Group testing for connected communities

Pavlos Nikolopoulos[†] **Sundara Rajan Srinivasavaradhan**[‡] **Tao Guo**[‡]
Christina Fragouli[‡] **Suhas Diggavi**[‡]
[†]EPFL, Switzerland [‡]University of California Los Angeles, USA

Abstract

In this paper, we propose algorithms that leverage a known community structure to make group testing more efficient. We consider a population organized in disjoint communities: each individual participates in a community, and its infection probability depends on the community (s)he participates in. Use cases include families, students who participate in several classes, and workers who share common spaces. Group testing reduces the number of tests needed to identify the infected individuals by pooling diagnostic samples and testing them together. We show that if we design the testing strategy taking into account the community structure, we can significantly reduce the number of tests needed for adaptive and non-adaptive group testing, and can improve the reliability in cases where tests are noisy.

1 Introduction

Group testing pools together diagnostic samples to reduce the number of tests needed to identify infected members in a population. In particular, if in a population of n members we have a small fraction infected (say $k \ll n$ members), we can identify the infected members using as low as $\mathcal{O}(k \log(\frac{n}{k}))$ group tests, as opposed to n individual tests [Du and Hwang, 1993, Aldridge et al., 2019, Kucirka et al., 2020]. Triggered by the need of widespread testing, such techniques are already being explored in the context of Covid-19 [Gollier and Gossner, 2020,Broadfoot, 2020,Ellenberg, 2020,Verdun et al., 2020,Ghosh et al., 2020, Kucirka et al., 2020]. Group testing has a rich history of several decades dating back to R. Dorfman in 1943 and a number of variations and setups have been examined in the literature [Dorfman, 1943, Du and Hwang, 1993, Aldridge et al., 2019, Yaakov Malinovsky, 2016].

The observation we make in this paper is that *we can leverage a known community structure to make group testing more efficient.* The work in group testing we know of, assumes "independent" infections, and ignores that an infection may be governed by community spread; we argue that taking into account the community structure can lead to significant savings. As a use case, consider an apartment building consisting of F families that have practiced social distancing; clearly there is a strong correlation on whether members of the same family are infected or not. Assume that the building management would like to test all members to enable access to common facilities. We ask, what is the most test-efficient way to do so.

Our approach enlarges the regime where group testing can offer benefits over individual testing. Indeed, a limitation of group testing is that it offers fewer or no benefits when k grows linearly with n [Riccio and Colbourn, 2000,Hu et al., 1981,Ungar, 1960, Aldridge, 2019, Aldridge et al., 2019]. Taking into account the community structure allows to identify and remove from the population large groups of infected members, thus reducing their proportion and converting a linear to a sparse regime identification. Essentially, the community structure can guide us on when to use individual, and when group testing.

Our main results are as follows. Assume that n population members are partitioned into F groups that we call *families*, out of which k_f families have at least one infected member.

- We derive a lower bound on the number of tests, which for some regimes increases (almost) linearly with k_f (the number of infected families) as opposed to k (the number of infected members).
- We propose an adaptive algorithm that achieves the lower bound in some parameter regimes.
- We propose a nonadaptive algorithm that accounts

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

for the community structure to reduce the number of tests when some false positive errors can be tolerated.
• We propose a new decoder based on loopy belief propagation that is generic enough to accommodate any community structure and can be combined with any test matrix (encoder) to achieve low error rates.
• We numerically show that leveraging the community structure can offer benefits both when the tests used have perfect accuracy and when they are noisy.

We present our models in Section 2, the lower bound in Section 3, our algorithms for the noiseless case in Section 4, and loopy belief propagation (LBP) decoding in Section 5. Numerical results are in Section 6.

Note: The proofs of our theoretical results (in Sections 3–4) are in the Appendix, along with an extended explanation of the rationale behind our algorithms.

2 Background and notation

2.1 Traditional group testing

Our work extends traditional group testing to infection models that are based on community spread. For this reason, we review here known results from prior work.

Traditional group testing typically assumes a population of n members out of which some are infected. Two infection models are considered: (i) in the *combinatorial model*, a fixed number of infected members k, are selected uniformly at random among all sets of size k; (ii) in the *probabilistic model*, each item is infected independently of all others with probability p, so that the expected number of infected members is $\bar{k} = np$. A group test τ takes as input samples from n_τ individuals, pools them together and outputs a single value: positive if any one of the samples is infected, and negative if none is infected. More precisely, let $U_i = 1$ when individual i is infected and 0 otherwise. Then the traditional group testing output Y_τ takes a binary value calculated as $Y_\tau = \bigvee_{i\in\delta_\tau} U_i$, where $\bigvee$ stands for the `OR` operator (disjunction) and δ_τ is the group of people participating in the test.

