Pooled Screening for Synergistic Interactions Subject to Blocking and Noise

Kyle Li[1], **Doina Precup**[1], **Theodore J. Perkins**[1,2,3]*

1 School of Computer Science, McGill University, Montreal, Quebec, Canada, **2** Ottawa Hospital Research Institute, Ottawa, Ontario, Canada, **3** Department of Biochemistry, Microbiology and Immunology, University of Ottawa, Ottawa, Ontario, Canada

Abstract

The complex molecular networks in the cell can give rise to surprising interactions: gene deletions that are synthetically lethal, gene overexpressions that promote stemness or differentiation, synergistic drug interactions that heighten potency. Yet, the number of actual interactions is dwarfed by the number of potential interactions, and discovering them remains a major problem. Pooled screening, in which multiple factors are simultaneously tested for possible interactions, has the potential to increase the efficiency of searching for interactions among a large set of factors. However, pooling also carries with it the risk of masking genuine interactions due to antagonistic influence from other factors in the pool. Here, we explore several theoretical models of pooled screening, allowing for synergy and antagonism between factors, noisy measurements, and other forms of uncertainty. We investigate randomized sequential designs, deriving formulae for the expected number of tests that need to be performed to discover a synergistic interaction, and the optimal size of pools to test. We find that even in the presence of significant antagonistic interactions and testing noise, randomized pooled designs can significantly outperform exhaustive testing of all possible combinations. We also find that testing noise does not affect optimal pool size, and that mitigating noise by a selective approach to retesting outperforms naive replication of all tests. Finally, we show that a Bayesian approach can be used to handle uncertainty in problem parameters, such as the extent of synergistic and antagonistic interactions, resulting in schedules for adapting pool size during the course of testing.

Citation: Li K, Precup D, Perkins TJ (2014) Pooled Screening for Synergistic Interactions Subject to Blocking and Noise. PLoS ONE 9(1): e85864. doi:10.1371/journal.pone.0085864

Editor: Andrew R. Dalby, University of Westminster, United Kingdom

Received September 23, 2013; **Accepted** December 7, 2013; **Published** January 16, 2014

Funding: This work was supported by grants from the National Sciences and Engineering Research Council of Canada (to DP and to TJP) and the Ottawa Hospital Research Institute (TJP). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: tperkins@ohri.ca

Introduction

The complex machinery of the cell is capable of producing strong, unexpected interactions between its individual components or other factors. A prime example of this is the phenomenon of synthetic lethality [1]. A pair of genes is synthetically lethal if the deletion of either gene individually has no or minimal influence on the organism, yet the deletion of both kills the organism. Networks of such interactions have been shown to contain important information about pathway and process relationships between genes [2], and so discovering these interactions is of great interest. Another important example is the Yamanaka factors, a set of four genes (Oct-3/4, SOX2, c-Myc and Klf4) whose overexpression can transform differentiated cells back into a pluripotent state very much like that of embryonic stem cells [3,4]. This discovery has had numerous implications for stem cell research, including ready production of embryonic-like stem cells without the use of embryos, generation of patient-specific stem cells, and a greater understanding of the networks controlling stemness and differentiation more generally [5,6]. Notably, none of the four factors are individually sufficient to restore a stem-like state, and indeed, Yamanaka and colleagues discovered the four factors by simultaneously overexpressing 24 known stem cell-related factors—a simple, though quite effective, pooling strategy [3]. Interactions are also important in the pharmaceutical world. While adverse interactions are a well-known clinical problem [7], interactions can also be beneficial. Multi-component therapies, which rely upon synergistic interactions between individually ineffective or weak drugs, are increasingly being used to address complex diseases such as cancer, HIV/AIDS, diabetes, and immune disorders [8–11].

Discovering interactions can be difficult. One reason is the sheer number of interactions that are possible. Abstractly, if we have n "factors" which may interact, then there are $\binom{n}{2}=O(n^2)$ possible pairwise interactions, $\binom{n}{3}=O(n^3)$ possible three-way interactions, and so on. Often, the number of actual interactions is vastly smaller than the number of potential interactions. For instance, in the largest screen for interactions between pairs of yeast genes to date [2], approximately 3% of the 5.4 million pairs tested showed a significantly unexpected influence on growth rate, and only a fraction of those were synthetically lethal. Similarly low rates of unexpected interactions have been observed in the relatively few attempts at high-throughput pooled drug screening [11–14]. Thus, exhaustive testing for interactions requires significant effort and has a rather low success rate.

Another source of difficulty is that interactions between factors may be masked by other factors, variously called blockers, inhibitors or antagonists [15–17]. In drug screening, the presence of one compound, which itself does not affect the biological target, may nevertheless neutralize the positive effect of compounds with which it is combined [17]. Blocking has also been identified as a challenge in screening DNA libraries [15,18]. While we are not aware of genes whose expression blocks the reprogramming ability of the Yamanaka factors, it was recently shown that depleting Mbd3 greatly increases the efficacy of reprogramming—that is, the fraction of cells that return to a stem-like state [19]. Thus, Mbd3 is a strong, though not absolute, inhibitor of the Yamanaka factor synergy.

A further difficulty is that one always has to consider the possibility that a test may produce a false positive or false negative result (e.g. [20–24]). In high-throughput screens, both types of false results are common, and the experimental design must be able to account for such errors. A naive strategy is simply to replicate each test a fixed number of times, say r. This allows one to gain greater certainty in the results, reducing the chance of both false positives and false negatives. However, this strategy increases the experimental burden by a factor of r, which is often considered prohibitive. An alternative, and probably more common strategy, is to perform an initial screen and then conduct confirmatory testing only on the positive results from the screen. This allows one to eliminate false positives from further consideration, but it does not address false negatives at all.

In principle, pooled screening offers ways to address all three of the difficulties just mentioned. To introduce the idea of pooled testing, let us consider the seminal work of Dorfman [25], who discussed the problem of testing blood samples of potential military recruits for signs of syphilis. The test for syphilis was very sensitive. For reasonable pool sizes p (meaning p different blood samples are combined and then tested), a negative reading could be assumed to mean that none of the original samples were positive. However, a positive reading would mean that at least one of the original samples was positive. In this case, the individual samples would then need to be retested to identify precisely which recruits were infected. Dorfman showed that if the overall prevalence of syphilis is sufficiently small, so that relatively few pools are positive, then performing the pooled screen plus the positive-pool follow-ups can be far more efficient than testing each recruit's blood individually. Moreover, he showed how to select an optimal pool size based on the estimated prevalence of the disease.

Since Dorfman's work, the theory and practice of pooled screening has expanded enormously (see [14,26,27] for theory as well as pointers to many application areas). In the most standard formulations of screening problems, which omit synergy and antagonism, methods for dealing with testing errors range from simple grid-based schemes [27] to the recently-developed and powerful Shifted Transversal Design [11,24,28,29]. A basic principle of such designs is that any individual factor appears in multiple pools, reducing the possibility of false negatives. Indeed, screens can be designed to automatically correct for a bounded number of testing errors (either false positives or false negatives) even without follow-up testing, giving a guaranteed degree of robustness (e.g., [15, 24, 28]).

Error-resilient schemes also provide some protection against antagonism. For instance, imagine a high-throughput drug screen in which there is a particular active compound c. Compound c will be tested multiple times in combination with other compounds, and will fail to be detected only if every one of those pools contains an inhibitor. (These could be viewed as "false negatives", though the tests are really correct, given the presence of the unknown inhibitors.) A better approach, however, is to employ a design that explicitly addresses the possibility of inhibitors [15,16,30–33]. Intuitively, given a bound on the number of inhibitors, such pooling designs ensure that the active compound(s) (or positive factors) occur in enough different pools with non-inhibiting factors that their effects will be detected.

The majority of the work on pooled screening does not address the issue of synergy or interactions between factors, although even single-factor schemes can be bent to this purpose. One can hold a factor x constant and search a library of other factors y for interactions with x using a pooled screen. This approach has been used in screening DNA libraries [15] and yeast two-hyrbid screening [29,34], to name two examples. There are, however, explicit schemes for searching for synergistic groups, sometimes called complexes, among a library of factors [12,27,35]. The essential problem is to create a screen ensuring that all combinations up to a certain size appear in one or more pools (depending on one's requirement for error tolerance). Relatedly, there is work on threshold-testing problems where it is not necessarily a particular combination of factors that produces a reading, but positive readings come when enough positive factors are included in a pool—and potentially, not too many inhibitors [36–38].

In this paper, we consider all three issues of synergy, antagonism and noisy testing. To our knowledge, the only previous works to address all three issues simultaneously are those of Chang *et al.* [32] and Chang *et al.* [33]. These works propose non-adaptive screening designs—that is, a way of selecting a set of pools given: the number of factors n, bounds on the number and sizes of synergistic groups, a bound on the number of inhibitors, and a bound on the number of errors (false positive or false negative) that will occur during the screen. After the screen is performed, the test results can be analyzed to identify all the synergistic groups correctly, without additional testing.

