Efficient online-group-screening designs for agent identification — Source link

Tianxiang Gao, Omri M. Finkel, Jeff Dangl, Vladimir Jojic

Institutions: University of North Carolina at Chapel Hill

Published on: 17 Nov 2017 - bioRxiv (Cold Spring Harbor Laboratory)

Efficient online-group-screening designs for agent identification

Tianxiang Gao[1], Omri Finkel[2], Jeff Dangl[2,3], and Vladimir Jojic[1]

[1] Department of Computer Science, University of North Carolina, Chapel Hill, North Carolina, USA
[2] Department of Biology, University of North Carolina, Chapel Hill, North Carolina, USA
[3] Howard Hughes Medical Institute, Chevy Chase, Maryland, USA
[1] {tgao,vjojic}@cs.unc.edu

Abstract. Identifying significant causal agents among a large number of candidates is challenging. When experimental resources are limited, exhaustively screening a large number of agents for the desired effect could incur a large cost and take a substantial amount of time. However, in many large scale experiments, such as high-throughput screening (HTS), the ratio of causal to non-causal agents is usually very low.

In this paper, we introduce a group-screening strategy to efficiently screen causal agents by grouping them into treatments. Our analysis shows that when a large number of candidates factors are screened and true agent percentage is very low (less than 1%), even in the worst case we could save up to 80% of the experiment runs. In the case where experiments span many rounds, we provide an online version of the group-screening that can determine the best strategy automatically based on the existing results. We applied this method to a real HTS experiment with 50,000 candidates that would require 9 rounds to finish in an exhaustive case. Our analysis showed that by applying the online-group-screening method, in the worst case, we can use 3 rounds and 19.7% (9828/50000) total tests to identify all the agents.

Finally, we show that with minor modifications, this framework extends to more complex agent discovery problems.

Keywords: experiment design, important factor identification, pre-screening, feature selection, 1-bit compressed sensing, high-throughput screening

1 Introduction

Understanding the causes of phenotypic changes is one of the key goals of the biological research. Demonstrating that a particular candidate (e.g gene, mutant, bacterial strain) effects the change, provides evidence for a mechanistic relationship between a phenotype and a candidate. We will call such candidates "agents". Candidates that have no effect on phenotypes are called "non-agents". Screening designs examine the ability of each candidate to induce the change in phenotype [1]. We define "treatment" as a group of candidates that can be tested for phenotypic changes. The total time required to test a treatment from setting up to collecting results is called one "round". Typically, multiple treatments can be tested in parallel in a single round. We use "efficiency" to represent the maximal possible number of treatments that can be tested per round. In this paper, we aim to provide a design method to minimize the number of total rounds needed for identifying all agents.

Generally, a naive method will exhaustively test all the candidates individually. However, when there are too many candidates to be tested when efficiency is low, a naive method will take many rounds to finish. In plant and animal experiments, efficiency is limited by factors such as greenhouse space or incubator/cage number. Furthermore, typical biology experiments can take days to weeks to finish. We summarize the duration for some experiments in Table 1. For high-throughput screening (HTS) experiments, efficiency is limited by machine/robot/human experimenter efficiency. For example, in the HTS experiment conducted by Ma *et al.* [2], only 40-60 96-well plates were analyzed in a single round. It took 9 rounds to test all 50,000 candidate compounds. According to a review from Macarron *et al.* [3], in many drug discovery companies, compound screening step can take up to 3 months to finish (for HTS of 1 million compounds). These limitations prevent biologists from exploring more candidates.

Table 1. Example experiment durations

	Experiment duration (1 round)	Environment	Agent types
Ma *et al.* [2]	30 minutes	Microtitre plates	chemical compounds
Wei *et al.* [4]	2 days	Microtitre plates	carbon resources
Buffie *et al.* [5]	21 days	Murine gut	bacteria strains

In many screening problems, true agents are actually very few, which is known as the "effect sparsity assumption" [6]. Using this assumption, it is possible to reduce the number of total treatments by applying methods like the one proposed by Gupta *et al.* [7]. They provided an adaptive sequential tree-searching algorithm for efficiently discovering relevant features in a binary classification problem requires $s \log n$ tests and $s \log n$ rounds, where n is the total number of features, and s is the number of relevant features. This algorithm can be recast in the language of agent identification. Features in this setting can be seen as candidates, samples as treatments required to be tested, and the binary label of a sample as the target phenotype. Thus collecting labeled samples is equivalent to measuring target phenotypes for the corresponding treatments. Loosely, the idea is to split the candidates into two treatments. If a treatment induces phenotypic change, the treatment is deduced to contain at least one agent, and the candidates in this treatment will be split further, recursively. If a treatment induces no phenotypic change, then none of its candidates are agents.