The performance of a group testing algorithm is measured by the number of group tests $T = T(n)$ needed to identify the infected members (for the probabilistic model, the expected number of tests needed). Setups that have been explored in the literature include:
• *Adaptive vs. non-adaptive testing*: In adaptive testing, we use the outcome of previous tests to decide what tests to perform next. An example of adaptive testing is *binary splitting*, which implements a form of binary search. Non-adaptive testing constructs, in advance, a *test matrix* $\mathbf{G} \in \{0,1\}^{T\times n}$ where each row corresponds to one test, each column to one member, and the non-zero elements determine the set δ_τ. Although adaptive testing uses less tests than non-adaptive, non-adaptive testing is more practical as all tests can be executed in parallel.
• *Scaling regimes of operation*: assume $k = \Theta(n^\alpha)$, we say we operate in the linear regime if $\alpha = 1$; in the sparse regime if $0 \le \alpha < 1$; in the very sparse regime if k is constant.

Known results. The following are well established results (see [Johnson, 2017, Du and Hwang, 1993, Aldridge et al., 2019] and references therein):
• In the combinatorial model, since T tests allow to distinguish among 2^T combinations of test outputs, then to identify all k infected members without error, we need: $2^T \ge \binom{n}{k} \Leftrightarrow T \ge \log_2 \binom{n}{k}$. This is known as the **counting bound** [Johnson, 2017, Du and Hwang, 1993, Aldridge et al., 2019] and implies that we cannot use less than $T = O(k \log \frac{n}{k})$ tests. In the probabilistic model, a similar bound has been derived for the number of tests needed on average: $T \ge nh_2(p)$, where h_2 is the binary entropy function.
• Noiseless adaptive testing can achieve the counting bound for $k = \Theta(n^\alpha)$ and $\alpha \in [0,1)$; for non-adaptive testing, this is also true of $\alpha \in [0, 0.409]$, if we allow a vanishing (with n) error [Aldridge et al., 2019, Coja-Oghlan et al., 2020, Coja-Oghlan et al., 2020].
• In the linear regime ($\alpha = 1$), group testing offers little benefits over individual testing. In particular, if the infection rate k/n is more than 0.38, group testing does not use fewer tests than 1-by-1 (individual) testing unless high identification-error rates are acceptable [Riccio and Colbourn, 2000, Hu et al., 1981, Ungar, 1960].

2.2 Community and infection models

In this paper, we additionally assume a known community structure: the population can be decomposed in F disjoint groups of individuals that we call *families*. Each family j has M_j members, so that $n = \sum_{j=1}^{F} M_j$. In the symmetric case, $M_j = M$ for all j and $n = FM$. Note, that the term "families" is not limited to real families—we use the same term for any group of people that happen to interact, so that they get infected according to some common infection principle.

We consider the following infection models, that parallel the ones in the traditional setup:
• **Combinatorial Model (I).** k_f of the families are *infected*—namely they have at least one infected member. The rest of the families have no infected members. In each infected family j, there exist k_m^j infected members, with $0 \le k_m^j \le M_j$. The infected families (resp. infected family members) are chosen uniformly at random out of all families (resp. members of the same family). For our analysis, we sometimes consider only the symmetric case, where $k_m^j = k_m$ for each family j.

• **Probabilistic Model (II).** A family is infected with probability q i.i.d. across the families. A member of an infected family j is infected, independently from the other members (and other families), with probability $p_j > 0$. If a family j is not infected, then $p_j = 0$. When $k_m^j = p_j M_j$ the two models behave similarly.

Our goal is two-fold: (a) provide new lower bounds for the number of tests T needed to identify all infected members without error; and (b) design community-aware testing algorithms that are more efficient than traditional group-testing ones, in the sense that they can achieve the same identification accuracy using significantly fewer tests and they can also perform close to the lower bounds in some cases.

2.3 Noisy testing and error probability

In this work we assume that there is no dilution noise, that is, the performance of a test does not depend on the number of samples pooled together. This is a reasonable assumption with genetic RT-PCR tests where even small amounts of viral nucleotides can be amplified to be detectable [Saiki et al., 1985, Kucirka et al., 2020]. However, we do consider noisy tests in our numerical evaluation (Section 6) using a Z-channel noise model[1]. We remark that this is simply a model one may use; our algorithms are agnostic to this and can be used with any other model.

