Appendix

A Appendix for Section 3: The lower bounds

A.1 Proof of Theorem 1

Proof. Ineq. (1) is because of the following counting argument: There are only 2^T combinations of test results. But, because of the community model I, there are $\binom{F}{k_f} \cdot \prod_{j=1}^{k_f} \binom{M_j}{k_m^j}$ possible sets of infected members that each must give a different set of results. Thus,

$$2^T \geq \binom{F}{k_f} \cdot \prod_{j=1}^{k_f} \binom{M_j}{k_m^j},$$

which reveals the result. The RHS of the latter inequality is because there are $\binom{F}{k_f}$ combinations of infected families, and for each infected family j, there are $\binom{M_j}{k_m^j}$ possible combinations of infected family members—hence for each combination of k_f infected families, there are $\prod_{j=1}^{k_f} \binom{M_j}{k_m^j}$ possible combinations of infected family members. The symmetric bound is obtained as a corollary by taking $M_j = M$ and $k_m^j = k_m$ for each infected family j. □

A.2 Proof of Theorem 2

Proof. Let $\mathbf{V}$ be the indicator random vector for the infection status of all families. By rephrasing [Li et al., 2014, Theorem 1], any probabilistic group testing algorithm using T noiseless tests can achieve a zero-error reconstruction of $\mathbf{U}$ if:

$$T \geq H(\mathbf{U}) = H(\mathbf{V}) + H(\mathbf{U}|\mathbf{V}) - H(\mathbf{V}|\mathbf{U}). \quad \text{(A.1)}$$

The first term is: $H(\mathbf{V}) = \sum_{j=1}^{F} H(V_j) = F h_2(q)$.

The second term is calculated as:

$$\begin{aligned} H(\mathbf{U}|\mathbf{V}) &= \sum_{v=1}^{n} H(U_v|V_{E_v}) \\ &= \sum_{v=1}^{n} \sum_{x\in\{0,1\}} \mathbb{P}(V_{E_v} = x) H(U_v|V_{E_v} = x) \\ &= \sum_{v=1}^{n} (qH(U_v|V_{E_v} = 1) + (1-q)H(U_v|V_{E_v} = 0)) \\ &= \sum_{v=1}^{n} q h_2(p_{E_v}) = q \sum_{j=1}^{F} M_j h_2(p_j), \end{aligned}$$

where E_v is the family containing vertex v.

Finally, we compute the third term as:

$$\begin{aligned} H(\mathbf{V}|\mathbf{U}) &= \sum_{j=1}^{F} H(V_j|\mathbf{U}) = \sum_{j=1}^{F} H(V_j|\mathbf{U}_{S_j}) \\ &= \sum_{j=1}^{F} \mathbb{P}(\mathbf{U}_{S_j} = \mathbf{0}) h_2(\mathbb{P}(V_j = 0|\mathbf{U}_{S_j} = \mathbf{0})) \\ &= \sum_{j=1}^{F} (1 - q + q(1-p_j)^{|S_j|}) h_2\left(\frac{1-q}{1-q+q(1-p_j)^{|S_j|}}\right) \end{aligned}$$

where S_j is the set of members who belong to family j and $|S_j| = M_j$. Combining all the 3 terms concludes the proof. □

B Appendix for Section 4.1: The noiseless adaptive case

B.1 Rationale for Alg. 1

Group testing already has a rich body of literature with near-optimal test designs in the case of independent infections, we do not try to improve upon them. Instead, we adapt these ideas to incorporate the correlations arisen from the community structure. All test designs described in this section are conceptually divided into two parts. This split is guided by the community structure and attempts to identify the different infection regimes inside the community, so that the best testing method (individual or classic group testing) is used. We show that such a two-part design is enough to significantly reduce the cost of group testing and also achieve the lower bound in some cases.

Two-part design: Two parts of Algorithm 1 serve complimentary goals:

The goal of Part 1 is to detect the infection *regime* inside each family j: i.e., to accurately estimate which of the F families have a high infection rate ("heavily" infected) and which are have a low or zero infection rate ("lightly" infected). Our interest in detecting the infection regime is motivated by prior work [Riccio and Colbourn, 2000,Hu et al., 1981], which has shown that group testing offers benefits over individual testing, only if the infection rate is low ($p_j \leq 0.38$). This

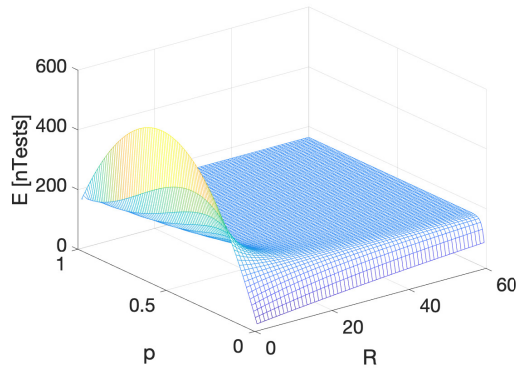


Figure 6: Expected number of tests from (7) as a function of size of representative set and probability of infection inside a family.

allows us to define the two regimes as follows: In the combinatorial model I (resp. probabilistic model II), a is considered heavily infected when $k_m^j/M_j \geq 0.38$ (resp. $p_j \geq 0.38$); conversely, it is considered lightly infected family when $k_m^j/M_j < 0.38$ (resp. $p_j < 0.38$).

For each family j, we regard $\hat{U}_{x(r_j)}$ as an estimate of the family's infection regime. If $\hat{U}_{x(r_j)}$ is positive, we consider the family to be highly infected and therefore perform individual testing for all of its members. Otherwise, if $\hat{U}_{x(r_j)}$ is negative, we consider the family to be lightly infected and group test its members with all other lightly infected families. The challenge is therefore to produce accurate enough regime estimates, such that the overall number of tests that are needed from Alg. 1 to achieve exact infection-status reconstruction for all members $i = 1, \ldots, n$ is minimal. We discuss this challenge further below.

Given all estimates $\hat{U}_{x(r_j)}$ from Part 1, the goal of the Part 2 is then to identify all infected members, by using the appropriate testing method (group or individual testing) according to the infection regime of each family (light or heavy). In this way, at the end of Part 2, the algorithm returns an estimate $\hat{U}_i$ of the true infection status U_i of each individual member i.