The drawbacks of this approach are: 1) it takes up to $s \log n$ rounds to identify all the agents. And in each round, it only tests one treatment, which does not fully utilize the experimental resources. For example, an experiment with 50,000 candidates and 7 agents will take more than 100 rounds to finish, no matter how many treatments we can test each round. Further, assays tested in a larger number of batches can exhibit "batch effects" [8]. 2) In most cases, we do not know the true agent number s. If the problem has $s \log n > n$, this approach may result in much more treatments than the naive method. 3) In some cases, the number of candidates in one treatment can be limited because of the dilution limitation or maximal capacity of an assay.

In this paper, we present a **group-screening** method that can efficiently optimize the time required for discovery and overcome the drawbacks of the tree-searching method. This method subdivides n candidates into b group-screening treatments. According to the Pigeon-hole principle [9], whenever b is larger than the true agent number s, some treatments will not contain any true agents. Thus, we do not need to test the candidates in those treatments. We prove that we can identify all the agents using no more than $2\sqrt{ns} + s$ tests. We extend this method to **online-group-screening** that can utilize the results gathered at the end of each round to decide the best strategy based on a probabilistic analysis. Therefore, we can dynamically decide when to use group-screening and how many tests we can expect to save.

Figure 1 illustrates the group screening and online-group-screening methods. We have applied these methods to a real HTS experiment. Our results illustrate significant savings in both time and total tests. We believe this method will make agent identification problems less expensive, more efficient, and thus increase the pace of discovery.

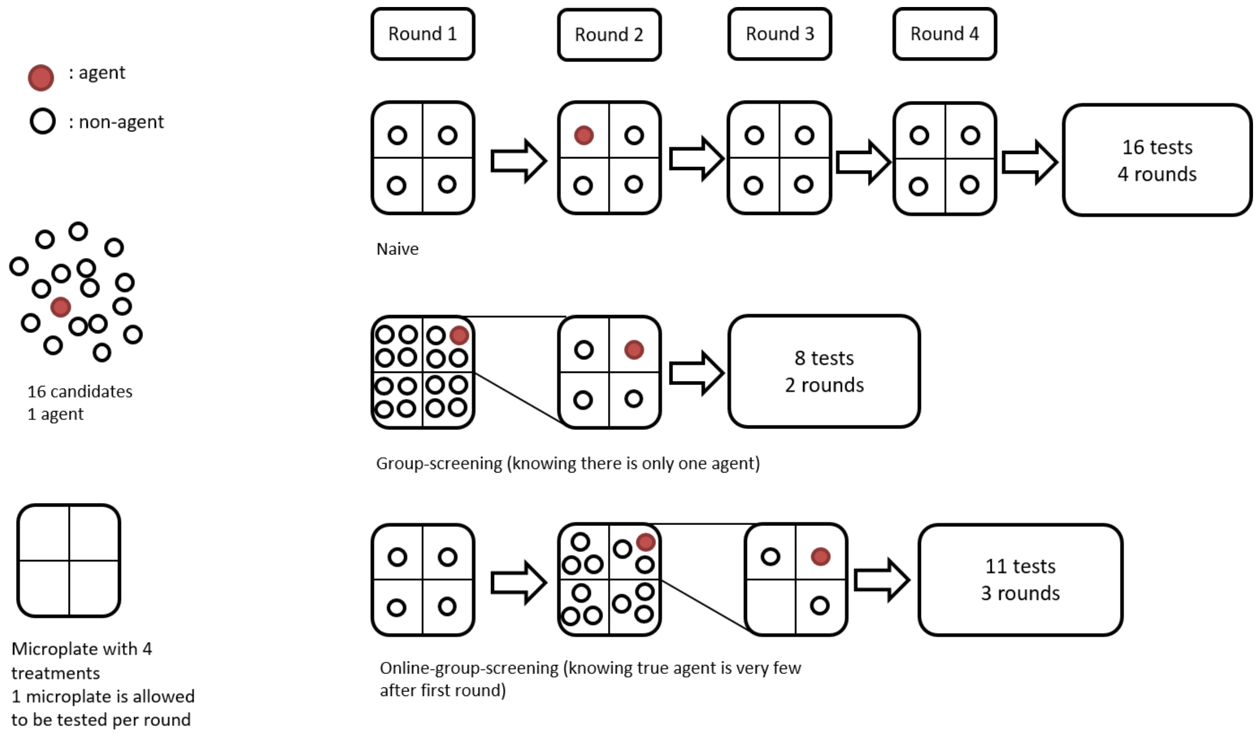
: agent
: non-agent
16 candidates
1 agent
Microplate with 4 treatments
1 microplate is allowed to be tested per round
Round 1
Round 2
Round 3
Round 4
16 tests
4 rounds
Naive
8 tests
2 rounds
Group-screening (knowing there is only one agent)
11 tests
3 rounds
Online-group-screening (knowing true agent is very few after first round)

Fig. 1. An example of naive, group-screening and online-group-screening algorithms. In this example, we have 16 candidates to test, one of which is a true agent. Each round we only allow at most 4 treatments to be tested in parallel, so efficiency is 4. The **naive** method exhaustively tests all candidates, using 16 tests and 4 rounds. If we know there is only one agent, the **group-screening** method randomly divides all 16 candidates into 4 group-screening treatments. After the first round, the treatment containing a true agent is further tested in the second round, using totally 8 tests in 2 rounds. When the true agent number is not known, we use the **online-group-screening** method. We run the first round as the naive method, and the results indicate that the true agent ratio is very low. Therefore, we use group-screening on the rest of the candidates. It takes 11 tests and 3 rounds in total.