Additionally, some of our identification algorithms may return with errors. For this, we use the following terminology: Let $\hat{U}_i$ denote the estimate of the state of U_i after group testing. *Zero error* captures the requirement that $\hat{U}_i = U_i$ for all $i \in \mathcal{N}$. Vanishing error requires that all error probabilities go to zero with n. Sometimes we also distinguish between *False Negative (FN)* and *False Positive (FP)* errors: FN errors occur when infected members are identified as non-infected (and vice-versa for FP).

2.4 Other related work

The idea of community-aware group testing is explored to some extent in our preprint [Nikolopoulos et al., 2020]. Also, a similar idea of using side-information from contact tracing in decoding is proposed by [Zhu et al., 2020, Goenka et al., 2020], independently from our work. That work is complementary to ours; we focus more on test designs rather than decoding, for which we use well-known algorithms such as COMP and LBP. Finally, test designs, lower bounds and decoding algorithms for independent but not identical priors are investigated by [Li et al., 2014].

The line of work on graph-constrained group testing (see for example [Cheraghchi et al., 2012, Karbasi and Zadimoghaddam, 2012, Luo et al., 2019]) solves the problem of how to design group tests when there are constraints on which samples can be pooled together, provided in the form of a graph; in our case, individuals can be pooled together into tests freely.

3 Lower bound on the number of tests

We compute the minimum number of tests needed to identify all infected members under the zero-error criterion in both community models (I) and (II).

Theorem 1 (Combinatorial community bound)**.** *Consider the combinatorial model (I) (of Section 2.2). Any algorithm that identifies all k infected members without error requires a number of tests T satisfying:*

$$T \geq \log_2 \binom{F}{k_f} + \sum_{j=1}^{k_f} \log_2 \binom{M_j}{k_m^j}. \tag{1}$$

For the symmetric case: $T \geq \log_2 \binom{F}{k_f} + k_f \log_2 \binom{M}{k_m}$.

Observations: We make two observations regarding the combinatorial community bound, in the case where the number of infected family members follows a "strongly" linear regime ($k_m \approx M_j$) and the number of infected families k_f follows a sparse regime (i.e., $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$):

(a) The bound increases almost *linearly* with k_f (the number of infected families), as opposed to k (the overall number of infected members). This is because, if the infection regime about families is sparse, the following asymptotic equivalence holds: $\log_2 \binom{F}{k_f} \sim k_f \log_2 \frac{F}{k_f} \sim (1 - \alpha_f) k_f \log_2 F$.

(b) If additionally to the sparse regime about families, an overall sparse regime ($k = \Theta(n^\alpha)$ for $\alpha \in [0, 1)$) holds, then the community bound may be significantly lower than the counting bound that does not take into account the community structure. Consider, for example, the symmetric case. The asymptotic behavior of the counting bound in the sparse regime is: $\log_2 \binom{n}{k} \sim k \log_2 \frac{n}{k} \sim k_f k_m \log_2 \frac{F}{k_f}$, where the latter is because $k_m \approx M$. So, the ratio of the counting bound to the combinatorial bound scales (as F gets large) as:

$$\frac{\log_2 \binom{n}{k}}{\log_2 \binom{F}{k_f} + k_f \log_2 \binom{M}{k_m}} \sim \frac{k_f k_m \log_2 \frac{F}{k_f}}{k_f \log_2 \frac{F}{k_f}} = k_m. \tag{2}$$

Although simplistic, observation (b) is important for practical reasons. Many times, the population is composed of a large number of families with members that

[1] In a Z-channel noise model, a test output that should be positive, flips and appears as negative with probability z, while a test output that is negative cannot flip. Thus: $\mathbb{P}(Y_\tau = 1 | U_{\delta_\tau}) = \left(\bigvee_{i \in \delta_\tau} U_i\right)(1 - z)$.

have close contacts (e.g. relatives, work colleagues, students who attend the same classes, etc.). In such cases, we do expect that almost all members of infected families are infected (i.e. $k_m \approx M_j$), even though the overall infection regime may still be sparse. Eq. (2) shows the benefits of taking the community structure into account in the test design, in such a case.