We focus instead on adaptive designs. In general, an adaptive design is a scheme for choosing a sequence of pools to test, in which the choice of next pool is allowed to depend on the outcomes of the earlier tests. Such designs can be quite sophisticated. We will, however, explore rather simple randomized designs, along the lines of Farach *et al.* [15]. In our view and experience, simplicity of a design is a point in favor of adoption. Moreover, the randomized designs we analyze lend themselves to analytical tractability. In particular, we are able to resolve questions such as: What is the expected screen size—the number of tests that need to be performed—to discover synergistic combinations of factors? What is the optimal pool size? How does noise in test readings require the design to be changed, and how does it affect sample size and optimal pool size? How can we design a screen if we do not know how many factors may be interacting or how many factors may be blocking an interaction?

Our analysis also differs from most previous work, and in particular Chang *et al.* [32] and Chang *et al.* [33], in the manner that testing errors are modeled. Most analyses assume an absolute bound e on the number of errors that will occur during a screen. While this is better than assuming no errors at all, we consider it unrealistic that the number of errors e is independent of the size of the screen. Instead, we assume that each test has a fixed probability of producing an error. Under this assumption, no non-adaptive screening strategy can absolutely guarantee success—which is another motivation for our interest in adaptive designs. Conversely, even simple randomized adaptive screens can be guaranteed to eventually find synergistic groups with probability one, if they are allowed to proceed long enough.

Although our analysis is largely developed from the point of view of group testing theory, pooled screening problems can also be related to the theory of learning sparse Boolean functions [39–47]. In a generic version of this problem, we assume the existence of a Boolean function of interest $f:\{0,1\}^n \rightarrow \{0,1\}$. However, we assume that f really only depends on $m \ll n$ of the input variables. Identifying those m relevant inputs is therefore of great interest—sometimes of greater interest than identifying exactly how f depends on those inputs. We can relate sparse Boolean functions to the pooled screening context by saying that the i^{th} input feature is 1 if the i^{th} factor is included in the pool, and f just returns whether or not the pool tests as positive. In the standard pooled testing problem, without synergy and without blockers, a pool is supposed to read positive if any of m individually-positive factors is present in the pool. Thus, f is simply the disjunction of the m corresponding input features. In a formulation that allows for synergy, and assuming for simplicity that we seek a single m-way synergy, f is instead the conjunction of the m corresponding input features. If we additionally allow for k blockers, then f would be the conjunction of the m synergy features and the negation of the k blocking features. In the present study, we will generally assume that m is small compared to n; however, we will make no such assumption about k. As such, our problem does not technically fit within the assumptions of a sparse Boolean function learning problem. Nevertheless, as has been shown for sparse Boolean function learning [41,42,46–48], we will see that relatively few pools—even if selected randomly—are sufficient for identifying the m factors of interest.

Results

Randomized pooled screening guarantees discovery of synergistic combinations, despite blocking, and can vastly outperform exhaustive testing

We begin by analyzing a basic scenario with synergy and blockers. We assume there are n individual factors in which we are interested—genes, drug compounds, etc.—and that we can test these factors either individually or in combinations. Each test results in either a positive or a negative outcome. A positive outcome is the outcome of interest—for instance, synthetic lethality between two genes, or synergy between drugs. A negative outcome means the factor or combination of factors tested either had no effect or merely had the expected effect, and is therefore not of interest. As mentioned above, in screens for gene interactions related to yeast growth rates, just a few percent of potential interactions turn out to be real, and strong interactions are rarer yet. Similarly, in a typical drug screen, a few percent of individual compounds may have some effect on a biological target, while the number of synergistic combinations is expected to be quite small. Here, we make the pessimistic assumption that the set of n factors contains just a single synergistic combination of m factors that has a positive effect. Therefore, the goal of the screen is to find this particular combination. We call these factors the *desirable factors*. Establishing the utility and feasibility of pooled screening under this scenario implies that it would be all the more useful under less pessimistic conditions. In the Discussion section, we outline how our results can straightforwardly be extended to the case of multiple synergistic combinations. We also assume that there are k factors that are *blockers*. Whenever one of the blockers is present in a pool along with the desired factors, the positive effect is completely abolished. The remaining $n-m-k$ factors are *neutral*, and do not effect the outcome of the test. For now, we will assume that the parameters n, m, and k are all known—although of course the identities of the m desirables and k blockers are unknown. Later, we will lift this assumption, treating m and k as unknowns about which we maintain probabilistic beliefs that can change during the course of testing. We will focus first on the noise-free case, in which a pool of factors gives a positive reading if and only if it contains all m desired compounds, none of the k blockers, and any number of neutral factors.

The number of factors, n, may be large—hundreds or thousands for genes, and possibly into the millions for large drug screens. We expect m to be relatively small, as in the examples of synthetic lethality ($m=2$), the Yamanaka factors ($m=4$), or drug cocktails (typically $m=2\ldots4$). We also allow $m=1$, an individually-active factor, as a special case, although our primary interest is in identifying synergies between factors. The number of blockers, k, can vary widely.

Pooled experiment designs often have two phases. In the first phase, a collection of pre-chosen pools are tested. In the second phase, sometimes called the *decoding* phase, members of positive pools are tested further, either individually or in groups, to determine the cause of the positive reading. In the first phase, the main design choices concern the sizes of the pools and the method used to assemble each pool. For our initial model of the screening problem, we consider the screening design shown in Figure 1A. Pools of size p are drawn repeatedly, uniformly randomly, and with replacement, from the n factors. This sequence of pools is tested until a positive reading is achieved. Recent work on drug screening has shown that a biased random selection procedure, which aims to better cover a feature space of chemical descriptors, can improve the efficiency of such randomized screening [13]. However, for simplicity and analytical tractability, we focus on the more straightforward selection method. For the screening procedure in Figure 1A, as long as random pools of size $p \leq n-k$ are chosen, the procedure will eventually find a positive pool with probability one—simply because there is at least one positive pool of this size. Thus, the first phase terminates with probability one.

Once a positive reading is obtained, the second phase of the design is responsible for identifying the desirable factors from the positive pool. There are m desirables that need to be discovered in the pool of size $p \geq m$. A simple approach, requiring p tests, is to exclude each factor from the pool in turn and test the rest of the factors as a pool. If the reading is negative, then the factor that was excluded is one of the m desired factors. If the reading is positive, then the factor that was excluded is neutral and can be discarded from the pool. More generally, because the positive pool cannot contain blockers, one could use any scheme for identifying m-way synergies from a library of factors [12,26,27]—but in this case from just the p factors in the positive pool, rather than the entire library of n factors.

In the classic work of Dorfman [25], the optimal pool size is determined by trading off the costs of the first and second phases. In his formulation, larger pools increase the efficiency of the first phase, but tend to decrease the efficiency of the second phase, as more pools will be positive and their larger size means that more follow-up tests will be needed. In our study, the $\leq p$ tests required by the decoding phase are generally inconsequential compared to the number of tests needed in the first phase. So, the efficiency of the first phase is our primary concern. Larger pool sizes are favored by the desire to reduce the number of tests needed. In particular, larger pools test more factors simultaneously and test more possible synergies simultaneously. However, the presence of blockers favors smaller pools, so that positive readings are not masked. This tension determines the optimal pool size. In some cases, particularly when the number of blockers k is very small, the second phase of the design can take substantial testing effort compared to the first phase, and should not be ignored. We return to this issue in the Discussion section.