In Section 2, we introduce our method and show its analytical and empirical performance. A case study of this method on a real HTS experiment is presented in Section 3. In Section 4, we illustrate how this framework can be extended to solve several different problems. Final discussion is in Section 5.

2 Method

In this section, we first specify the problem of agent identification. Next, we introduce the group-screening method and its theoretical savings compared to the naive method. Finally, we will present an online version that can adaptively choose the best method.

2.1 Problem specification

We assume that there are n candidates $U = \{x_1, \ldots, x_n\}$. The *true* agent set S is a subset of U with $s = |S|$ elements. We call any $X \subseteq U$ a treatment, and the outcome of a treatment is a function $f(X) = \mathbb{I}[X \cap S \neq \emptyset]$, where $\mathbb{I}[x] = \begin{cases} 1, & \text{x is true} \\ 0, & \text{x is false.} \end{cases}$

For simplicity, we will refer to treatments that yield an observation of 1 as **positive** outcomes, and those that yield 0 as **negative** outcomes.

We define a **round** as the time spent from setting up a treatment test to collecting the treatment test result. We assume all treatments require an equal amount of time to be tested. We define **efficiency** w as the maximal number of parallel treatments that can be tested in a single round. For example, in Figure 1, we can test 4 treatments in a single round, so efficiency $w = 4$.

We define the *sample bound* $O(n, s)$ of an algorithm as the upper bound of the total tests required for exactly identifying S for n candidates. We also define a *Time bound* $T(n, s, w)$ as the upper bound of the total rounds needed for exactly identifying S for n candidates at efficiency w.

A **naive algorithm** is to test each candidate in its own treatment. We can exactly identify all agents using n tests in $\lceil n/w \rceil$ rounds. Hence, this method has sample bound $O(n, s) = n$ and time bound $T(n, s, w) = \lceil n/w \rceil$.

In the following sections, we will describe a novel **group-screening algorithm** that has sample bound $O(n, s) = 2\sqrt{ns} + s$ and time bound $T(n, s, w) = \lceil \lceil \sqrt{ns} \rceil / w \rceil + \lceil (\lceil \sqrt{ns} \rceil + s)/w \rceil$.

2.2 Group-screening

The intuition behind the group-screening method is that by splitting the candidate set to an appropriate number of groups, we may be able to identify a significant portion of non-agents. We call these groups "group-screening treatments".

If a group-screening treatment has a negative outcome, we can mark all the candidates in this treatment as non-agents. In the next step, we only need to test the candidates from group-screening treatments that have a positive outcome. The size of a group-screening treatment refers the number of candidates pooled together. Group number b is the number of group-screening treatments to be tested. In an extreme setting, putting each candidate in its own treatment, where $b = n$, recovers the naive method. We present this method in Algorithm 1.

Algorithm 1: Group-screening algorithm

```
Group-screening($U$,$b$)
input  : Candidate set $U$, group number $b$
output: agent set $A$
Init : $A \leftarrow \emptyset$, $X_{1,\dots,b} \leftarrow \emptyset$ ;
for $i \leftarrow 1$ to $n$ do
    $j \leftarrow (i \bmod b) + 1$;
    $X_j \leftarrow X_j \cup x_i$ ;
end
for $i \leftarrow 1$ to $b$ do
    if $X_i$ has negative outcome then
        $U \leftarrow U \setminus X_i$
    end
end
return Naive(U);
```

We show two lemmas for group-screening method here:

Lemma 1. *Using group-screening for n candidates with b group-screening treatments, if there are at most s agents, Algorithm 1 will require at most $g(n, b, s) = b + min(s, b)(\frac{n}{b} + 1)$ tests in total.*

Proof. If $b \leq s$, in the worst case, each of the b group-screening treatments will have a positive outcome, hence the algorithm will require n more tests to find all the agents resulting in $b+n$ tests in total. If $b > s$, there will be at most s group-screening treatments with positive outcomes, and at least $b - s$ treatments will have negative outcomes. The algorithm will require $b + s\lceil\frac{n}{b}\rceil$ tests in total. Combining both cases, we will require no more than $g(n, b, s) = b + \min(s, b)(\frac{n}{b} + 1)$ tests. ■