Theorem 2 (Probabilistic Community bound)**.** *Consider the probabilistic model (II) (of Section 2.2). Any algorithm that identifies all k infected members without error requires a number of tests T satisfying:*

$$T \geq Fh_2(q) + \sum_{j=1}^{F} qM_j h_2(p_j) - w_j h_2\left(\frac{1-q}{w_j}\right) \quad (3)$$

where $w_j = 1 - q + q(1-p_j)^{M_j}$.

Two observations: (a) If for each family j, p_j and M_j are such that $q(1-p_j)^{M_j} \to 0$ (i.e. the probability of the peculiar event, where a family is labeled "infected" and yet has no infected members, is negligible), the combinatorial and probabilistic bounds are asymptotically equivalent. In particular, using the standard estimates of the binomial coefficient [Ash, 1990, Sec. 4.7], the combinatorial bound in (1) is asymptotically equivalent to $Fh_2(k_f/F) + \sum_{j=1}^{k_f} M_j h_2(k_m^j/M_j)$, which matches the probabilistic bound in (3): $Fh_2(q) + q\sum_{j=1}^{F} M_j h_2(p_j) = Fh_2(\bar{k_f}/F) + \sum_{j=1}^{\bar{k_f}} M_j h_2(\bar{k_m^j}/M_j)$, with $k_f = \bar{k_f} + o(1)$ and $k_m^j = \bar{k_m^j} + o(1)$ in place of their expected values $\bar{k_f} = Fq$ and $\bar{k_m^j}$.

(b) Theorem 2 extends from zero-error recovery to constant-probability recovery by applying Fano's inequality (similarly to Thm 1 of [Li et al., 2014]), and in doing so, the right-hand side of (3) gets multiplied by the desired probability of success $\mathbb{P}(suc)$.

4 Algorithms

4.1 Adaptive algorithm

Alg. 1 describes our algorithm for the fully adaptive case, which consists of two parts (the interested reader may find the detailed rationale for our algorithm in Appendix B). In both parts, we make use of a classic adaptive-group-testing algorithm *AdaptiveTest*(), which is an abstraction for any existing (or future) adaptive group-testing algorithm. We distinguish between 2 different kinds of input for *AdaptiveTest*(): (a) a set of selected members, which is the typical input of group-testing algorithms; (b) a set of selected *mixed samples*. A mixed sample is created by pooling together samples from multiple members that usually have some common characteristic. For example, mixed sample $x(r_j)$ denotes an aggregate sample of a set of representative members r_j from family j. A mixed sample is "positive," if at least one of the members that compose it is infected, and "negative" otherwise. Because in some cases we only care about mixed samples, we can treat them in the same way as individual samples—hence use group testing to identify the infection state of mixed samples as we do for individuals.

Algorithm 1 Adaptive Community Testing

$\hat{U}_i$ is the estimated infection status of member i.
$\hat{U}_x$ is the estimated infection status of a mixed sample x.
SelectRepresentatives() is a function that selects a representative subset from a set of members.
AdaptiveTest() is an adaptive algorithm that tests a set of items (mixed samples or members).

1: **for** $j = 1, \ldots, F$ **do**
2: $\quad r_j = SelectRepresentatives\left(\{i : i \in j\}\right)$
3: **end for**
4: $\left[\hat{U}_{x(r_1)}, \ldots, \hat{U}_{x(r_F)}\right] = AdaptiveTest\left(x(r_1), \ldots, x(r_F)\right)$
5: Set $A := \emptyset$
6: **for** $j = 1, \ldots, F$ **do**
7: $\quad$ **if** $\hat{U}_{x(r_j)} =$ "positive" **then**
8: $\quad\quad$ Use a noiseless, individual test for each family member: $\hat{U}_i = U_i$, $\forall i \in j$.
9: $\quad$ **else**
10: $\quad\quad A := A \cup \{i : i \in j\}$
11: $\quad$ **end if**
12: **end for**
13: $\left\{\hat{U}_i : i \in A\right\} = AdaptiveTest\left(A\right)$
14: **return** $\left[\hat{U}_1, \ldots, \hat{U}_n\right]$

Part 1 (lines 1-4): The goal of this part is to detect the infection *regime* inside each family j, so that the family is tested accordingly at the next part: using group testing, if j is "lightly" infected, or individual testing, otherwise. Our idea is motivated by the result presented in Section 2.1 that group testing is preferable to individual, only if infection rate is low (i.e. $p_j \leq 0.38$). Therefore, the challenge is to accurately detect the infection regime spending only a limited number of tests. In this paper, we limited our exploration to using only one mixed sample in this regard, but more sophisticated techniques are also possible, some of which are discussed in Appendix B.2.