We begin by computing the expected time (i.e., number of tests) until a positive reading is found in the first phase. In order to get a positive reading for a particular test, we must assemble a pool that

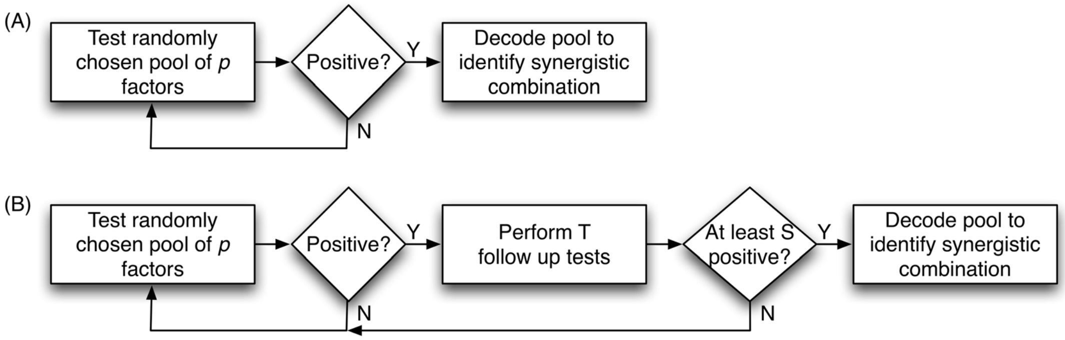
(A)
Test randomly chosen pool of *p* factors
Positive?
Y
N
Decode pool to identify synergistic combination
(B)
Test randomly chosen pool of *p* factors
Positive?
Y
N
Perform T follow up tests
At least S positive?
Y
N
Decode pool to identify synergistic combination

Figure 1. Sequential designs for pooled screening, in the case of noiseless tests (A), and with the possibility of noisy tests (B).
doi:10.1371/journal.pone.0085864.g001

contains all m desirable factors, $p-m$ neutral factors, and no blockers. The total number of pools of this sort is $\binom{n-m-k}{p-m}$, out of a total of $\binom{n}{p}$ possible pools of size p. Thus, the probability that a randomly selected pool of size p gives a positive reading is

$$P_{nmk}(p)=\frac{\binom{n-m-k}{p-m}}{\binom{n}{p}}. \tag{1}$$

As we have assumed that pools are drawn independently and tested sequentially until a positive reading is found, the time until obtaining a positive reading is just a geometric waiting time random variable, with success probability $P_{nmk}(p)$. Therefore, the expected number of tests as a function of p is

$$T_{nmk}(p)=\frac{\binom{n}{p}}{\binom{n-m-k}{p-m}}. \tag{2}$$

To give some intuition for the relationship between p and $T_{nmk}(p)$, Figure 2 shows $T_{nmk}(p)$ for varying values of n, m, k and p. In panel A, the three curves show $T_{nmk}(p)$ for the three cases: seeking a single active factor from a set of $n=10^6$, seeking a synergistically active pair of factors from a set of $n=1415$ candidates, and seeking a synergistically active trio of factors from a set of $n=183$ candidates. In each case, we assume 10% of the factors are blockers. These values were chosen because the straightforward approach of testing all $\binom{n}{m}$ subsets takes approximately one millions tests in each case—a feasible number by current, high-throughput methods. If we imagine running through those one million tests, but stopping as soon as the desirable combination is identified, then the *expected* testing effort of the naive, exhaustive screen is 500000 tests.

An immediate observation is that pooled testing can greatly increase the efficiency with which the desirable factor(s) are identified. For the $m=1$ case, testing at the optimal pool size of $p^*=9$ reduces the expected number of tests needed to 258119—a twofold reduction in testing effort. For $m=2$ and $m=3$, the optimal pool sizes of $p^*=19$ and $p^*=26$ reduce the expected number of tests to 35796 and 5175 respectively, corresponding to fourteen-fold and nearly 100-fold reductions in testing effort respectively. Achieving these improvements requires relatively large pool sizes. Pool sizes as large as 19 or 26 are feasible for gene overexpression or knockdown studies (as in [3]). They may or may not be feasible in drug screening, due to general toxicity effects; the study of Severyn *et al.* [11] used a pool size of ten. In any given case, there may be limits on how large a value of p is feasible. However, even if one sticks to a relatively modest $p=5$, the expected numbers of tests needed for the cases $m=1$, 2, and 3 go down to 304832, 137489 and 124118 respectively—a savings of 40% to 70%. In this simple scenario, then, we see that pooled testing has the potential for greatly increasing the efficiency of discovering desirable factors or combinations of factors.

Figure 2B shows the effect of varying the number of blockers, k, for $n=100$ and $m=2$. Increasing k has a dramatic effect on the expected number of tests as well as the optimal pool size. Qualitatively, both panels show that $T_{nmk}(p)$ appears to initially decrease with p and then increase. Intuitively, $T_{nmk}(p)$ decreases at small p because of the increased chance of including all the desirable factors in the pool, but it increases at larger p because of the increased chance of including blockers. (Shortly, we will show analytically that $T_{nmk}(p)$ is unimodal in all but a few degenerate situations.) This implies that there is usually a single, optimal choice of p, although the curves also show that there can be a significant range of values of p for which $T_{nmk}(p)$ is nearly as good as at the optimal pool size.

Finally, we note that if one takes $p=m$, the minimum pool size with which it is possible to discover an m-way synergy, the expected number of tests is just $T_{nmk}(m)=\binom{n}{m}$. This is exactly the same as exhaustively running through all possible m-way combinations—although it is twice the expected effort, if we assume that the exhaustive screen ends when the combination is found.

To determine the optimal pool size for given n, m and k, consider the ratio

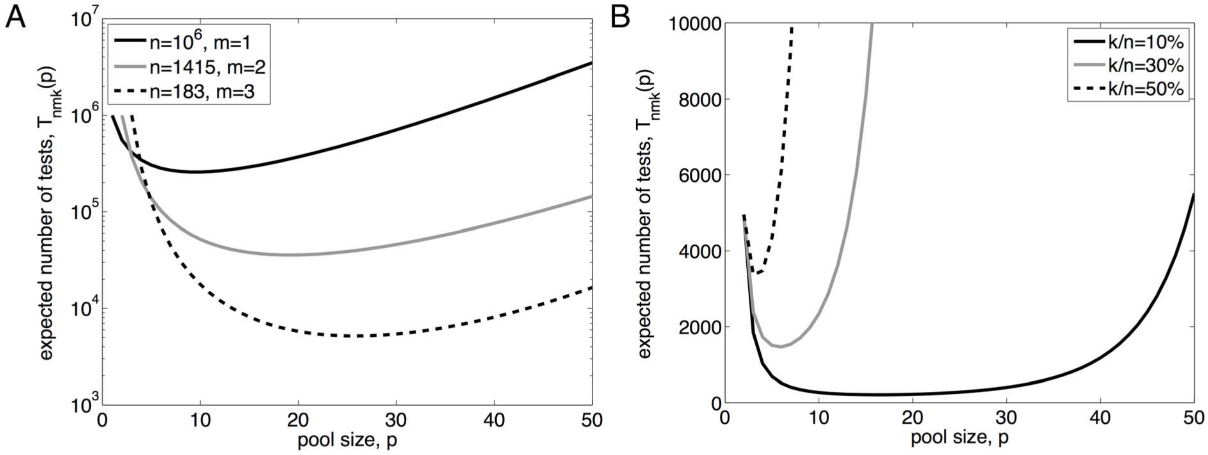

Figure 2. Expected number of tests in phase one, assuming noiseless tests, as a function of pool size, p. (A) Three different choices of library size, n, and synergistic group size, m, for which exhaustive combinatorial testing requires one million tests (or 500000 expected tests, assuming early stopping is allowed). In each case, the number of blockers is $k = \lceil n/10 \rceil$, or approximately 10% of the library. (B) Varying k, with $n = 100$ and $m = 2$.

doi:10.1371/journal.pone.0085864.g002

$$\frac{T_{nmk}(p+1)}{T_{nmk}(p)} = \frac{\dfrac{\binom{n}{p+1}}{\binom{n-m-k}{p+1-m}}}{\dfrac{\binom{n}{p}}{\binom{n-m-k}{p-m}}} = \frac{\dfrac{\dfrac{n!}{(p+1)!(n-p-1)!}}{\dfrac{(n-m-k)!}{(p+1-m)!(n-p-k-1)!}}}{\dfrac{\dfrac{n!}{p!(n-p)!}}{\dfrac{(n-m-k)!}{(p-m)!(n-p-k)!}}} = \frac{(p+1-m)(n-p)}{(p+1)(n-p-k)}. \tag{3}$$

If this ratio is less than one, then testing with pool size $p+1$ is more efficient than testing with pool size p. This leads to the following criterion.

$$\frac{T_{nmk}(p+1)}{T_{nmk}(p)} < 1 \Leftrightarrow \frac{(p+1-m)(n-p)}{(p+1)(n-p-k)} < 1 \Leftrightarrow p < \frac{mn-k}{m+k} \tag{4}$$

This confirms the observation (see Figure 2) that $T_{nmk}(p)$ is unimodal in p, decreasing as p increases to a certain threshold and then increasing afterwards. If $\frac{mn-k}{m+k}$ is an integer, then $p^* = \frac{mn-k}{m+k}$ is the optimal choice of p. If $\frac{mn-k}{m+k}$ is not an integer, then the smallest integer greater than that, denoted $\left\lceil \frac{mn-k}{m+k} \right\rceil$ is the optimal choice. In either case, we can write the optimal choice for p as

$$p^* = \arg\min_p T_{nmk}(p) = \left\lceil \frac{mn-k}{m+k} \right\rceil \tag{5}$$

Equation 5 always yields a valid choice for p that falls in the range $[m, n-k]$ (see Materials and Methods for proof). This ensures termination of the screening procedure with probability one. It is possible that $p^* = \left\lceil \frac{mn-k}{m+k} \right\rceil = m$, so that the optimal pool size is no larger than the minimum necessary to discover an m-way synergy. In particular, this can happen when the number of blockers, k, is very large. For example, in the case $n = 100$ and $m = 2$, the optimal pool size is $p^* = 2$ when $k \geq 66$. In such cases, $T_{nmk}(p)$ is monotonically increasing in p. The case $k = 0$ is a degenerate situation in which $T_{nmk}(p)$ is monotonically decreasing in p and $p^* = \left\lceil \frac{mn}{m} \right\rceil = n$. As stated above, optimal pool size for us is determined by a tradeoff between library and synergy size, favoring large pools, and the number of blockers, favoring small pools. In the absence of blockers, there is really no need for phase one—the primary purpose of which can be viewed as identifying a pool or sub-library that still contains the synergistic group, but without any blockers. In this case, approaches for finding synergistic combinations without blockers should be employed [35].