Selection of family representatives: Function *SelectRepresentatives*() at line 2 refers to *any* sampling function on a set of family members, as long as it returns a fixed number of members from family j. That is, one may use their own sampling function, as long as the accuracy of Part 1 is well defined. In this paper, we consider only random-sampling functions without replacement (i.e. $|r_j|$ members are randomly chosen from the family members and each subset of that size has the same probability of being selected as the representative subset). But perhaps, more elaborate sampling functions may be considered in other contexts. For example, if the internal structure of family j can be represented through a contact graph, in which only specific family nodes have external contacts with other families, it may make sense to include (some of) these nodes into the representative group with certainty.

When only one mixed sample per family is used to identify the heavily/lightly infected families, the cardinality of the representative subset $|r_j|$ is essential, but the optimal choice of it is not trivial. $|r_j|$ affects the accuracy of regime estimate—hence the performance of our algorithm in terms of the expected number of tests that it uses. Unfortunately, choosing the number of representatives optimally is not easy even in the symmetric case that is examined in Section 4.1. Ideally, in the symmetric case, we would like to choose $|r_j| = R$ such that the bounds in Lemmas 1 and 2 are minimized. However, this requires solving equations of the form $ye^y = x$, which is generally possible through Lambert functions for $x \geq -\frac{1}{e}$, but the latter does not hold in our case. Fig. 6 demonstrates that there exists no unique R that is optimal for any infection probability p in $(0, 1)$ through an example of $F = 50$ families with $M = 60$ members each. The figure plots the bound of Lemma 2 as a function of p and R. As we can see, there is no single minimizer $R^\star$: if $p < 0.15$, then R must be picked equal to 0 (which yields traditional group testing); otherwise, if $p > 0.15$, then R must be selected equal to M.

Therefore, in order to optimally choose R, a rough estimate about p has to be known a priori. If the latter is not possible, then one may use a few more tests at the first stage of our algorithm to better detect whether a family is heavily infected. We provide such an optimization in the next section.

Function *AdaptiveTest*()**:** In both parts of our algorithm, we make use of a classic adaptive-group-testing algorithm, which we call *AdaptiveTest*(). This may be regarded as an abstraction for any existing (or future) adaptive algorithm in the group-testing literature. In our analysis, however, we mostly focus on the classic binary splitting algorithm because of its good performance in realistic cases, where the numbers of infected families and/or members (k_f, k_m^j) are unknown [Sobel and Groll, 1959].

In this section, we consider only adaptive algorithms that offer noiseless (zero-error) reconstruction. Note, however, the fact that *AdaptiveTest*() offers exact reconstruction is not enough to guarantee an accurate detection of any family's infection regime in Part 1. For example, consider the following case, where the true infection rate within a family j is not very low (say $p_j = 0.6$), yet none of the family representative in set r_j happened to be infected. Intuitively, the error probability of detection in Part 1 should depend

on the number of selected representatives $|r_j|$ from each family j and the infection rate among its members p_j. In our analysis, we examine different scenaria w.r.t. these parameters and discuss which parametrization (i.e. value of $|r_j|$) optimizes the expected number of the tests required by our algorithm.

B.2 Modified/Optimized versions of Alg. 1

• One modification of our algorithm is the following: In Part 1, instead of selecting only one representative group for each family, we select m_s representative subgroups, each of size s, and we treat each of these subgroups as a single "(super)-member". That is, we identify whether each subgroup is positive (has at least one positive member) or not, and based on this information, using for example majority vote, we can classify the family as heavily or lightly infected; essentially we can solve an estimation problem as in [Aldridge et al., 2019] (see Chapter 5.3), [Walter et al., 1980, Sobel and Elashoff, 1975]. In this regard, Alg. 1 is just a special case of this approach, with $m_s = 1$ and $s = |r_j|$.

Intuitively, we expect that such a modification would increase the estimation accuracy of $\hat{p_j}$ and reduce the error of the related hypothesis test, at the cost of few more tests. As a result, it could need fewer tests on expectation than Alg. 1, hence perform better in some cases. However, the potential improvement would depend on parameters such as the family size - for instance for small size families it is not expected to be large. To keep things simple, we prefer not to analyze this algorithm in this paper and defer it to future work.

• Another modification could be the following: instead of leveraging the community structure to perform individual tests where needed, we could use it to improve traditional binary splitting algorithm by running it on multiple testing groups that are related to the community structure. For example, consider a symmetric case where: we split all $n = FM$ members into M groups of F people (one from each family), then run binary splitting to each of these groups.

This modification is also related to Hwang's binary splitting algorithm, but achieves only logarithmic benefits compared to binary splitting, as opposed to our algorithm that may perform much better in real cases (see Section 4.1). In fact, the expected number of tests needed by this modified algorithm would be at most $k \log_2 ({}^{n}/_{M}) + O(k)$: each group g has k_g infected member and binary splitting needs $k_g \log_2 ({}^{n}/_{M}) + O(k_g)$ tests to identify all of them. By adding together the number of tests for each group g, we deduce the result.

• A last modification occurred to us after a related comment of one of our reviewers, who we thank. As discussed in Section 4.1, when a sparse regime holds for families (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$) and a heavily linear regime holds within each family (i.e. $k_m \approx M$), the benefits of Lemma 1 with regard to Hwang's binary splitting (HBSA) cannot be more than $1/\log(n/k)$. This is because, in Eq. (4), we get the additive term $k_f M > FM = k$, which comes from the second stage of Algorithm 1.

Nevertheless, if $k_m > M - k_m$ (i.e., the infection rate inside each family is more than 0.5), then at the second stage of our algorithm it makes more sense to look for not-infected members and stop testing once we find them. In that case, we need at least $k_f * (M - k_m)$ tests, which can be less than k, and therefore could lead to more benefits on average.

For example, consider the case where $k_m = M - 1$. Then the expected number of individual tests needed to find the 1 not-infected member inside each infected family can be computed as follows: Without loss of generality, suppose that we test the members at some fixed ordering without replacement and the not-infected member has a uniformly random position in that ordering. Then, the probability of the not-infected item being at a given position i in the ordering is equal to $1/M$ and we need i tests to find it. As a result, the expected number of tests is $\sum_{i=1}^{M} i * 1/M = (M + 1)/2$. From linearity of expectation, the expected number of tests for all infected families at the second stage of our algorithm (if we further assume that all infected families are identified without error at the first stage—i.e., $\phi_c = 1$) will be: $k_f * (M + 1)/2 < k_f * (M - 1) = k_f * k_m = k$. Hence, is this particular regime, the modification of our algorithm can achieve benefits more than $1/\log(n/k)$.