If $b > s$, we can find b^* that will minimize $g(n, b, s)$ wrt b by taking the derivative of g and equating it to zero. This gives $b^* = \sqrt{ns}$. As b is an integer, we can take $b^* = [\sqrt{ns}]$. We call b^* the optimized group number, as it minimizes the sample bound $g(n, b, s)$. By choosing this optimized group number, we can derive the following theorem:

Theorem 1. *Using group-screening method on n candidates with b group-screening treatments, if there are at most s agents, Algorithm 1 with $b = [\sqrt{ns}]$ will require at most $O(n, s) = 2\sqrt{ns} + s$ tests.*

Proof. As $n \geq s$, we have $b^* \geq s$. Therefore, we have: $g(n, b^*, s) = [\sqrt{ns}] + s(\frac{n}{[\sqrt{ns}]} + 1) \leq 2\sqrt{ns} + s$. Hence, we can test $b = [\sqrt{ns}]$ group-screening treatments to identify all agents with no more than $O(n, s) = 2\sqrt{ns} + s$ tests. ■

Under the optimal group number $b = [\sqrt{ns}]$, Algorithm 1 requires $\lceil[\sqrt{ns}]/w\rceil$ rounds for group-screening step, and at most $\lceil([\sqrt{ns}] + s)/w\rceil$ for naive method step. Hence, the upper bound on time is $T(n, s, w) = \lceil[\sqrt{ns}]/w\rceil + \lceil([\sqrt{ns}] + s)/w\rceil$ rounds.

2.3 Decision rules for methods

The sample bound for the group-screening method can go beyond n when s is large, which is less efficient than using the naive method. Given n candidates with at most s agents, we wish to choose the most efficient method. Here, we discuss the decision rules for method selection. The naive method is a deterministic algorithm and will always need to test n treatments to identify agents among n candidates. Thus, if the sample bound is lower than the naive method's cost n, we can be sure to save tests in the worst case.

We give the following lemmas without proof as they require only simple algebra.

Lemma 2. *Given n candidates, if there are at most s agents and $\frac{s}{n} < (\sqrt{2}-1)^2 \approx 0.1716$, group-screening algorithm (Algorithm 1) will always require less than n tests to identify all of the agents.*

A similar analogy can be conducted for time bound:

Lemma 3. *Given n candidates and efficiency w, if there are at most s agents and $\frac{s}{n} < (\sqrt{8-4n/w}/2-1)^2$, the group-screening algorithm (Algorithm 1) will always need at most $\lceil n/w \rceil$ rounds to identify all of the agents.*

We summarize these features in Table 2.

Table 2. Comparison between naive and group-screening methods

Method	Naive	Group-screening
Sample bound	n	$2\sqrt{ns}+s$
Time bound	n/w	$\lceil\lceil\sqrt{ns}\rceil/w\rceil + \lceil(\lceil\sqrt{ns}\rceil+s)/w\rceil$
Decision bound for minimizing tests	NA	$s/n < 0.1716$
Decision bound for minimizing rounds	NA	$\frac{s}{n} < (\sqrt{8-4n/w}/2-1)^2$

To better illustrate the savings, we introduce the concept of saving ratio. The saving ratio indicates the percentage of tests or rounds that we can save. Let n^* be the actual total tests, we define test saving ratio $\Omega_t = 1 - n^*/n$. Similarly, let r^* be the actual total rounds, we define round saving ratio as $\Omega_r = 1 - r^*/\lceil n/w \rceil$. Using Lemma 2 and 3, we can derive the lower bound for saving ratios using the group-screening method:

$$\Omega_t \geq 1 - \frac{2\sqrt{ns}+s}{n} = 1 - 2\sqrt{s/n} - s/n$$
$$\Omega_r \geq 1 - \frac{\sqrt{ns}/w + (\sqrt{ns}+s)/w + 1}{n/w} = 1 - 2\sqrt{s/n} - s/n - w/n$$

Intuitively, the smaller the true agent number, the more we can save. The relationship between true agent ratio s/n and test saving ratio bound is shown in Figure 2 (a).

The round saving ratio bound is a function of both true agent ratio s/n and efficiency ratio w/n. The smaller the true agent ratio and efficiency ratio are, the more we can save. We showed this relationship in Figure 2 (b).

These bounds provide a reference for how much we can save by using the group-screening method. However, this is a worst case analysis. On average we can save more than this. We showed the worst case bound and empirical savings in Figure 3. The average saving ratio is typically larger than the worst case bound.