First, a representative subset r_j of family-j members is selected using a sampling function *SelectRepresentatives*() (lines 1-3). Then, a mixed sample $x(r_j)$ is produced for each subset r_j, and an adaptive group-testing algorithm is performed on top of all representative mixed samples (line 4). If our choice of *AdaptiveTest*() offers exact reconstruction

(which is usually the case), then: $\hat{U}_{x(r_j)} = U_{x(r_j)}$.

Part 2 (lines 5-13): We treat $\hat{U}_{x(r_j)}$ as an estimate of the infection regime inside family j: if $\hat{U}_{x(r_j)}$ is positive, then we consider the family to be heavily infected (i.e k_m^j/M_j or $p_j \geq 0.38$), otherwise lightly infected (i.e. k_m^j/M_j or $p_j < 0.38$). Since group testing performs better than individual testing only in the latter case (section 2.1), we use individual testing for each heavily-infected family (lines 7-8), and adaptive group testing for all lightly-infected ones (line 13).

Analysis for the number of tests. We now compute the maximum expected number of tests needed by our algorithm to detect the infection status of all members without error. For simplicity of notation, we present our results through the symmetric case, where $M_j = M$, $k_m^j = k_m$ (combinatorial case) or $p_j = p$ (probabilistic case), and $|r_j| = R$ for all families: Let *SelectRepresentatives*() be a simple function that performs uniform (random) sampling without replacement, and consider 2 choices for the *AdaptiveTest*() algorithm: (i) Hwang's generalized binary splitting algorithm (HGBSA) [Hwang, 1972], which is optimal if the number of infected members of the tested group is known in advance; and (ii), traditional binary-splitting algorithm (BSA) [Sobel and Groll, 1959], which performs well, even if little is known about the number of infected members.

Lemma 1 (Expected number of tests - Symmetric combinatorial model)**.** *Consider the choices (i) and (ii) for the AdaptiveTest() defined above. Alg. 1 succeeds using a maximum expected number of tests:*

$$\begin{aligned}\bar{T}_{(i)} \leq & k_f\phi_c\left(\log_2\frac{F}{k_f\phi_c}+1+M\right)\\ &+k\left(1-\phi_c\right)\left(\log_2\frac{n-k_fM\phi_c}{k\left(1-\phi_c\right)}+1\right) \qquad (4)\\ \bar{T}_{(ii)} \leq & k_f\phi_c\left(\log_2 F+1+M\right)+\\ &+k\left(1-\phi_c\right)\left(\log_2\left(n-k_fM\phi_c\right)+1\right), \qquad (5)\end{aligned}$$

where the inequalities are because of the worst-case performance of HGBSA and BSA, and ϕ_c is the expected fraction of infected families whose mixed sample is positive:

$$\phi_c=\begin{cases}0 & \text{, if } R=0\\ 1-\binom{M-k_m}{R}/\binom{M}{R} & \text{, if } 1\leq R\leq M-k_m\\ 1 & \text{, if } M-k_m<R\leq M.\end{cases}$$

Lemma 2 (Expected number of tests - Symmetric probabilistic model)**.** *If Alg. 1 uses BSA in place of AdaptiveTest(), then it succeeds using a maximum expected number of tests:*

$$\begin{aligned}\bar{T} \leq & Fq\phi_p\left(\log_2 F+1+M\right) \qquad (6)\\ &+nqp\left(1-\phi_p\right)\left(\log_2\left(n\left(1-q\phi_p\right)\right)+1\right), \qquad (7)\end{aligned}$$

where the inequality is due to the worst performance of BSA, and $\phi_p = 1-(1-p)^R$ is the expected fraction of infected families whose mixed sample is positive.

Lemmas 1 and 2 are derived (in Appendix B) as a repeated application of the performance bounds of HGBSA and BSA: if out of n members, k are infected uniformly at random, then HGBSA (resp. BSA) achieves exact identification using at most: $\log_2\binom{n}{k}+k$ (resp. $k\log_2 n+k$) tests [Aldridge et al., 2019, Baldassini et al., 2013].