In the more general case, where $M - k_m > 1$, the relevant probabilities for the computation of the expected number of tests can be obtained from the negative hypergeometric distribution (since sampling is without replacement).

In the extreme case, where for each infected family k_m is known and equal to M, all we need to do is to identify the infected families and label all their members as infected. In that case the benefit would be k_f/k. Note, that to achieve these higher benefits described above, the knowledge of the number of infected members per family is required, but this is also the case for HBSA.

[3] The symmetric example is only used here only to better illustrate the advantages of the modification proposed. The idea is similar for the asymmetric case.

B.3 Proof of Lemma 1

Proof. Let ϕ_c be the expected fraction of infected families whose mixed sample is positive. Since *SelectRepresentatives*() is uniform random sampling without replacement, we can compute ϕ_c when $1 \leq R \leq M - k_m$ using the hypergeometric distribution $Hyper(M, k_m, R)$, as follows: the probability of a random mixed sample $x(r_j)$ being negative (i.e. all members of r_j are negative) is given by the PMF of $Hyper(M, k_m, R)$ evaluated at 0, and it is therefore equal to $\binom{M-k_m}{R}/\binom{M}{R}$, which yields $\phi_c = 1 - \binom{M-k_m}{R}/\binom{M}{R}$. We also define the following for completeness: $\phi_c = 0$ when $R = 0$ and $\phi_c = 1$ when $M - k_m < R \leq M$.

Fixing the number of positive mixed samples in Part 1 of Alg. 1 to its expected value: $k_f \cdot \phi_c$, we now compute the maximum number of tests needed by the algorithm to succeed.

Alg. 1 performs testing at lines 4, 8, 13.

• At line 4, it identifies the positive mixed samples to mark the corresponding families as heavily infected and all others as lightly infected. If HGBSA is used for *AdaptiveTest*(), then Alg. 1 is expected to succeed at this step using $k_f\phi_c \log_2 \frac{F}{k_f\phi_c} + k_f\phi_c$ tests. Similarly, if BSA is used for *AdaptiveTest*(), then then Alg. 1 is guaranteed to succeed at this step using at most $k_f\phi_c \log_2 F + k_f\phi_c$ [Aldridge et al., 2019, Baldassini et al., 2013].

• At line 8, the expected number of individual tests is equal to: $Mk_f\phi_c$. This is the same irrespectively from whether *AdaptiveTest*() is binary splitting or Hwang's algorithm as it only depends on ϕ_c.

• At line 13, the expected number of items that are tested is: $n - k_f\phi_c M$, and the expected number of infected members is: $k - k_f\phi_c k_m = k(1-\phi_c)$. So, if HGBSA is used for *AdaptiveTest*(), then Alg. 1 is guaranteed to succeed at this step using $k(1-\phi_c)\log_2 \frac{(n-k_f\phi_c M)}{k(1-\phi_c)} + k(1-\phi_c)$ tests. Similarly, if BSA is used, then Alg. 1 is expected to succeed in at most: $k(1-\phi_c)\log_2(n - k_f\phi_c M) + k(1-\phi_c)$ tests [Aldridge et al., 2019, Baldassini et al., 2013].

We add together all the above terms that are related to HGBSA or BSA, and the result follows. □

B.4 Proof of Lemma 2

Proof. Let ϕ_p be the expected fraction of infected families whose mixed sample is positive. Then, because of the probabilistic setting, $\phi_p = 1 - (1-p)^R$.

Alg. 1 performs testing at lines 4, 8, 13.

• At line 4, the expected number of mixed samples that are positive is $Fq\phi_p$. So, if BSA is used in the place of *AdaptiveTest*(), then the maximum number of tests needed to identify all mixed samples is on expectation $Fq\phi_p \log_2 F + Fq\phi_p$ [Aldridge et al., 2019, Baldassini et al., 2013].

• At line 8, the expected number of individual tests is equal to: $Fq\phi_p M$.

• At line 13, the expected number of items that are tested is: $n - Fq\phi_p M$, and the expected number of infected members is equal to the expected number of all infected members minus the expected number of the ones that are identified though individual testing at line 8: i.e., $FqMp - Fq\phi_p Mp = FqMp(1-\phi_p) = nqp(1-\phi_p)$. So, if BSA is used in the place of *AdaptiveTest*(), it is expected to succeed using at most $nqp(1-\phi_p)\log_2(n - Fq\phi_p M) + nqp(1-\phi_p)$ tests [Aldridge et al., 2019, Baldassini et al., 2013].

We add together all the above terms and the result follows. □

C Appendix for Section 4.3: The Noiseless Non-adaptive case

C.1 Zero error requirements

For our design of $\mathbf{G}_2$, we have the following lemma and observation.

Lemma 7. *To achieve zero-error w.r.t.* $\mathbf{G}_2$*, we need* $T_2 \geq n$.

Proof. A trivial implementation for $\mathbf{G}_2$ is to use an identity matrix of size n; since each member is tested individually, we can identify all the infected members correctly. We next argue that $T_2 \geq n$ for the zero-error case. We prove this through contradiction. Assume that $T_2 < n$. Then, from the pigeonhole principle, there exists one member, say m_1 that does not participate in any test alone -it always participates together with one or more members from a set $\mathcal{S}_1$. Assume that all members in S_1 are infected, while m_1 belongs in an infected family but is not infected -our decoding will result in a FP. □

Observation: $\mathbf{G}_2$ leads to zero error if and only if it has the following property:
Zero Error Property: Any subset of $\{1, 2, \cdots, n\}$ of size $(F - k_f)M + k_f(M - k_m)$ equals the union of some testing rows of $\mathbf{G}_2$. Namely, the members of the not-infected families together with the not-infected members of the infected families, need to be the only participants in some rows of $\mathbf{G}_2$, for all possible not-infected families and not-infected members. This requirement can lead to an alternative proof of Lemma 7.