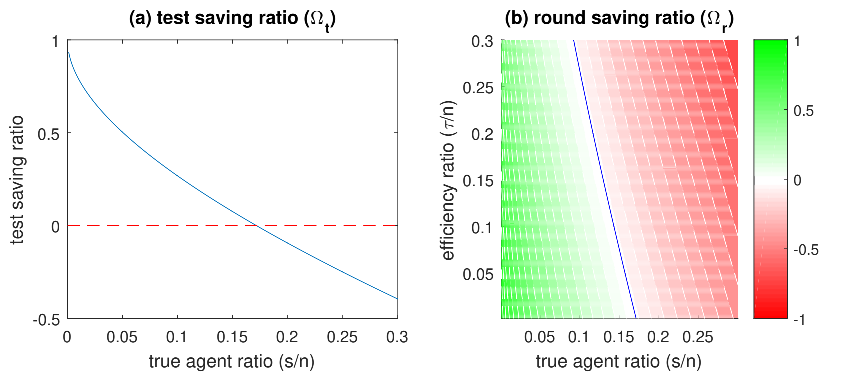

Fig. 2. (a) The test saving ratio lower bound as a function of true agent ratio. The blue line indicates the worst case saving ratio under the given true agent ratio. The smaller the true agent ratio, the more tests we can save. The red dashed line indicates the point of no savings (naive method). (b) Round saving ratio bound as a function of efficiency ratio and true agent ratio. The blue line indicates the decision boundary of positive savings in terms of rounds. Green indicates the savings, and red indicates the losses. We expect to save when true agent ratio is low and efficiency ratio is low.

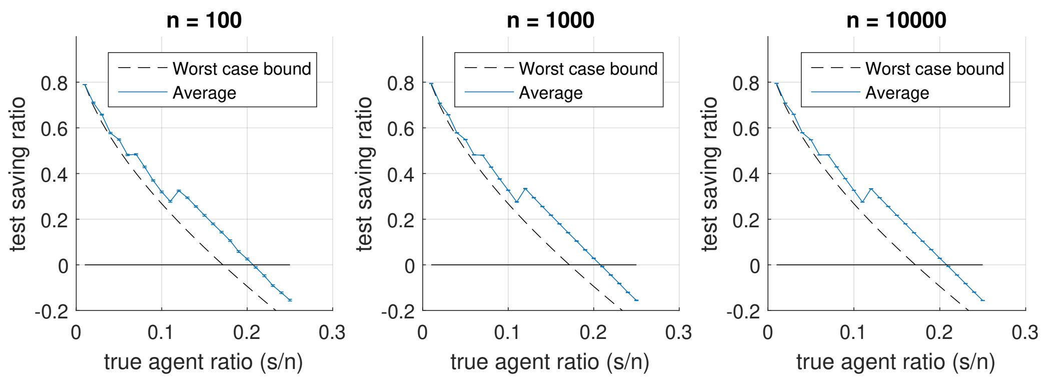

Fig. 3. Average test saving ratios for different number of candidates. From left to right, the cases for $n = 100, n = 1000$, and $n = 10000$. *Blue* line represents the average test saving ratio over 100 simulations, the error bar indicates the standard error. From simulation we found that the average test saving ratio is generally larger than the worst case bound.

2.4 An online-group-screening method

In most cases, we do not know the true number of agents s before we start the screening procedure. However, as we test more treatments, we gather more information about the frequency of the true agents. Suppose we have tested m treatments, if k true agents are identified, we can obtain a reasonable estimate for overall frequency of agents among all candidates using hypergeometric distribution. Here, we provide a probabilistic bound on the total number of agents.

Theorem 2. *Given the outcome of m treatments from n candidates and k agent detections, let s be the true agent number in n candidates, $Pr(s < n(k/m + t)) > 1 - e^{-2t^2 m}$, where $t \geq 0$.*

Proof. Let $h(s,n,m,k) = \frac{\binom{s}{k}\binom{n-s}{m-k}}{\binom{n}{m}}$ be the hypergeometric distribution probability of detecting k agents from randomly chosen m candidates among n total candidates with s agents. Chvatal [10] gives a bound for cumulative hypergeometric distribution tail $H(s,n,m,k) = \sum_{i=k}^{m} h(s,n,m,k)$ using Hoeffding [11] inequality $H(s,n,m,(s/n+t)m) \leq \exp^{-2t^2m}$ where $t \geq 0$. This relationship can be rewritten into: $Pr(k \geq (s/n+t)m) \leq \exp^{-2t^2m}$. Replacing s with the true non-agent number $n-s$ and k with non-agent detections $q = m-k$ yields the inequality in Theorem 2 using a sequence of algebraic manipulations on the probability of complementary events. ■

We can compute the upper bound on the number of agents which will hold with certain probability a. This upper bound is given by $s^* = (k/m + \sqrt{\frac{\ln(1-a)}{-2m}})n$. Therefore, the rest of the candidates will contain at most $s^* - k$ agents with probability more than a. The choice of a depends on our strategy for making use of existing treatment results. Large a is conservative, as it decreases the chance of both loss and saving. Small a is risky, as it increases the chance of both loss and saving. In our practical experience, we found that $a = 0.5$ will give us reasonable average savings, and we use $a = 0.5$ in the following analysis.

Given the tolerance for risk, we can set the probability a and obtain the corresponding upper bound on true agent number s. If the worst case saving ratio under such s is positive, we can choose to use the group-screening method for the rest of the candidates in the next round. Otherwise, we will continue to use the naive method for the next round. This online based algorithm is shown in Algorithm 2.