Observations: (a) If heavily/lightly infected families are detected without errors in Part 1, our algorithm can asymptotically achieve (up to a constant) the lower combinatorial bound of Theorem 1 in particular community structures. We show this via 2 examples:

First, consider a sparse regime for families (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0,1)$) and a moderately linear regime within each family (i.e. $k_m/M \approx 0.5$). In this case: $\log_2\binom{F}{k_f} \sim k_f\log_2(F/k_f)$, $\log_2\binom{M}{k_m} \sim Mh_2(k_m/M) \sim M$ and the bound in (1) becomes: $k_f\left(\log_2 F/k_f + M\right)$. If R is chosen such that all infected families (which are also heavily infected as $k_m/M > 0.38$) are detected without errors (e.g. if $R > M-k_m$), then $\phi_c = 1$; thus, the RHS of (4) becomes almost equal (up to constant k_f) to the lower bound (1).

Second, consider the opposite example, where the infection regime for families is very high, while each separate family is lightly infected. In this case, $k = k_fk_m \approx k_f$; therefore, the lower bound becomes: $T \sim k_f\log_2(F/k_f) + k_fk_m\log_2(M/k_m) \approx k\log_2(n/k)$. If R is chosen such that none of the (lightly infected) families is marked as heavily infected in Part 1 (e.g. if $R=0$, which reduces to using traditional community-agnostic group testing), then $\phi_c = 0$, and the RHS of (4) is almost equal (up to k) to the bound in (1).

(b) The upper bound in (5) shows that our algorithm achieves significant benefits compared to classic BSA when the infected families are heavily-infected and R is chosen such that $\phi_c = 1$ (e.g. $R > M-k_m$); this is because $\bar{T}_{(ii)} \leq k_f\left(\log_2 F+1+M\right) \ll k\log_2 n+k$). Also, it achieves the same performance as BSA, when families are lightly-infected and R is chosen such that $\phi_c = 0$ (e.g. $R=0$); this is because $\bar{T}_{(ii)} \leq k\log_2 n + k$). Since the former case (heavy infection) is more realistic, our algorithm is expected to per. form a lot better than classic group testing in practice.

The examples in observation (a) and the above analysis indicate two things: First, the knowledge of the community structure is more beneficial when families are heavily infected; traditional group testing performs equally well in low infection rates. Our experiments showed that the community structure helps whenever

$p > 0.15$ and the benefits increase with p. Second, a rough estimate of the families' infection rate has to be known a priori in order to optimally choose R. In Appendix B, we demonstrate that this is unavoidable in the symmetric scenario we examine and when only one mixed sample per family is used to identify which families are heavily/lightly infected.

(c) In the most favorable regime for our community-aware group testing, where very few families have almost all their members infected (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$ and $k_m \approx M$), even if R is chosen optimally such that $\phi_c = 1$, the ratio of the expected number of tests needed by Algorithm 1 (see (4)) and HGBSA cannot be less than $1/\log(n/k)$, which upper bounds the benefits one may get. In Appendix B.2, we detail this observation and provide an optimized version of our algorithm that improves upon the gain of $1/\log(n/k)$.

4.2 Two stage algorithm

The adaptive algorithm can be easily implemented as a two-stage algorithm, where we first perform one round of tests, see the outcomes, and then design and perform a second round of tests. The first round of tests implements part 1, checking whether a family is highly infected or not; the second round of tests implements part 2, performing individual tests for the members of the highly infected families, and in parallel, group testing for the members of the remaining families.

As we did before for the adaptive case, we here make use of a classic non-adaptive group-testing algorithm, which we call *NonAdaptiveTest*(), and abstracts any existing (or future) non-adaptive algorithm in the group-testing literature. Thus to translate Alg. 1 to a two-stage algorithm, lines 4 and 13 simply become:

$$4: \left[\hat{U}_{x(r_1)}, ..., \hat{U}_{x(r_F)}\right] = \textit{NonAdaptiveTest}\left(x(r_1), ..., x(r_F)\right)$$
$$13: \left\{\hat{U}_i : i \in A\right\} = \textit{NonAdaptiveTest}\left(A\right). \quad (8)$$

Number of tests: In some regimes, the two-stage algorithm can operate with the same (order) number of tests as the adaptive algorithm, at a cost of a vanishing error probability: for example, for the tests in line 4, if $k_f = \Theta(F^{\alpha_f})$ with $\alpha_f < 0.409$, we can use approximately $(1 - \alpha_f)F^{\alpha_f} \log_2 F$ tests and achieve vanishing error probability leveraging literature non-adaptive algorithms [Aldridge et al., 2019, Scarlett and Cevher, 2016, Johnson et al., 2019, Coja-Oghlan et al., 2020, Coja-Oghlan et al., 2020].

4.3 Non-adaptive algorithm

For simplicity of notation, we describe our non-adaptive algorithm using again the symmetric case.