C.2 Rationale for the structure of $\mathbf{G}_2$

Our goal is to design a non-trivial matrix $\mathbf{G}_2$ that can identify almost all the infected members with high probability and a small number of tests. We next discuss two intuitive properties we would like our designs to have to minimize the error probability.

Desirable Property 1: Use identity matrices as building blocks.
Intuition: ideally, after removing the $(F - k_f)M$ columns corresponding to the members in non-infected families, we would like the remaining columns to form an identity matrix so that we can identify all the infected members correctly. To reduce the number of tests, there should be more than one members included in each test. Thus we use overlapping identity matrices, one corresponding to each family. We assume the index for the n members is family-by-family, i.e., the indexes for the members in the same family are consecutive. Then each family corresponds to an identity sub-matrix I_M in $\mathbf{G}_2$. Now the problem becomes how to arrange the identity sub-matrices.

Desirable Property 2: The identity matrices corresponding to different families either appear in the same set of M rows in $\mathbf{G}_2$ or they do not appear in any shared rows.
Intuition: otherwise, a family would share tests with more other families. Then the probability that this family shares tests with infected families becomes larger. This would increase the probability that two infected families share tests after removing all the non-infected family columns, which in turn would increase the FP probability.

C.3 Proof of Lemma 3

Proof. The probabilities can be explained as follows:

(i) For $\mathbb{P}^{I}_{\text{joint}}$ in (9), the numerator gives the number of possibilities that each block row contains at most one infected family, which is obtained by randomly choosing k_f block rows (the summation) and then from each chosen block row choosing one family to be infected (c_i possible choices for i-th block row). The denominator is the total number of infection possibilities, and then the fraction denotes the probability that each block row contains at most one infected family. Thus, $\mathbb{P}^{I}_{\text{joint}}$ is obtained as the probability that there is some block row that contains two or more infected families.

(ii) For $\mathbb{P}^{II}_{\text{joint}}$ in (10), $(1-q)^{c_i}$ is the probability that there is no infected family in the i-th block row, and $c_i q(1-q)^{c_i-1}$ is the probability that there is only one infected family in the i-th block row. The multiplication $\prod$ denotes the probability that any one block row contains at most one infected family. Thus, $\mathbb{P}^{II}_{\text{joint}}$ is obtained as the probability that there is some block row that contains two or more infected families.

□

C.4 Proof of Lemma 4

Proof. Consider $c_i > c_j + 1$, let $c_i' = c_i - 1$ and $c_j' = c_j + 1$. For the combinatorial model, we can verify the difference of the probability for c_i' and c_i by

$$\begin{aligned}\sum_{\substack{|\mathcal{B}|=k_f:\\ \mathcal{B}\subseteq\{1,2,\cdots,b\}}} \prod_{\ell\in\mathcal{B}} c_\ell' - \sum_{\substack{|\mathcal{B}|=k_f:\\ \mathcal{B}\subseteq\{1,2,\cdots,b\}}} \prod_{\ell\in\mathcal{B}} c_\ell &= (c_i' c_j' - c_i c_j)\cdot X \\ &= (c_i - c_j - 1)\cdot X \\ &> 0,\end{aligned}$$

where X is a positive value independent of c_i and c_j. This implies that the minimum of the probability $\mathbb{P}^{I}_{\text{joint}}$ in (10) achieves its minimum roughly at the symmetric case where all c_i's are equal, i.e., $c_i = c$ for all $i \in \{1, 2, \cdots, b\}$.

Similarly, for the probabilistic model, consider the probability in (10), we can calculate that

$$\begin{aligned}&\prod_{\ell=1}^{b}\left[(1-q)^{c_\ell'} + c_\ell' q(1-q)^{c_\ell'-1}\right] \\ &\quad - \prod_{\ell=1}^{b}\left[(1-q)^{c_\ell} + c_\ell q(1-q)^{c_\ell-1}\right] \qquad \text{(C.1)}\\ &= [(c_i - c_j) - (1-q)^2]q^2(1-q)^{c_i+c_j-2}\cdot Y \\ &> 0, \qquad \text{(C.2)}\end{aligned}$$

where $Y = \prod_{\ell\neq i,j}\left[(1-q)^{c_\ell} + c_\ell q(1-q)^{c_\ell-1}\right] > 0$ is independent of c_i and c_j. This implies that the minimum of the probability in (10) achieves its minimum roughly at the symmetric case where all c_i's are equal, i.e., $c_i = c$ for all $i \in \{1, 2, \cdots, b\}$. □

C.5 Proof of Lemma 5

The lemma is obtained under the assumption that the number of families F is a multiple of b and c. If F cannot be factorized, the error probabilities in Lemma 5 can be viewed as an upper bound for the corresponding error probabilities. This can be seen by simply adding F' auxiliary families so that $F + F' = bc$.

Proof. In the symmetric case, i.e., $c_i = c$ for all $i \in \{1, 2, \cdots, b\}$, the probabilities in (9) and (10) become

$$\mathbb{P}^{I}_{\text{joint}} = 1 - \frac{\binom{b}{k_f} c^{k_f}}{\binom{F}{k_f}}, \tag{C.3}$$

$$\mathbb{P}^{II}_{\text{joint}} = 1 - \left((1-q)^{c-1}(1-q+cq)\right)^{b}. \tag{C.4}$$

For the symmetric combinatorial model, the number of infected members in an infected family $k^j_m = k_m$ for all infected families j. If two families appear in the same set of M tests, the probability that all infected members in one family share the same k_m tests as the other family is simply

$$\mathbb{P}(\text{no FP}|\text{joint}) = \frac{1}{\binom{M}{k_m}}. \tag{C.5}$$

Thus the probability that FPs happen is

$$Pe = \mathbb{P}(\text{FP}|\text{joint}) \cdot \mathbb{P}^{I}_{\text{joint}} = \left[1 - \frac{1}{\binom{M}{k_m}}\right]\left[1 - \frac{\binom{b}{k_f} c^{k_f}}{\binom{F}{k_f}}\right]. \tag{C.6}$$