Algorithm 2: Online-group-screening algorithm

```
Online-group-screening($U$,$w$)
input  : Candidate set to test $U = \{x_1, \dots, x_n\}$, efficiency $w$
output: identified agent set $A$
$n \leftarrow |U|$; $R \leftarrow \lceil n/w \rceil$; $k \leftarrow 0$; $A \leftarrow \emptyset$;
for $r \leftarrow 1$ to $R$ do
    $i \leftarrow (r-1)w$; $m \leftarrow rw$;
    $S \leftarrow$ Naive$(x_{i+1}, \dots, x_{i+w-1})$;
    $A = A \cup S$; $k \leftarrow |S|$;
    $n \leftarrow n - w$;
    $s^* \leftarrow (k/m + \sqrt{\frac{\ln(1-a)}{-2m}})n$;
    $b \leftarrow [\sqrt{ns^*}]$;
    if $\lceil b/w \rceil + \lceil (b+s)/w \rceil \leq (R-r)$ then
        $B =$ Group-screening$(W, b)$ ;
        $A = A \cup B$;
        break;
    end
end
return $A$;
```

We showed an empirical analysis for the online-group-screening method in Figure 4. From the result, we found that even without knowing the true agent number s, our method will still be able to make savings. On the other hand, this method rarely incurs a loss.

2.5 Directions regarding the true experiments

Here, we discuss some practical concerns in true experiments with regard to the online-group-screening method.

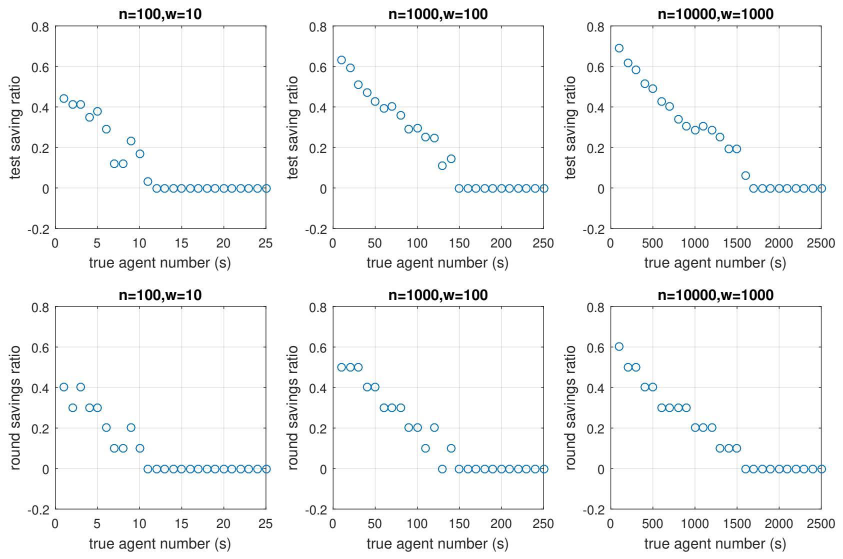

Fig. 4. An empirical analysis for the online-group-screening method. We ran 100 simulations for different settings of n, w, where true agent ratio is uniformly generated. The first row is the test saving ratio. The second row is the round saving ratio. We only show the result for the first 25% true agent ratio, as the rest 75% have 0 saving ratios in all the settings. No loss has occurred in all 100 simulations for all the settings.

Candidate limit per treatment. Many biological or chemical experimental settings naturally limit the number of candidates in a single treatment. Mixing too many candidates may dilute the agents below the threshold required for their effects. Given n candidates and candidate limit per treatment set as τ, at least $\lceil n/\tau \rceil$ groups are required. We can change the Algorithm 2 by modifying $b \leftarrow [\sqrt{ns^*}]$ to $b \leftarrow \max([\sqrt{ns^*}], \lceil n/\tau \rceil)$ to satisfy this limit.

Grouping preference In real experiments, we generally have side information about candidates before we start the screening procedure. Candidates believed to have similar effects should be placed in the same treatment. This design increases the chances that agents are concentrated in a single treatment.

We provide a Matlab implementation of this method:
`https://bitbucket.org/clingsz/groupscreening/src/`.

3 Case study for a high-throughput screening experiment

Our method performs well in settings where total candidate number is very large and true agents are very few. This is usually the case for high-throughput screening experiments. We investigated our method on a specific problem from Ma *et al.* [2], where 50,000 chemically diverse compounds were screened for inhibition of cAMP/flavone-stimulated Cl^- transport in epithelial cells expressing cystic fibrosis transmembrane conductance regulator (CFTR), which has a strong relationship with secretory diarrhea.