Test Matrix Structure. Our test matrix $\mathbf{G}$ is divided into two sub-matrices: $\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix}$.

▷ The sub-matrix $\mathbf{G}_1$ of size $T_1 \times n$ identifies the infected families using one mixed sample from each family, similar to line 4 of Alg. 1. We want $\mathbf{G}_1$ to identify all (non-)infected families with small error probability. If the number of tests available is high, we set $T_1 = F$, i.e., we use one row for each family test. Otherwise, in sparse k_f regimes, we set T_1 closer to $O(k_f \log \frac{F}{k_f})$.

▷ The sub-matrix $\mathbf{G}_2$ of size $T_2 \times n$ has a block matrix structure and contains F identity matrices I_M, one for each family. $\mathbf{G}_2$ is designed as follows: (i) each block column contains only one identity matrix I_M, i.e., each member is tested only once; (ii) each block row i ($i \in \{1, 2, \cdots, b\}$) contains c_i identity matrices I_M, i.e., there are c_i members included in the corresponding tests. As a result: $T_2 = bM$. An example with $F = 6$, $b = 3$, $c_1 = 2$, $c_2 = 1$, $c_3 = 3$ is:

$$\mathbf{G}_2 = \begin{bmatrix} I_M & 0_{M\times M} & 0_{M\times M} & I_M & 0_{M\times M} & 0_{M\times M} \\ 0_{M\times M} & I_M & 0_{M\times M} & 0_{M\times M} & 0_{M\times M} & 0_{M\times M} \\ 0_{M\times M} & 0_{M\times M} & I_M & 0_{M\times M} & I_M & I_M \end{bmatrix}.$$

Decoding. From the outcome of the tests in $\mathbf{G}_1$ we identify the $F - k_f$ non-infected families, and proceed to remove the corresponding columns (non-infected members) from $\mathbf{G}_2$. We use the remaining columns of $\mathbf{G}_2$ to identify infected members according to the rules (which follow the logic of combinatorial orthogonal matching pursuit (COMP) decoding [Chan et al., 2014, Cai et al., 2017]):
(i) A member is identified as non-infected if it is included in at least one negative test in $\mathbf{G}_2$.
(ii) All other members, that are only included in positive tests in $\mathbf{G}_2$, are identified as infected.

Error Probability. It is perhaps not hard to see that: after the removal of the columns, the block structure of $\mathbf{G}_2$ helps us obtain a test matrix that is close to an identity matrix – hence perform "almost" individual testing[2]. Also, note that our decoding strategy for $\mathbf{G}_2$ leads to zero FN errors. Building on these ideas, the following lemmas guide us though a design of $\mathbf{G}_2$ that minimizes the (FP) error probability.

• *Requiring zero-error decoding is too rigid:* the optimal solution is the trivial solution that tests each member individually, but this would require $T_2 \geq n$.

• *The symmetric choice $c_i = c$ minimizes the error probability.* As said, we design $\mathbf{G}_2$ such that FP errors are minimized. A FP may happen if identity matrices I_M corresponding to two or more infected families

[2] An extended analysis about $\mathbf{G}_2$ is in Appendix C.2.

appear in the same block row of $\mathbf{G}_2$. In this case, some non-infected members may be included in the same test with infected members from other families and identified as infected by mistake.

Lemma 3. *Under models (I) and (II), the probability that there is some block row containing two or more infected families is:*

$$\mathbb{P}^{I}_{joint} = 1 - \frac{\sum\limits_{|\mathcal{B}|=k_f:\ \mathcal{B}\subseteq\{1,2,\cdots,b\}} \prod\limits_{i\in\mathcal{B}} c_i}{\binom{F}{k_f}}, \tag{9}$$

$$\mathbb{P}^{II}_{joint} = 1 - \prod_{i=1}^{b} \left[(1-q)^{c_i} + c_i q (1-q)^{c_i-1}\right]. \tag{10}$$

The following lemma offers a test-matrix design that minimizes the system FP probability, defined as:

$$\mathbb{P}(\text{any-FP}) \triangleq \mathbb{P}(\exists i : \hat{u}_i = 1 \text{ and } u_i = 0). \tag{11}$$

Lemma 4. *The $\mathbb{P}$(any-FP) is minimized for both models (I) and (II), if $c_i = c$ for all $i \in \{1, \cdots, b\}$.*