For the symmetric probabilistic model, the infection probability in an infected family $p_j = p$ for all infected families j. If two families appear in the same set of M tests, then there is no false positives only when the two families have the same number of infected members and the infected (non-infected) members in one family must appear in the same set of tests as infected (non-infected) members of the other family. The probability that two families both have i infected members is $\left[p^i(1-p)^{M-i}\right]^2$, and the probability that all infected members in one family share tests with only infected members in the other family is simply $\frac{1}{\binom{M}{i}}$. Thus, the probability that there is no false positives is given as follows,

$$\mathbb{P}(\text{no FP}|\text{joint}) = \sum_{i=1}^{M} \left[p^i(1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}. \tag{C.7}$$

Thus the probability that a false positive happens can be obtained as

$$\begin{aligned} Pe &= \mathbb{P}(\text{FP}|\text{joint}) \cdot \mathbb{P}^{II}_{\text{joint}} \\ &= \left[\sum_{i=1}^{M} \left[p^i(1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}\right] \\ &\quad \cdot \left[1 - \left((1-q)^{c-1}(1-q+cq)\right)^{b}\right]. \end{aligned} \tag{C.8}$$

Replacing b by T_2/M and c by FM/T_2 completes the result. □

C.6 Proof of Lemma 6 and Discussions

Proof. For the combinatorial model (I), it is hard to explicitly calculate the expected error rate. The upper bound in (12) is obtained by assuming that if there exist errors (FPs), then all non-infected members in infected families are misidentified as infected in the decoding of $\mathbf{G}_2$. (Note that all non-infected members in non-infected families are correctly identified by decoding of $\mathbf{G}_1$.)

For the probabilistic model (II), the upper bound for the expected error rate in (13) is obtained by

$$\begin{aligned} R_{II}(\text{error}) &= \frac{1}{n} \cdot b \cdot \left[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \right. \\ &\quad \left. \cdot \left(\sum_{i=1}^{j} \binom{j}{i} p^i (1-p)^{j-i} (j-i)\right) \cdot M\right] \qquad \text{(C.9)} \\ &= \frac{bM}{n} \cdot \left[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \right. \\ &\quad \left. \cdot \left(j(1-p) - j(1-p)^j\right)\right] \qquad \text{(C.10)} \\ &< \frac{(1-p)\,T_2}{n} \cdot \left[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \cdot j\right] \\ &= \frac{(1-p)\,T_2}{n} \cdot \left[cq - cq(1-q)^{c-1}\right], \\ &= (1-p)q\left[1 - (1-q)^{c-1}\right], \qquad \text{(C.11)} \end{aligned}$$

where the expression in the bracket in (C.9) for each j denotes the expected number of FPs in one block row if there are j families infected in this block row, (C.10) is obtained from the expected value of binomial distribution, and (C.11) follows by substituting $c = \frac{n}{T_2}$. □

We here make the following observation about the system FP probability $\mathbb{P}$(any-FP): As we explore further in Section 6 non-adaptive group testing requires more tests than adaptive. Assume that $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$ and choose $R = M - 1$ in Algorithm 1. Adaptive testing allows to achieve zero error with $k_f \log_2 F + k_f M$ tests; if we use the same (order) number of tests with a non-adaptive strategy, i.e., $T_1 = k_f \log_2 \frac{F}{k_f}$ and $T_2 = k_f (\log_2 k_f + M)$, we get $\mathbb{P}$(any-FP) in Lemma 5 approximately equal to $\left(1 - \frac{1}{M}\right)\left[1 - \frac{\binom{T_2/M}{k_f}}{\left(\frac{T_2/M}{k_f}\right)^{k_f}} \frac{(F/k_f)^{k_f}}{\binom{F}{k_f}}\right]$ which is bounded away from 0. The latter can be seen as follows: i) $T_2/M \approx k_f \ll F$; ii) $\frac{\binom{n}{k}}{\left(\frac{n}{k}\right)^k} \Big/ \frac{\binom{n+m}{k}}{\left(\frac{n+m}{k}\right)^k} = \left(\frac{n}{n+m}\right)^k \cdot \prod_{i=1}^{m} \frac{n+i-k}{n+i}$ is decreasing with m and can be very small when $m \gg n$.

Fig. 7 depicts $\mathbb{P}$(any-FP) and R(error) for parameters $F = 64$, $k_f = 6$, $k_m = 4$, $M = 5$, $q = 1/8$, and $p = 0.8$.

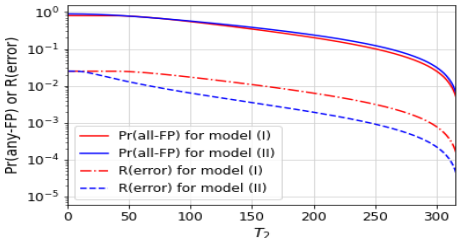


Figure 7: System FP probability and FP error rate.

D Appendix for Section 5: Loopy Belief Propagation algorithm

We here describe our loopy belief propagation algorithm (LBP) and update rules for our probabilistic model (II). We use the factor graph framework of [Kschischang et al., 2001] and derive closed-form expressions for the sum-product update rules (see equations (5) and (6) in [Kschischang et al., 2001]).

The LBP algorithm on a factor graph iteratively exchanges messages across the variable and factor nodes. The messages to and from a variable node V_j or U_i are *beliefs* about the variable or distributions (a local estimate of $\mathbb{P}(V_j|\text{observations})$ or $\mathbb{P}(U_i|\text{observations})$). Since all the random variables are binary, in our case each message would be a 2-dimensional vector $[a, b]$ where $a, b \geq 0$. Suppose the result of each test is y_τ, i.e., $Y_\tau = y_\tau$ and we wish to compute the marginals $\mathbb{P}(V_j = v|\, Y_1 = y_1, Y_2 = y_2, ..., Y_T = y_T)$ and $\mathbb{P}(U_i = u|\, Y_1 = y_1, Y_2 = y_2, ..., Y_T = y_T)$ for $v, u \in \{0, 1\}$. The LBP algorithm proceeds as follows:

1. *Initialization:* The variable nodes V_j and U_i transmit the message $[0.5, 0.5]$ on each of their incident edges. Each variable node Y_τ transmits the message $[1 - y_\tau, y_\tau]$, where y_τ is the observed test result, on its incident edge.

2. *Factor node messages:* Each factor node receives the messages from the neighboring variable nodes and computes a new set of messages to send on each incident edge. The rules on how to compute these messages are described next.