Figure 3A shows p^* as a function of m and k, for $n = 100$. Optimal pool size drops rapidly at low k, and plateaus at $p^* = m$ for high values of k. Figure 3B shows the expected number of tests for various m and k if the optimal pool size is used. The qualitative shape of the curves for $T_{nmk}(p^*)$ can be understood by analyzing the ratio $T_{nmk+1}(p)/T_{nmk}(p)$.

$$\frac{T_{nmk+1}(p)}{T_{nmk}(p)} = \frac{\dfrac{\binom{n}{p}}{\binom{n-m-k-1}{p-m}}}{\dfrac{\binom{n}{p}}{\binom{n-m-k}{p-m}}} = \frac{n-m-k}{n-p-k} \tag{6}$$

For fixed p, T_{nmk} is non-decreasing in k, because $n-m-k \geq n-p-k$. The ratio is greatest when p is large. If we

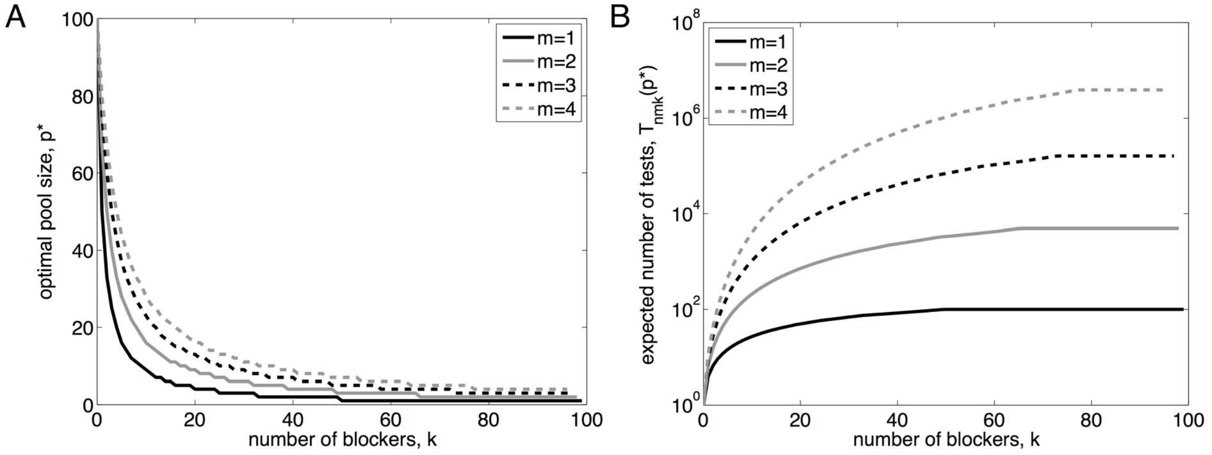

Figure 3. Optimal pool size and testing complexity. (A) The optimal pool size, p^*, as a function of the number of blockers, k, for varying numbers of desirable factors, m. (B) The corresponding expected number of tests needed to identify the desirable factors. These curves are for library size $n = 100$.

doi:10.1371/journal.pone.0085864.g003

consider p^* instead of a fixed p, then intuitively, p^* is largest when k is small, so we expect $T_{nmk}(p^*)$ to be increasing fastest (in terms of ratio) when k is small and to level off at large k, when $p^* = m$.

Testing errors do not change optimal pool size, and are best handled by follow-up testing on positive pools

In realistic situations, test results, especially from high throughput methods, may be erroneous. In such a scenario, a positive reading no longer guarantees that we have the desired combination in the pool. Similarly, a negative reading does not imply that we failed to have the desired combination and no blockers in the pool. In this section, we assume that the test has a fixed probability $0 \le e < 1/2$ of producing an erroneous reading, either a false positive or a false negative. We assume the same error rate for both positive and negative readings for simplicity, but the case of different error rates is a straightforward extension.

In this new setting, a positive reading arises either from a truly positive pool with a correct reading (true positive) or from a truly negative pool with an incorrect reading (false positive). Let p^+ be the event that a randomly drawn pool is truly positive, p^- be the event that a randomly drawn pool is truly negative, and t^+ the event that a randomly drawn pool tests as positive. Then, the probability of getting a positive reading for a randomly chosen pool of size p is

$$\begin{aligned} P_{nmke}(p) &\overset{\Delta}{=} P(t^+) \\ &= P(p^+)(1-e) + P(p^-)e \\ &= P_{nmk}(p)(1-e) + (1-P_{nmk}(p))e \\ &= (1-2e)P_{nmk}(p) + e. \end{aligned} \tag{7}$$

Note that $P_{nmk}(p) = P(p^+)$ is the probability of drawing a truly positive pool, just as in the previous section.

To accommodate for testing errors, one generally uses replicates to maintain a certain level of confidence. There are many ways to incorporate replicates. The most straightforward scheme is to repeat every test some fixed number of times. In a subsequent phase, tests that are "sufficiently positive" are followed up. (Sufficiently positive may mean that all or most of the replicates test positive, though there are other ways to combine multiple tests, depending on the setting.) In the present setting, however, it turns out that replicating the tests on every pool is inefficient. Because the vast majority of tests are expected to be negative, replicating all those tests to be sure that they are negative wastes significant effort.

We propose an alternative, as shown in Figure 1B. We start with random pool selection and repeat until we obtain a first positive reading. This pool could be a false positive, so we perform T replicate tests on the same pool. A confidence level $S \in (T/2, T]$ is defined such that if there are at least S positive readings out of the T replicates, we consider the pool to be a true positive, and we proceed to the decoding phase (whose testing effort we ignore, as it is generally small). If, however, the T replicates do not contain S positive readings, we consider that the initial positive reading was a false positive, and we continue testing randomly drawn pools. Note that, in this scheme, it is possible that a truly positive pool is read as negative. In this case, the procedure simply continues to draw random pools of factors for testing. Although an occurrence of a positive pool may be "missed", eventually, the procedure should draw a truly positive pool that tests as positive, both initially and in at least S of the T confirmation tests.

The expected number of tests required by this procedure obeys

$$T_{nmke}(p) = \frac{1}{P_{nmke}(p)} + T + (1 - P(s^+|t^+))T_{nmke}(p) \tag{8}$$

where $P(s^+|t^+)$ is the probability that a pool that tests positive in its first test also tests positive in at least S of the T follow-up tests—that is, it is confirmed as a truly positive pool. The rationale behind this recurrence is simply that the total expected number of tests can be decomposed into the number of tests until obtaining the first positive reading, the automatic T follow-up tests, and, if the follow-up tests do not confirm that the pool is a true positive, the expected tests from then on (which, as the process is memoryless, is the same as at the start of the process). The expected number of tests is thus

$$T_{nmke} = \frac{1}{P(s^+|t^+)}\left(\frac{1}{P_{nmke}(p)} + T\right) \tag{9}$$

We will shortly consider the behavior of T_{nmke} for various parameter values, but first let us derive the optimal pool size. The probability $P(s^+|t^+)$ depends on the pool size, even though the notation does not make it explicit. Thus, the above formula for T_{nmke} depends on p in two different places. Remarkably, it turns out that the optimal pool size is the same as for the noise-free case. In fact, that same pool size minimizes both the $\frac{1}{P(s^+|t^+)}$ term and the $\frac{1}{P_{nmke}(p)}$ term simultaneously. The latter claim follows readily from Equation 7, keeping in mind that $e<\frac{1}{2}$, so that $1-2e$ is positive. The claim that the same p^* also minimizes $\frac{1}{P(s^+|t^+)}$ takes more effort to show. We relegate its proof to the Materials and Methods.

