, both items $\{x_1, x_2\}$ are in S_{31} (or S_{32}), then we can recursively solve the problem with half of the size of S_3, otherwise, (ii) x_1 and x_2 are in different set S_{31} and S_{32}, and we can determine x_1 and x_2 at the same time by binary search and including S_2 in every test. As for $y_1 \in S_1$, $y_2 \in S_2$, we can also determine them by binary search simultaneously and depending whether P_1 and P_2 is AND-mechanism, x_1 or x_2 has to be included in each test. Therefore, $\lceil \log_2(\max\{|S_1|,|S_2|\}) \rceil + \lceil \log_2|S_3| \rceil$ tests are sufficient. □

Lemma 4: Let G be divided into three disjoint subsets S_1, S_2 and S_3. If $x_1 \in S_1$, $y_1 \in S_2$ and $\{x_2, y_2\} \subseteq S_3$, locating all the items in S_1, S_2 and S_3 takes $\max\{2\lceil \log_2|S_3| \rceil, \lceil \log_2|S_1| \rceil + \lceil \log_2|S_2| \rceil\}$ tests.

Proof: We apply Algorithm 1 on S_3 to determine both items in $2\lceil \log_2|S_3| \rceil$ tests. To determine the items in S_1 and S_2, it requires $\lceil \log_2|S_1| \rceil$ and $\lceil \log_2|S_2| \rceil$ tests respectively by binary search. Since the tests for S_3 and S_1 (S_3 and S_2) can be done simultaneously, thus $\max\{2\lceil \log_2|S_3| \rceil, \lceil \log_2|S_1| \rceil + \lceil \log_2|S_2| \rceil\}$ tests are sufficient. □