3. *Iteration and completion.* The algorithm alternates between steps 2 and 3 above a fixed number of times (in practice 10 or 20 times works well) and computes an estimate of the posterior marginals as follows – for each variable node V_j and U_i, we take the coordinatewise product of the incoming factor messages and normalize to obtain an estimate of $\mathbb{P}(V_j = v|y_1...y_T)$ and $\mathbb{P}(U_i = u|y_1...y_T)$ for $v, u \in \{0, 1\}$.

Next we describe the simplified variable and factor node message update rules. We use equations (5) and (6) of [Kschischang et al., 2001] to compute the messages.

Leaf node messages: At every iteration, the variable node Y_τ continually transmits the message $[0, 1]$ if $Y_\tau = 1$ and $[1, 0]$ if $Y_\tau = 0$ on its incident edge. The factor node $\mathbb{P}(V_j)$ continually transmits $[1 - q, q]$ on its incident edge; see Fig. 8 (a) and (b).

Variable node messages: The other variable nodes Vj and U_i use the following rule to transmit messages along the incident edges: for incident each edge e, a variable node takes the elementwise product of the messages from every other incident edge e' and transmits this along e; see Fig. 8 (c).

Factor node messages: For the factor node messages, we calculate closed form expressions for the sum-product update rule (equation (6) in [Kschischang et al., 2001]). The simplified expressions are summarized in Fig. 8 (d) and (e). Next we briefly describe these calculations.

Firstly, we note that each message represents a probability distribution. One could, without loss of generality, normalize each message before transmission. Therefore, we assume that each message $\mu = [a, b]$ is such that $a + b = 1$. Now, the the leaf nodes labeled $\mathbb{P}(V_j)$ perennially transmit the prior distribution corresponding to V_j.

Next, consider the factor node $\mathbb{P}(U_i|V_j)$ as shown in Fig. 8 (d). The message sent to U_i is calculated as

$$\begin{aligned}\mu_u &= \sum_{v\in\{0,1\}} \mathbb{P}(U_i = u|V_j = v)w_v \\ &= w_0(1-u) + w_1 p_j^u (1-p_j)^{1-u}.\end{aligned}$$

Similarly, the message sent to V_i is

$$\begin{aligned}\nu_v &= \sum_{u\in\{0,1\}} \mathbb{P}(U_i = u|V_j = v)s_u \\ &= s_0(v(1-p_j) + 1 - v) + s_1 v p_j.\end{aligned}$$

Finally for the factor nodes $\mathbb{P}(Y_\tau|U_{\delta_\tau})$ as shown in Fig. 8 (e), note that the messages to Y_τ play no role since they are never used to recompute the variable

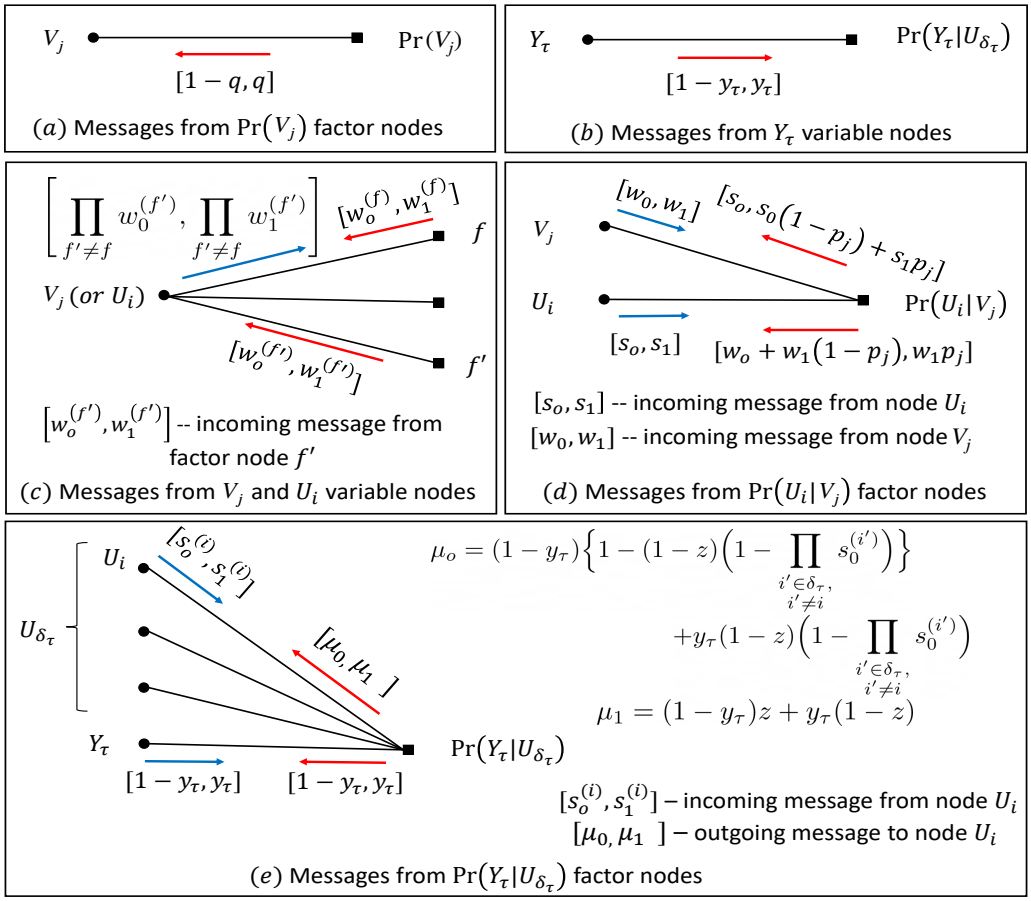


Figure 8: The update rules for the factor and variable node messages.

messages. The messages to U_i nodes are expressed as

$$\mu_u = \sum_{\substack{y\in\{0,1\},\\ \{u_{i'}\in\{0,1\}: i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = y|U_{\delta_\tau} = u_{\delta_\tau}) (1-y_\tau)^{1-y} y_\tau^{y} \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big)$$
$$= (1-y_\tau) \sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big)$$
$$+ y_\tau \sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 1|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big).$$

From our Z-channel model, recall that $\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) = 1$ if $u_i = 0\ \forall\ i \in \delta_\tau$ and $\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) = z$ otherwise. Thus we split the summation terms into 2 cases – one where $u_{i'} = 0$ for all i' and the other its complement. Also combining this with the assumption that the messages are normalized, i.e., $s_0^{(i)} + s_1^{(i)} = 1$, we get