Figure 4A shows the expected number of tests under the noisy-testing model, for $n=100$, $e=0.1$, $T=20$, $S=15$, and varying m and k, assuming the optimal pool size is used. With these choices of S and T, if a pool initially tests positive but is actually a negative pool, the chance of erroneously confirming it as a positive pool is approximately 10^{-11}. The chance of failing to confirm a truly positive pool is approximately 1%; if this did happen, of course, the procedure would continue to look for a positive pool. Hence, with very high probability, the procedure finishes by identifying a truly positive pool. Qualitatively, the expected number of tests is very similar to the expectation under the noise-free model (see Figure 3B). Figure 4B shows the ratio of the expected number of tests under the noisy model to the expectation under the noise-free model, for varying error rates, e, $k=10$, and with all other parameters the same. Naturally, when the error rate is small, little extra effort is involved. In the case $m=1$, there is about 75% extra testing effort, which is almost wholly due to the $T=20$ follow up tests. As the error rate increases, so does the testing effort ratio, reaching a value between three and four for these parameter settings. A more standard approach to dealing with the possibility of noisy readings is to replicate each test some fixed number of times, usually at least three. Even at a high level of noise, $e=0.1$, the testing scheme we propose is approximately as efficient as naive triplicate testing. Further, our scheme has the added benefit of producing a correct answer with very high probability. In contrast, the naive replication scheme has a nontrivial chance of missing the desirable combination, and is almost certain to return a large number of false positives. For instance, if we require all three replicates to be positive, then the truly synergistic combination has $1-(1-e)^3$ chance of being called positive. With $e=0.1$, there is a 27% chance that it will not be correctly identified. At the same time, the expected number of false positives is $\binom{n}{m}e^3$. For instance, with $n=100$, $m=2$, and $e=0.1$ we expect 5 false positives, and with $m=3$ we expect 162 false positives. If we would require only 2 of 3 replicates to test positive, our chance of detecting the truly synergistic combination increases, but so does the expected number of false positives.

Bayesian adaptive scheduling when the numbers of synergistic and/or blocking factors are unknown

So far, we have assumed that all the parameters of the problem (n, m, k and e) are known. In reality, this is usually not the case. Of course, the total number of factors n, is typically known. Often, the error rate of the assay, e, has been established based on calibration testing or previous screens. The value of m may well be unknown, though it is generally expected to be relatively small, say between 1 and 4. We expect that the number of blockers, k, will often be unknown and that we can expect significant uncertainty about its value.

With unknown m and/or k, there are several ways to proceed. We could optimistically assume that $m=1$ and $k=0$. However, if our assumption is violated, we may find ourselves with an endless stream of negative results and no way to explain them. On the other hand, we could pessimistically assume that m is large (say, four) and that $k=n-m$. In this case, we would only test pools of size m, but this approach misses out on much potential gain in efficiency by using larger pools. Furthermore, such a situation is not what we expect in practice. We are left, then, with m likely being small and k being somewhere between 0 and $n-m$.

In this section, we explore a Bayesian approach to handling uncertainty about the true values of m and k. We assume that before screening begins, there is a prior belief $P_0(m,k)$ that represents our estimate of the chance that there are m desirable factors and k blockers. As we will show shortly, these beliefs can be updated as screening proceeds. For simplicity, we assume noise-free testing ($e=0$), although the following can be generalized to the noisy testing case. With noise-free testing, if we obtain a positive pool, then we move to the decoding phase and our belief over m and k becomes irrelevant. As long as we continue to get negative readings, however, we can update our belief about m and k.

Suppose that for the t^{th} test, we choose a random pool of factors of size p_t. Below we describe several possible schemes for choosing the pool sizes as a function of t. Let N_t denote the event that the first t tests come out negative. Let $P_t(m,k)$ denote our belief over m and k after t negative tests. The following equation describes how we can update the belief as each negative test result comes in. (Again, if we get a positive test, then we can discard our beliefs and proceed to the decoding phase.)

$$\begin{aligned}
P_t(m,k) &= P(m,k|N_t)\\
&= P(m,k|N_t,N_{t-1})\\
&= \frac{P(N_t|m,k,N_{t-1})P(m,k|N_{t-1})}{P(N_t|N_{t-1})}\\
&= \frac{P(N_t|m,k,N_{t-1})P(m,k|N_{t-1})}{\sum_{m',k'} P(N_t,m',k'|N_{t-1})}\\
&= \frac{P(N_t|m,k,N_{t-1})P(m,k|N_{t-1})}{\sum_{m',k'} P(N_t|m',k',N_{t-1})P(m',k'|N_{t-1})}\\
&= \frac{(1-P_{nmk}(p_t))P_{t-1}(m,k)}{\sum_{m',k'} (1-P_{nm'k'}(p_t))P_{t-1}(m',k')}
\end{aligned} \tag{10}$$

Because all the terms on the right hand side are known, this shows how the beliefs at time t depend on the beliefs at time $t-1$.

This leaves the question of how to choose the sizes of pools to test. For given n, p^* can be viewed as a function of m and k. As m and k are unknown, we propose the expedient of choosing p_t based on our beliefs after the first $t-1$ tests, $P_{t-1}(m,k)$. Specifically, we propose the pool size should be chosen by averaging the optimal pool size over the unknown parameters, rounding as necessary.

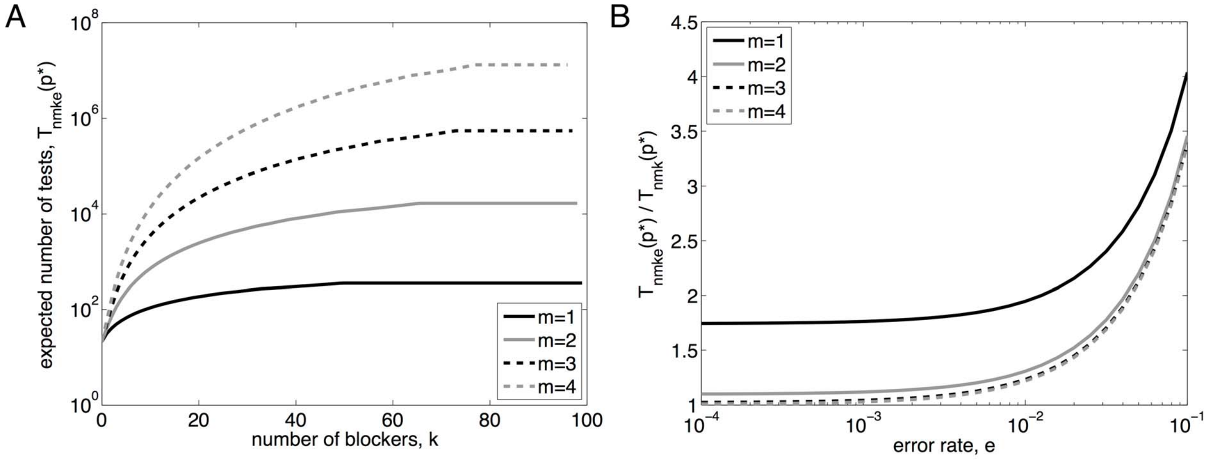

Figure 4. Number of tests needed, if one allows for the possibility of noisy test outcomes. (A) The expected number of tests as a function of the number of blockers, k, and the number of desirable factors, m, for $n=100$ and noise level $e=0.1$, assuming screening with optimal pool size. (B) The ratio of the number of tests expected under the noisy model to the number expected under the noiseless model, for varying error rates e, with $k=10$.

doi:10.1371/journal.pone.0085864.g004

$$\begin{aligned} p_t &= \mathrm{round}\left(\sum_{m,k} P_{t-1}(m,k)p^*(n,m,k)\right) \\ &= \mathrm{round}\left(\sum_{m,k} P_{t-1}(m,k)\left\lceil\frac{mn-k}{m+k}\right\rceil\right) \end{aligned} \tag{11}$$

Certainly, other choices are possible. For example, we might choose p_t to minimize the expected number of tests, assuming no further change in pool size.

$$p_t = \arg\min_p \sum_{m,k} P_{t-1}(m,k)T_{nmk}(p) \tag{12}$$

Or, we might choose p_t based on the expected values of m and k, or some percentiles of their distribution. We leave a detailed exploration of such strategies for future work, and limit our attention to Equation 11 for determining pool sizes.

As an example, suppose that $n=100$ and that we know $m=2$, but that the number of blockers is uncertain. Figure 5A plots the sequence p_t in two different cases. For one, we assume a uniform initial belief over k: $P_0(k)=\dfrac{1}{n-m+1}$ for $0\le k\le n-m$. For the other case, we assume that each non-desirable factor has a 10% chance of being a blocker, leading to a binomial belief, $P_0(k)=\binom{n-m}{k}0.1^k 0.9^{n-m-k}$. The figure also shows $E_t(k)$, the expected value of k according to the belief distributions, as testing progresses. The figure shows that, in either case, the expected value of k increases as testing progresses. Intuitively, this is because a large number of negative test results are more likely if k is higher. As a result, the suggested pool size drops as testing progresses, because larger (estimated) values of k favor smaller pool sizes. Formally, this can be derived readily from Equation 10, using the fact that $P_{nmk}(p)$ is non-increasing in k. For the uniform initial belief, $E_t(k)$ stops changing shortly after the 6000^{th} trial. This is because $p_t=m$ at the point. In this case, the probability of a positive test, $P_{nmk}(m)$, is the same regardless of k, so we gain no new information about the value of k.