In this experiment, candidates are $n = 50,000$ chemical compounds. $s = 7$ compounds with significant effect – the true agents – were found using the high-throughput screening method. Each round, 40-60 96-well plates can be analyzed in parallel. Hence, the work efficiency is $w = 5760$. To fully analyze all 50,000 candidates, the naive method takes 9 rounds and 50,000 tests in total. By knowing $s = 7$, the group-screening algorithm showed that by randomly dividing all the candidates into 592 group-screening treatments, where each treatment contains at most 85 candidates, in the worst case we can identify all the true agents using 2 rounds and 1187 tests in total.

If we have no prior information about s, we can use the online-group-screening method by utilizing the results acquired from the first round. Our algorithm decided to randomly split all the rest of the candidates into 4024 group-screening treatments, with at most 11 candidates in each treatment. Finally, we can find all true agents using 3 rounds and 9828 tests, a five-fold saving in test effort. We summarized these savings and compared with the naive method in Table 3.

Table 3. Comparison between naive and online-group-screening methods on HTS

Method	Naive	Online-group-screening	group-screening (given $s \le 7$)
Total rounds	9	3	2
Total tests	50,000	9828	1187

If the maximal number of candidates is limited in each treatment, the online-group-screening method can still save tests. We showed the total number of tests and rounds required under different limitations in Figure 5. Even if at most 2 candidates are allowed in each treatment, we can save both time and tests by almost 30%.

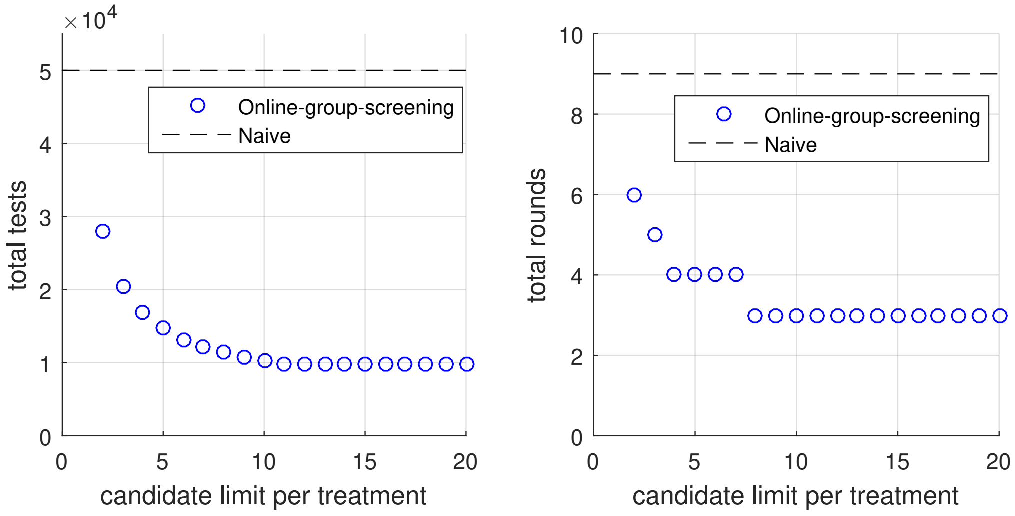

Fig. 5. Total tests and rounds needed for the online-group-screening method when candidate number in each treatment is limited. Left: total tests needed. Right: total rounds needed. Blue circles represent the result of online-group-screening method, black dashed line represents the total tests and rounds required by naive method.

4 Extensions

We can slightly modify our problem specifications to solve other variants. Next, we show 2 variant problems to illustrate the adaptability of our framework: essential set identification and agent identification for multiple phenotypes.

4.1 Essential set identification

Identifying the essential set is a common problem in biology. For example, in order to understand the molecular and biological functions of genes in a cell, Hutchison *et al.* [12] built a minimal genome by including only the essential genes in bacteria. Discovering exact genes that are essential for bacterial survival and growth is a prerequisite for successful constructing such minimal genome. A straightforward naive method is to exclude each gene one by one from the whole genome to test whether it is essential. Suppose we have n candidate genes $U = \{x_1, x_2, \ldots, x_n\}$. The essential set S is a subset of U with $s = |S|$ elements. We refer to any $X \subseteq U$ as a treatment, where we construct a bacteria without any of the candidates in X and let the bacteria grow. The outcome of a treatment is a function $f(X) = \mathbb{I}[X \cap S \neq \emptyset]$.

In this function, if the bacteria does not grow when **excluding** candidates in X, then $f(X) = 1$. This indicates that some of the genes in X are essential. Conversely, if the bacteria grow, $f(X) = 0$, then none of the candidate compounds in X are essential since this exclusion does not affect the bacteria growth. The essential set identification problem can be mapped onto our framework of agent identification.

4.2 Agent identification for multiple phenotypes

Causal factors may drive multiple phenotypes. For example, plant growth promoting bacteria can influence plant height, flowering time, number of flowers, or the root architecture [13–16]. When screening for plant growth promoting traits of bacteria, the expected phenotype is often unknown a-priori. Here, we show how our framework can be adapted to the problem of agent identification for multiple phenotypes.