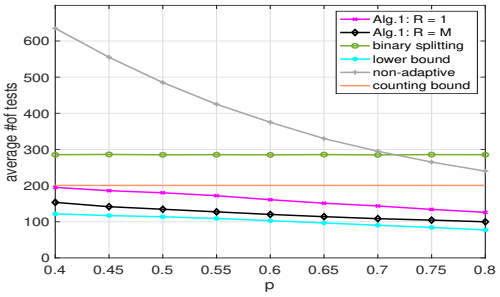
$$\sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big)$$
$$= \mathbb{1}_{u=1} z + \mathbb{1}_{u=0}\Big\{1-(1-z)(1-\prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')})\Big\},$$

and

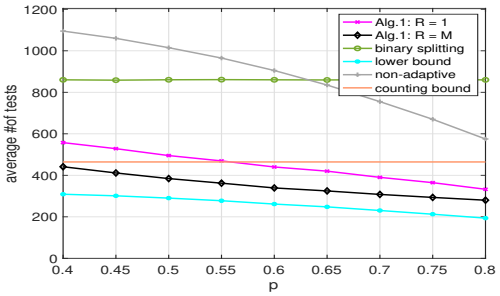
$$\sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 1|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big)$$
$$= \mathbb{1}_{u=1}(1-z) + \mathbb{1}_{u=0}\Big((1-z)(1-\prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')})\Big).$$

Substituting $u = 0$, and $u = 1$ we obtain the messages

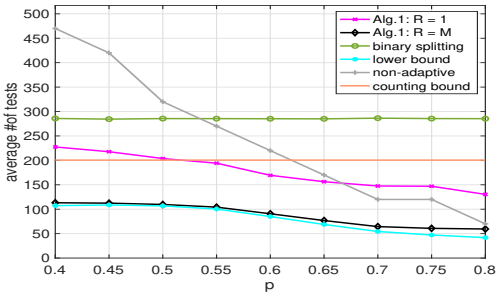
$$\mu_0 = (1-y_\tau)\Big\{1-(1-z)(1-\prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')})\Big\}$$
$$+ y_\tau(1-z)(1-\prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')}),$$



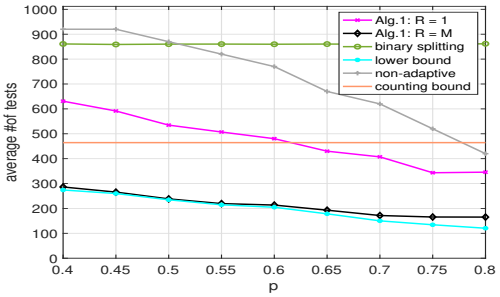
(a) Community 1—sparse regime.



(b) Community 1—linear regime.



(c) Community 2—sparse regime.



(d) Community 2—linear regime.

Figure 9: Experiment (i):
Noiseless case—Average number of tests.

and

$$\mu_1 = (1 - y_\tau)z + y_\tau(1 - z).$$

For our probabilistic model, the complexity of computing the factor node messages increases only linearly with the factor node degree.

E Appendix for Section 6: Other Results

We next provide additional experimental results to the ones provided in Section 6.

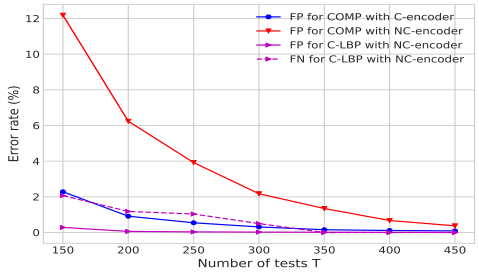
(i) *Noiseless testing – Average number of tests:* In Figure 9, we reproduce additional numerics akin to the ones in Section 6 for number of tests in the noiseless-testing case. As earlier, we measure the average number of tests needed by 3 algorithms that achieve zero-error reconstruction (Alg. 1 with $R = 1$, Alg. 1 with $R = M$, and classic BSA), and a version of our non-adaptive algorithm (Section 4.3) that uses $T_1 = F$ tests for submatrix $\mathbf{G}_1$ and has an overall FP rate around 0.5%. Alg. 1 assumes no prior knowledge of the number of infected families/classes or members/students, hence uses BSA as group-testing algorithm for the *AdaptiveTest*() function.

Fig. 9 depicts our results: We observe that both versions of Alg. 1 (black and magenta lines) need significantly fewer tests compared to classic BSA (green line), while staying below the counting bound. This indicates the potential benefits from the community structure, even when the number of infected members is unknown. More interestingly, when $R = M$, Alg. 1 performs close to the lower bound in most realistic scenarios $p \in [0.5, 0.8]$ (as also shown in Section 4.1). The grey line shows number of tests needed by our nonadaptive algorithm; we observe that even that algorithm needs fewer tests than BSA when p gets larger than 0.5, of course at the cost of a (FP) error rate of 0.5%.

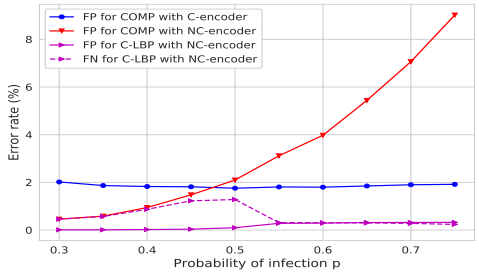
(ii) *Noiseless testing – Average error rate:* In Fig. 10, we reproduce additional numerics akin to the ones in Section 6 for average error rates in the noiseless-testing case. As earlier, we quantify the additional cost in terms of error rate, when one goes from a two-stage adaptive algorithm that achieves zero-error identification to much faster single-stage nonadaptive algorithms.

Fig. 10a is a reproduction of Fig. 3 for $p = 0.8$, and as can be seen its behavior is very similar to Fig. 3.