Figure 5B shows the expected number of tests for each of these pool-size sequences, for different possible true values of k. The horizontal line shows the expected number of tests under a naive exhaustive screen, which for $n=100$ and $m=2$ involves $\binom{100}{2}/2=2475$ tests. The thicker solid curve shows the number of tests taken by the pool-size schedule induced by a uniform belief for k. If k truly is small, this strategy requires less than 20% as much testing as the naive strategy, and there are some savings even if the true k is as large as 35. The dotted line shows the tests expected under the fixed choice $p=8$, the initial pool size suggested by a uniform belief for k. For k up to 33 it actually does better than the adaptive Bayesian strategy, because it persists in using the larger, beneficial pool size $p=8$, whereas the Bayesian strategy switches to a lower pool size if testing runs long enough. However, for large values of k, the Bayesian strategy does better by a large margin, because it correctly deduces the benefit of the smaller pool size. The dashed lines show the expected number of tests using the Bayesian strategy with the binomial initial belief over k, as well as the fixed strategy $p=18$, which is the optimal pool size for the initial belief. If the true k is close to what is predicted by the binomial belief, then these strategies perform dramatically better than the naive screen, and modestly better than the uniform prior. On the other hand, if the initial belief is far from correct, then these strategies perform disastrously, because they persist in using a pool size that is far too large. As with the uniform initial belief, the fixed choice $p=18$ outperforms the Bayesian adaptive strategy for the smallest values of k (though the difference is slight), and the adaptive strategy does better for larger values of k.

In Figure 6A, we consider the case that $n=100$ and $k=10$, but that m is unknown. We explore two cases for the initial belief over m: a uniform distribution on $1\le m\le 4$, and a geometric distribution: $P_0(m)=2^{-m}$. The figure shows that, as testing proceeds, the belief distributions for m shift towards higher values, as higher values of m are a more likely cause of large numbers of negative tests. In this case, the pool size increases during the course of testing, as larger values of m imply larger optimal pool sizes. In Figure 6B, we consider the case that $n=100$ but neither m nor k are known with certainty. We assume an initial belief that is uniform over all combinations of m and k with $1\le m\le 4$ and

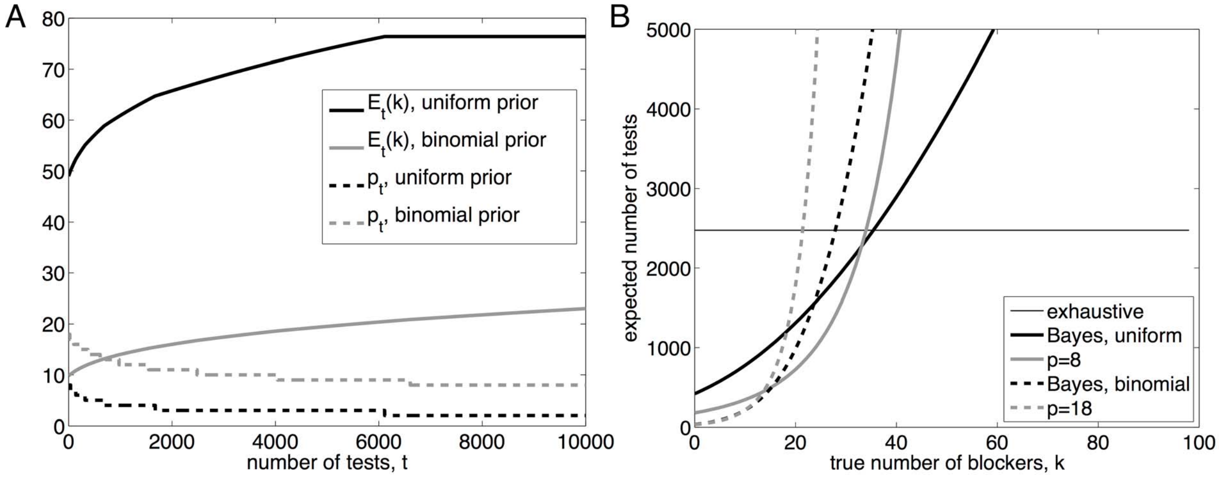

Figure 5. Pooled screening with uncertainty in the number of blockers. (A) The evolution of the expected value of k with respect to the belief over k, along with the pool size schedule p_t, as testing proceeds. Repeated negative tests increase the belief that k is larger, driving down the recommended pool size. (B) The expected number of tests until discovering the desirable combination of factors, for several fixed pool-sizes and for the Bayesian adaptive strategy. These results assume a set of $n=100$ factors from which we seek a synergistic pair ($m=2$)."Uniform" refers to the assumption that the belief about k is initially uniform on the range $[0, n-m]$."Binomial" refers to the assumption that the initial belief about k is binomial($n-m$, 0.1).
doi:10.1371/journal.pone.0085864.g005

$0 \leq k \leq n-m$. As the number of tests increases, the belief distributions for both m and k shift towards larger values. The increase for m is difficult to see in the figure, but the expected value changes from approximately 2.5 to 3.1 over the course of 100000 negative tests. The increasing estimates of m and k place competing forces on p_t. The former tends to increase the optimal pool size while the latter tends to decrease it. For the particular choices made here regarding n and the initial belief, the pool size turns out to decrease until roughly trial 71000, at which point it starts switching back and forth between six and seven. This continues until roughly trial 96000, at which point the pool size stays at six up to the 100000^{th} trial.

Discussion

We have explored several closely-related models of pooled screening, allowing for both synergy and antagonism between factors. A basic finding is that pooled testing can have significant benefits in terms of efficiency compared to naive one-at-a-time testing of factors or exhaustive combinatorial testing, as has been found in many other studies (see Hughes-Oliver [14,26,27] for a review). This is true even with such impediments as high levels of potential antagonism, noisy readings, or uncertainty in key problem parameters. We derived formulae for the theoretically optimal choice of pool size, showing, to our surprise, that it is unaffected by noise in the testing procedure—although noise does

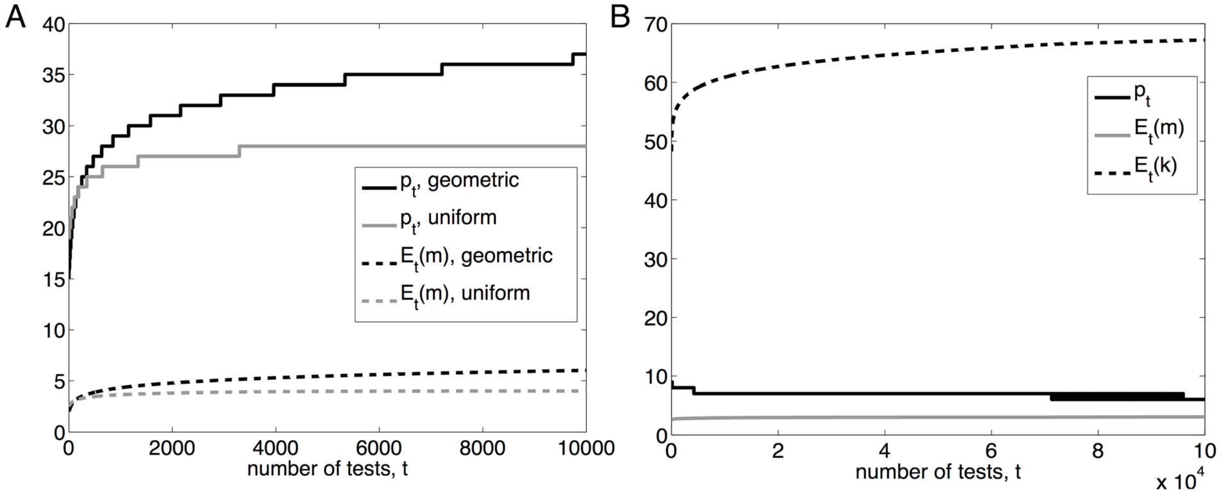

Figure 6. Pooled screening with uncertainty in the number of desirable factors and blockers. (A) When the number of blockers, k, is known but the number of desirable factors, m, is not known, increasing numbers of negative tests are evidence for higher values of m, which raises the optimal pool size. (B) The situation in which both m and k are unknown. Beliefs for both m and k shift towards larger values over time (though it is difficult to see the increase in $E_t(m)$ on this scale), resulting in competing forces on the ideal pool size.
doi:10.1371/journal.pone.0085864.g006

necessitate follow-up testing to confirm results. We also argued that, in the case of noisy testing, it is better not to test every pool in replicate, but only to retest the positive pools. This means that some truly positive pools may pass by as false negatives. Nevertheless, the testing effort involved in being certain about the many truly negative pools is not justified in a high-throughput screening scenario. It is better to wait for a subsequent pool to test positive.