Given measurements of v phenotypes and unknown true agent sets $S_1, S_2, \ldots, S_v$ responsible for the corresponding phenotypes, we define the outcome of a treatment for the ith phenotype as: $f_i(X) = \mathbb{I}[X \cap S_i \neq \emptyset]$. Therefore, $\mathbf{f}(X) = [f_1, f_2, \ldots, f_v]^T$ is a vector that represents the outcome for v phenotypes. For this problem, a naive method can still identify all the v agent sets using n treatments by testing each candidate one by one. Here, we define the global agent set $S = S_1 \cup S_2 \cup \cdots \cup S_v$, and treatment outcome function $f(X) = \mathbb{I}[X \cap \cup_i S_i \neq \emptyset]$. Effectively, a treatment has positive outcome if any phenotype is affected by an agent, and negative if no phenotype was affected.

Thus, this problem is equivalent to the agent identification for a single abstract phenotype defined as a logical OR of the multiple phenotypes under study.

5 Discussion

Efficient identification of agents among a large number of candidates is a challenging problem in biology and medicine. In this paper, we introduced an efficient online-group-screening framework that can 1) automatically decide the optimal strategy based on the outcome of a small set of pilot treatments, 2) requires less experimental rounds and total tests than the exhaustive method even in the worst case. We showed both the worst case savings and empirical average savings under different settings. Our method adapted to an existing dataset gives the expected time and cost savings. We also showed that this framework can be easily modified to solve related problems. We believe this method will make agent identification problems less expensive and much more efficient, and thus increase the pace of discovery.

Future work This approach can be further applied to the identification of more complex causal mechanisms, for example, involving interacting agents. If true interactions between candidates are few, we can still expect to save cost and time.

References

1. Stelios D Georgiou. Supersaturated designs: A review of their construction and analysis. *Journal of Statistical Planning and Inference*, 144:92–109, 2014.
2. Tonghui Ma, Jay R Thiagarajah, Hong Yang, Nitin D Sonawane, Chiara Folli, Luis JV Galietta, and AS Verkman. Thiazolidinone cftr inhibitor identified by high-throughput screening blocks cholera toxin–induced intestinal fluid secretion. *The Journal of clinical investigation*, 110(11):1651–1658, 2002.
3. Ricardo Macarron, Martyn N Banks, Dejan Bojanic, David J Burns, Dragan A Cirovic, Tina Garyantes, Darren VS Green, Robert P Hertzberg, William P Janzen, Jeff W Paslay, et al. Impact of high-throughput screening in biomedical research. *Nature reviews Drug discovery*, 10(3):188–195, 2011.
4. Zhong Wei, Tianjie Yang, Ville-Petri Friman, Yangchun Xu, Qirong Shen, and Alexandre Jousset. Trophic network architecture of root-associated bacterial communities determines pathogen invasion and plant health. *Nature Communications*, 6, 2015.
5. Charlie G Buffie, Vanni Bucci, Richard R Stein, Peter T McKenney, Lilan Ling, Asia Gobourne, Daniel No, Hui Liu, Melissa Kinnebrew, Agnes Viale, et al. Precision microbiome reconstitution restores bile acid mediated resistance to clostridium difficile. *Nature*, 517(7533):205–208, 2015.
6. George EP Box and R Daniel Meyer. An analysis for unreplicated fractional factorials. *Technometrics*, 28(1):11–18, 1986.
7. Ankit Gupta, Robert D Nowak, and Benjamin Recht. Sample complexity for 1-bit compressed sensing and sparse classification. In *ISIT*, pages 1553–1557, 2010.
8. Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.
9. Israel N Herstein. *Topics in algebra*. John Wiley & Sons, 2006.
10. Vašek Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.
11. Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.
12. Clyde A Hutchison, Ray-Yuan Chuang, Vladimir N Noskov, Nacyra Assad-Garcia, Thomas J Deerinck, Mark H Ellisman, John Gill, Krishna Kannan, Bogumil J Karas, Li Ma, et al. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.
13. Stéphane Compant, Brion Duffy, Jerzy Nowak, Christophe Clément, and Essaïd Ait Barka. Use of plant growth-promoting bacteria for biocontrol of plant diseases: principles, mechanisms of action, and future prospects. *Applied and environmental microbiology*, 71(9):4951–4959, 2005.
14. J Kevin Vessey. Plant growth promoting rhizobacteria as biofertilizers. *Plant and soil*, 255(2):571–586, 2003.
15. Ben Lugtenberg and Faina Kamilova. Plant-growth-promoting rhizobacteria. *Annual review of microbiology*, 63:541–556, 2009.
16. Stéphane Compant, Christophe Clément, and Angela Sessitsch. Plant growth-promoting bacteria in the rhizo- and endosphere of plants: their role, colonization, mechanisms involved and prospects for utilization. *Soil Biology and Biochemistry*, 42(5):669–678, 2010.