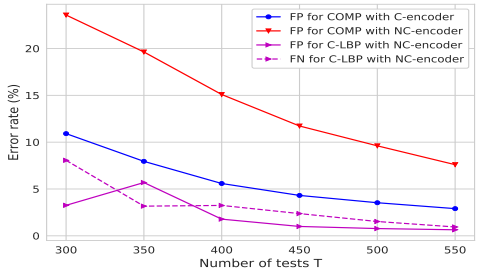
Fig. 10b depicts the FP and FN error rates (averaged over 500 runs) as a function of $p \in [0.3, 0.8]$ for Community 1 for the linear regime. We observe that any community-aware nonadaptive algorithm performs better than traditional nonadaptive group testing (red line) when $p > 0.5$ – the absolute performance gap ranges from 0.2% (when $p = 0.5$) to 8.5% (when $p = 0.8$). "COMP with C-encoder" has a stable FP rate across for all p values that was close to 2%, and a zero FN rate by construction. Unlike the



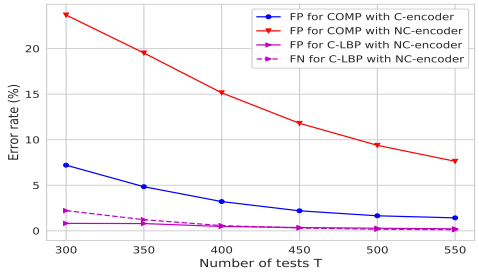
(a) Noiseless case: Average error rate $p = 0.8$ for sparse regime.



(b) Noiseless case: Average error rate with few tests for linear regime.



(c) Noiseless case: Average error rate $p = 0.6$ for linear regime.



(d) Noiseless case: Average error rate $p = 0.8$ for linear regime.

Figure 10: Experiment (ii): Noiseless case—Average error rate.

sparse regime, the LBP consistently produces better error rates compared to the COMP decoder. However, for low values of p, LBP produces more FN errors. For $p > 0.6$, both the FN and FP error rates are close to 0 for LBP.

Fig. 10c and Fig. 10d examine the effect of the number of tests in the linear regime. For $p = 0.6$, "C-LBP with NC-encoder" performs better than "COMP with C-encoder" for $T > 450$ until which both have high error rates. On the other hand, for $p = 0.8$, "C-LBP with NC-encoder" performs better than "COMP with C-encoder" for all values of T. More importantly, "COMP with C-encoder" seems to saturate to a non-zero FP error rate, while "C-LBP with NC-encoder" is able to attain close to zero error FP and FN rates. These results contrast with the results for the sparse regime.

(iii) *Noisy testing:*

In Figure 11, we reproduce additional numerics akin to the ones in Section 6 for average error rates in the noisy-testing case. As earlier, we assuming the Z-channel noise of Section 2.3 with parameter $z = 0.15$, and we evaluate the performance of our community-based LBP decoder of Section 5 against a LBP that does not account for community—namely its factor graph has no V_j nodes.

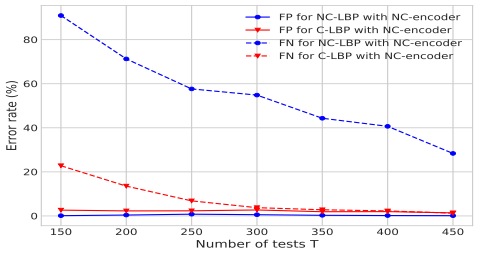
Fig. 11a is a reproduction of Fig. 4 for $p = 0.6$, and as can be seen its behavior is very similar to it.

Fig. 11b and Fig. 11c depict our results for Community 1 and for $p = 0.6$ and $p = 0.8$ in the linear infection regime. We observe that the knowledge of the community structure reduces the FN rates achieved by LBP. The FP error rates are always close to 0 while the, FN error rates drop significantly (up to 60% when tests are few), which is important in our context since FN errors lead to further infections.

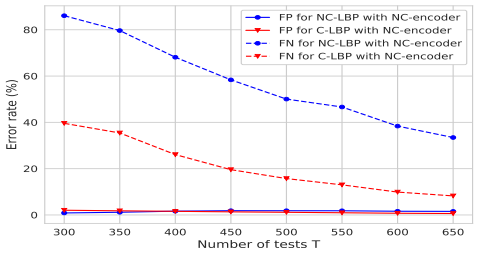
(iv) *Asymmetric case—Linear regime:*

Here we offer the results about an asymmetric setup that parallels the one of Section 6. Infections follow again the probabilistic model (II), and the size of each family is randomly selected from the interval $[5, 50]$ and the infection rate of each infected family is randomly selected from the range $[0.4, 0.8]$. But, this time $q = 5\%$.

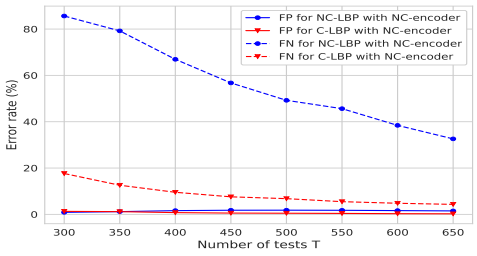
Figure 5 depicts our results. BSA needs on average $6.19\times$ (that can reach up to $13.87\times$) more tests compared to the probabilistic bound, while the two versions of Algorithm 1 with $R = 1$ and $R = M$ need only $2.74\times$ and $1.19\times$ (that can reach up to $9.7\times$ and $2.03\times$) more tests, respectively. Also, similarly to the sparse regime, there is a significantly smaller range between the 25-th and 75-th percentiles of the box-plots related to Algorithm 1 that indicates its more predictable performance compared to BSA.



(a) Noisy case: Average error rate $p = 0.6$ for sparse regime.



(b) Noisy case: Average error rate $p = 0.6$ for linear regime.



(c) Noisy case: Average error rate $p = 0.8$ for linear regime.

Figure 11: Experiment (iii): Noisy case—Average error rate.

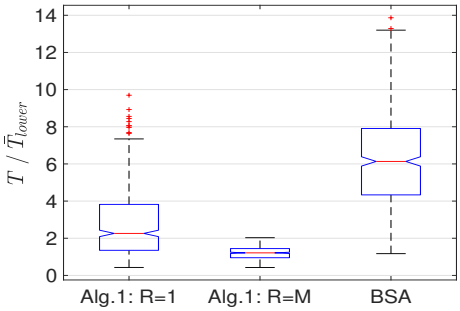


Figure 12: Asymmetric case—Linear regime: Cost efficiency for number of tests.