Throughout the paper, we assumed that the pool size should be chosen to minimize the testing cost of phase one—the discovery of a positive pool—and ignored the testing cost of phase two—identification of the synergistic combination from the positive pool. This choice allowed us to derive analytical formulae for the expected number of tests at a fixed pool size and for the optimal pool size. This choice is also justified because in many cases the phase one cost greatly exceeds the phase two costs. However, this is not always the case, especially when the number of blockers is small. For instance, consider searching for a single active compound in a library of 100 factors, only one of which is a blocker. By our memoryless design, the optimal pool size is 50, and allows identification of a positive pool in 3.96 expected tests. By information-theoretic reasoning, subsequently identifying the responsible factor must involve at least $\log_2 50 = 5.64$ tests. Thus, the second phase has higher expected testing cost than the first phase. A more complete approach would consider the costs in both phases. Although explicit formulae might not be possible to derive, numerical computations could readily be performed to determine the optimal pool size.

We have also assumed that the goal of screening is to identify a single m-way synergistic combination. Our results can be extended to discovering a fixed number c of such combinations by running the proposed scheme repeatedly, until the c distinct combinations are discovered. We omit a detailed analysis. Intuitively, for a given pool size, each randomly sampled pool would have c times the chance of being positive as would be the case with a single synergistic combination. Thus, the first combination would be discovered c times as quickly as in the case of a single combination. The second combination must be one of the $c-1$ remaining combinations, and so would happen $c-1$ times as quickly; and so on. The total expected number tests would thus be approximately a factor of $\frac{1}{c} + \frac{1}{c-1} + \ldots + 1 \approx \log c$ larger than the number of tests required for discovering a single combination. The optimal pool size should not change, because it should be chosen to maximize the probability that any individual randomly-chosen pool is a true positive, regardless of how many synergistic groups there are.

In the memoryless randomized policies we have studied, the pool size is part of the experimental design, but the actual factors chosen to constitute each pool are not—they are simply drawn uniformly randomly from the set of factors. As shown in Remlinger *et al.* [13], for example, biasing this random choice with prior information about the factors can increase the success of screening. In that paper, synergistic effects were modeled but treated as undesirable—a false positive reading. Nevertheless, it would be interesting to try to incorporate some notion of biased random sampling into our analyses, to see if results with similar flavor hold. For that matter, simply keeping track of which pools have already been tested, and favoring pools that have not yet been tested, would likely reduce expected testing effort.

The efficacy of pooling depends strongly on the number of blockers—a parameter which we expect would often be hard to know or even estimate a priori. We showed that even if this parameter is unknown, one can adopt a Bayesian view and maintain a belief state over the number of blockers, which is updated as testing proceeds. The pool size can then be chosen based on this belief. Of course, if one's belief does not match reality at all, poor performance—worse than choosing the minimal pool size—is possible. However, we showed through numerical calculations that for a wide range of the unknown number of blockers, a Bayesian choice of pool size can greatly outperform choosing the minimal pool size.

We also showed that the size of the synergistic combination, m, if unknown, can be treated in a Bayesian manner. The procedure we described could also be extended to account for testing errors and even unknown error rates, by maintaining a belief over the error rate parameter. A much greater extension would be to treat the identities of the desirable and blocking factors themselves in a Bayesian manner. If we dispense with the m and k parameters, and imagine that each factor is either desirable, neutral, or blocking, then there are 3^n possible ground truths. In principle, we can imagine maintaining beliefs over these 3^n possibilities, and using techniques from the theory of partially observable Markov decision processes [49] to determine an optimal screening strategy. Exact methods would likely be too computationally intensive to apply to this problem, but it would be interesting to see if approximate methods could be applied to generate superior screening designs.

Although genetic and drug interactions were the primary motivations for our work, other application areas could be addressed by the same ideas. While we have already mentioned yeast two-hybrid screening for protein-protein interactions [24,29], yeast one-hybrid screening for protein-DNA interactions [50] might benefit similarly. Likewise, looking for cancer therapeutics based on small interfering RNAs [51] might benefit from pooling, under the assumption that multiple interfering RNAs are needed to down-regulate the multiple pathways that become mis-regulated in cancer. Recombinant congenic experiments, in which genetically healthy and genetically diseased animals are first cross-bred and then inbred to isolate complex genetic causes of disease, also have a strong flavor of pooled screening [52]. Some of the ideas in this paper might provide novel and useful views of these other types of screening procedures.

Materials and Methods

Feasibility of p^*

In the first part of our results, where we analyzed a noise-free model of the screening problem, we claimed that the formula for p^*, namely $p^* = \left\lceil \frac{mn-k}{m+k} \right\rceil$, always produces a value in the range $[m,n-k]$. In the case $k=0$, this is immediately true. Otherwise, for $k>0$, the following shows that p^* is always at least as large as m:

$$n \geq m+k$$

$$\Rightarrow n > m+k-1$$

$$\Rightarrow mn > m^2 - m + mk$$

$$\Rightarrow mn - k > m^2 - m + mk - k$$

$$\Rightarrow mn - k > (m-1)(m+k)$$

$$\Rightarrow \frac{mn-k}{m+k} > m-1$$

$$\Rightarrow \left\lceil \frac{mn-k}{m+k} \right\rceil \geq m$$

On the other hand, we can see that p^* is no larger than $n-k$ as follows:

$$0 \leq n-m-k$$

$$\Rightarrow -1 < n-m-k$$

$$\Rightarrow -k < nk-mk-k^2$$

$$\Rightarrow mn-k < mn+nk-mk-k^2$$

$$\Rightarrow mn-k < (m+k)(n-k)$$

$$\Rightarrow \frac{mn-k}{m+k} < n-k$$

$$\Rightarrow \left\lceil \frac{mn-k}{m+k} \right\rceil \leq n-k$$

Probability of confirming a positive test in the noisy model

In our analysis of screening with noisy testing, we claimed that the pool size p^* that maximizes the probability of choosing a truly positive test, $P(p^+) = P_{nmk}(p)$, also maximizes the probability of a test being confirmed as positive under the noisy testing model, $P(s^+|t^+)$. Here, we prove that claim.

$$\begin{aligned} P(s^+|t^+) &= P(s^+|p^+)P(p^+|t^+) + P(s^+|p^-)P(p^-|t^+) \\ &= P(s^+|p^+)\frac{P(t^+|p^+)P(p^+)}{P(t^+)} + P(s^+|p^-)\frac{P(t^+|p^-)P(p^-)}{P(t^+)} \\ &= \frac{P(s^+|p^+)P(t^+|p^+)P(p^+) + P(s^+|p^-)P(t^+|p^-)P(p^-)}{P(t^+|p^+)P(p^+) + P(t^+|p^-)P(p^-)} \\ &= \frac{P(s^+|p^+)(1-e)P_{nmk}(p) + P(s^+|p^-)e(1-P_{nmk}(p))}{(1-e)P_{nmk}(p) + e(1-P_{nmk}(p))} \end{aligned} \quad (13)$$

In the formula above, $P(s^+|t^+)$ depends on the pool size, p, only through the $P_{nmk}(p)$ terms. Although it may not be immediately obvious, this formula is increasing as a function of $P_{nmk}(p)$. To show this, replace $P_{nmk}(p)$ with the variable x in Equation 13, obtaining:

$$P(s^+|t^+) = \frac{P(s^+|p^+)(1-e)x + P(s^+|p^-)e(1-x)}{(1-e)x + e(1-x)} \quad (14)$$

We treat x as a real variable in the range (0,1), and differentiate with respect to it.

$$\begin{aligned} &\frac{\partial P(s^+|t^+)}{\partial x} \\ &= \frac{((1-e)P(s^+|p^+) - eP(s^+|p^-))((1-2e)x+e) - (P(s^+|p^+)(1-e)x + P(s^+|p^-)e(1-x))(1-2e)}{((1-2e)x+e)^2} \\ &= \frac{(e-e^2)(P(s^+|p^+) - P(s^+|p^-))}{((1-2e)x+e)^2} \end{aligned} \quad (15)$$

Because $0 < e < \frac{1}{2}$, we have $e - e^2 > 0$. So, the first factor in the numerator is positive. $P(s^+|p^+)$ is the probability of obtaining at least S successes out of T tries, where each try succeeds with probability $1-e$. $P(s^+|p^-)$ is the probability of at least S successes when the probability of each success is only $e < 1-e$. Thus, the former is larger, and the second factor in the numerator above is also positive. This means that $P(s^+|t^+)$ is strictly increasing as a function of $x = P_{nmk}(p)$. Therefore, choosing p that maximizes $P_{nmk}(p)$ also maximizes $P(s^+|t^+)$.

Acknowledgments

We thank Joelle Pineau for useful comments on an earlier version of this work, and Eric Brown for insightful discussions of pooled drug screening.

Author Contributions

Conceived and designed the experiments: KL DP TJP. Performed the experiments: KL DP TJP. Analyzed the data: KL DP TJP. Wrote the paper: KL DP TJP.

References

1. Tong AHY, Evangelista M, Parsons AB, Xu H, Bader GD, et al. (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. Science 294: 2364–2368.
2. Costanzo M, Baryshnikova A, Bellay J, Kim Y, Spear ED, et al. (2010) The genetic landscape of a cell. Science 327: 425–431.
3. Takahashi K, Yamanaka S (2006) Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. Cell 126: 663–676.
4. Takahashi K, Tanabe K, Ohnuki M, Narita M, Ichisaka T, et al. (2007) Induction of pluripotent stem cells from adult human fibroblasts by defined factors. Cell 131: 861–872.
5. Park IH, Arora N, Huo H, Maherali N, Ahfeldt T, et al. (2008) Disease-specific induced pluripotent stem cells. cell 134: 877–886.
6. Puri MC, Nagy A (2012) Concise review: embryonic stem cells versus induced pluripotent stem cells: the game is on. Stem Cells 30: 10–14.
7. Baxter K, Sharp JM (2008) Adverse drug interactions. Adverse Drug Reaction Bulletin 248: 951–954.
8. Fitzgerald J, Schoeberl B, Nielsen U, Sorger P (2006) Systems biology and combination therapy in the quest for clinical efficacy. Nature Chemical Biology 2: 458.
9. Keith C, Borisy A, Stockwell B (2005) Innovation: Multicomponent therapeutics for networked systems. Nature Reviews Drug Discovery 4: 71–78.
10. Zimmermann G, Lehar J, Keith C (2007) Multi-target therapeutics: when the whole is greater than the sum of the parts. Drug Discovery Today 12: 34–42.
11. Severyn B, Liehr RA, Wolicki A, Nguyen KH, Hudak EM, et al. (2011) Parsimonious discovery of synergistic drug combinations. ACS chemical biology 6: 1391–1398.

12. Borisy A, Elliott P, Hurst N, Lee M, Lehar J, et al. (2003) Systematic discovery of multicomponent therapeutics. Proceedings of the National Academy of Sciences 100: 7977–7982.
13. Remlinger K, Hughes-Oliver J, Young S, Lam R (2006) Statistical Design of Pools Using Optimal Coverage and Minimal Collision. Technometrics 48: 133–143.
14. Hughes-Oliver J (2006) Pooling experiments for blood screening and drug discovery. In: Dean A, Lewis S, editors, Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics, Springer New York. pp. 48–68.
15. Farach M, Kannan S, Knill E, Muthukrishnan S (1997) Group testing problems with sequences in experimental molecular biology. In: Compression and Complexity of Sequences 1997. Proceedings. IEEE, pp. 357–367.
16. De Bonis A, Vaccaro U (1998) Improved algorithms for group testing with inhibitors. Information Processing Letters 67: 57–64.
17. Xie M, Tatsuoka K, Sacks J, Young SS (2001) Group testing with blockers and synergism. Journal of the American Statistical Association 96: 92–102.
18. Barillot E, Lacroix B, Cohen D (1991) Theoretical analysis of library screening using an n- dimensional pooling strategy. Nucleic Acids Research : 6241–6247.
19. Rais Y, Zviran A, Geula S, Gafni O, Chomsky E, et al. (2013) Deterministic direct reprogramming of somatic cells to pluripotency. Nature : In press.
20. Bar-Lev S, Stadje W, Van der Duyn Schouten F (2006) Group testing procedures with incomplete identification and unreliable testing results. Applied Stochastic Models and Data Analysis 22: 281–296.
21. D'yachkov A, Hwang F, Macula A, Vilenkin P, Weng C (2005) A construction of pooling designs with some happy surprises. Journal of Computational Biology 12: 1129–1136.
22. Macula A (1999) Probabilistic nonadaptive group testing in the presence of errors and DNA library screening. Annals of Combinatorics 3: 61–69.
23. Thai M, MacCallum D, Deng P, Wu W (2007) Decoding algorithms in pooling designs with in- hibitors and error-tolerance. International Journal of Bioinformatics Research and Applications 3: 145–152.
24. Thierry-Mieg N (2006) A new pooling strategy for high-throughput screening: the Shifted Transver-sal Design. BMC bioinformatics 7: 28.
25. Dorfman R (1943) The detection of defective members of large populations. The Annals of Math- ematical Statistics 14: 436–440.
26. Balding D, Bruno W, Knill E, Torney D (1996) A comparative survey of non-adaptive pooling designs. In: Genetic mapping and DNA sequencing, New York: Springer. pp. 133–154.
27. Du DZ, Hwang F (2000) Combinatorial Group Testing and Its Applications. Singapore: World Scientific.
28. Kainkaryam R, Woolf P (2008) poolhits: A shifted transversal design based pooling strategy for high-throughput drug screening. BMC bioinformatics 9: 256.
29. Xin X, Rual J, Hirozane-Kishikawa T, Hill D, Vidal M, et al. (2009) Shifted Transversal Design smart-pooling for high coverage interactome mapping. Genome Research 19: 1262.
30. Langfeldt S, Hughes-Oliver J, Ghosh S, Young S (1997) Optimal group testing in the presence of blockers. Institute of Statistics Mimeograph Series 2297.
31. De Bonis A (2008) New combinatorial structures with applications to efficient group testing with inhibitors. Journal of Combinatorial Optimization 15: 77–94.
32. Chang H, Chen HB, Fu HL (2010) Identification and classification problems on pooling designs for inhibitor models. Journal of Computational Biology 17: 927–941.
33. Chang FH, Chang H, Hwang FK (2011) Pooling designs for clone library screening in the inhibitor complex model. Journal of combinatorial optimization 22: 145–152.
34. Rual JF, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, et al. (2005) Towards a proteome- scale map of the human protein–protein interaction network. Nature 437: 1173–1178.
35. Macula A, Rykov V, Yekhanin S (2004) Trivial two-stage group testing for complexes using almost disjunct matrices. Discrete Applied Mathematics 137: 97–107.
36. Damaschke P (2006) Threshold group testing. In: General theory of information transfer and combinatorics, Springer. pp. 707–718.
37. Chen HB, Fu HL (2009) Nonadaptive algorithms for threshold group testing. Discrete Applied Mathematics 157: 1581–1585.
38. Chen HB, De Bonis A (2011) An almost optimal algorithm for generalized threshold group testing with inhibitors. Journal of Computational Biology 18: 851–864.
39. Akutsu T, Bao F (1996) Approximating minimum keys and optimal substructure screens. In: Computing and Combinatorics, Second Annual International Conference (COCOON '96). volume 1090 of Lecture Notes in Computer Science, pp. 290–299.
40. Blum A, Langley P (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence 97: 245–272.
41. Akutsu T, Kuhara S, Maruyama O, Miyano S (1998) Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In: Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms. pp. 695–702.
42. Akutsu T, Miyano S, Kuhara S (1999) Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In: Proceedings of the Pacific Symposium on Biocomputing. pp. 17–28.
43. Arpe J, Reischuk R (2003) Robust inference of relevant attributes. In: Proceedings of the 14th International Conference on Algorithmic Learning Theory (Lecture Notes in Computer Science, Vol. 2842). pp. 99–113.
44. Bshouty NH, Mossel E, O'Donnell R, Servedio RA (2005) Learning DNF from random walks. Journal of Computer and System Sciences 71: 250–265.
45. Fukagawa D, Akutsu T (2005) Performance analysis of a greedy algorithm for inferring boolean functions. Information Processing Letters 93: 7–12.
46. Perkins TJ, Hallett MT, Glass L (2004) Inferring models of gene expression dynamics. Journal of Theoretical Biology 230: 289–299.
47. Perkins TJ, Hallett M (2010) A trade-off between sample complexity and computational complexity in learning boolean networks from time series data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7: 118–125.
48. Krupa B (2002) On the number of experiments required to find the causal structure of complex systems. Journal of Theoretical Biology 219: 257–267.
49. Bertsekas D (2005) Dynamic programming and optimal control, vol. I. Athena Scientific.
50. Deplancke B, Mukhopadhyay A, Ao W, Elewa A, Grove C, et al. (2006) A gene-centered C. elegans protein-DNA interaction network. Cell 125: 1193–1205.
51. Ngo V, Davis R, Lamy L, Yu X, Zhao H, et al. (2006) A loss-of-function RNA interference screen for molecular targets in cancer. Nature 441: 106–110.
52. Demant P, Hart A (1986) Recombinant congenic strains—a new tool for analyzing genetic traits determined by more than one gene. Immunogenetics 24: 416–422.