Lemma 5. *For $\mathbf{G}_2$ as in Lemma 4, the system FP probability for models (I) and (II) equals:*

$$\mathbb{P}^{I}(\textit{any-FP}) = \left[1 - \frac{1}{\binom{M}{k_m}}\right]\left[1 - \frac{\binom{T_2/M}{k_f}(FM/T_2)^{k_f}}{\binom{F}{k_f}}\right].$$

$$\mathbb{P}^{II}(\textit{any-FP}) = \left[1 - \sum_{i=1}^{M}\left[p^i(1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}\right] \cdot \left[1 - \left((1-q)^{\frac{FM}{T_2}-1}\left(1 - q + \frac{FMq}{T_2}\right)\right)^{T_2/M}\right].$$

$\mathbb{P}$(any-FP) can be pessimistic; a more practical metric is the average fraction of members that are misidentified (error rate): $R(\text{error}) \triangleq |\{i : \hat{u}_i \neq u_i\}|/n$.

Lemma 6. *For $\mathbf{G}_2$ as in Lemma 4, the error rate is calculated for models (I) and (II) as:*

$$R_I(\textit{error}) < \frac{k_f(M-k_m)}{FM} \cdot \mathbb{P}^{I}_{joint}, \tag{12}$$

$$R_{II}(\textit{error}) < (1-p)q\left[1-(1-q)^{c-1}\right]. \tag{13}$$

5 Loopy belief propagation decoder

We now describe our new algorithm for decoding infection status of the individuals (and families). This is accomplished by estimating the posterior probability of the corresponding individual (or family) being infected via *loopy belief propagation* (LBP). LBP computes the posterior marginals exactly when the underlying factor graph describing the joint distribution is a tree (which is rarely the case) [Kschischang et al., 2001]. Nevertheless, it is an algorithm of practical importance and has achieved success on a variety of applications. Also, LBP offers soft information (posterior distributions), which can be proved more useful than hard decisions in the context of disease-spread management.

We use LBP for our probabilistic model, because it is fast and can be easily configured to take into account the community structure leading to more reliable identification. Many inference algorithms exist that estimate the posterior marginals, some of which have also been employed for group testing. For example, GAMP [Zhu et al., 2020] and Monte-Carlo sampling [Cuturi et al., 2020] yield more accurate decoders. However, taking into account the statistical information provided by the community structure was proved not trivial with such decoders. Moreover, the focus of this work is to examine whether benefits from accounting for the community structure (both at the test design and the decoder) exist; hence we think that considering a simple (possibly sub-optimal) decoder based on LBP is a good first step; we defer more complex designs to future work.

We next describe the factor graph and the belief propagation update rules for our probabilistic model (II). Let the infection status of each family j be $V_j \sim \text{Ber}(q)$. Moreover, let $V(U_i)$ denote the family that U_i belongs to.

$$\mathbb{P}(V_1,...,V_F,U_1,...,U_n,Y_1,...,Y_T) = \prod_{j=1}^{F}\mathbb{P}(V_j)\prod_{i=1}^{n}\mathbb{P}(U_i|V(U_i))\prod_{\tau=1}^{T}\mathbb{P}(Y_\tau|U_{\delta_\tau}), \tag{14}$$

where δ_τ is the group of people participating in the test. Equation (14) can be represented by a factor graph, where variable nodes correspond to each random variable V_j, U_i, Y_τ and factor nodes correspond to $\mathbb{P}(V_j), \mathbb{P}(U_i|V(U_i)), \mathbb{P}(Y_\tau|U_{\delta_\tau})$.

Given the result of each test is y_τ, i.e., $Y_\tau = y_\tau$, LBP computes the marginals $\mathbb{P}(V_j = v|Y_1 = y_1,...,Y_T = y_T)$ and $\mathbb{P}(U_i = u|Y_1 = y_1,...,Y_T = y_T)$, by iteratively exchanging messages across the variable and factor nodes. The messages are viewed as *beliefs* about that variable or distributions (a local estimate of $\mathbb{P}$(variable|observations)). Since all random variables are binary, each message is a 2-dimensional vector.

We use the factor graph framework from [Kschischang et al., 2001] to compute the messages: Variable nodes Y_τ continually transmit the message $[0,1]$ if $Y_\tau = 1$ and $[1,0]$ if $Y_\tau = 0$ on its incident edge, at every iteration. Each other variable node (V_j and U_i) uses the following rule: for incident each edge e, the node computes the elementwise product of the messages from every other incident edge e' and transmits this along e. For the factor node messages, we derive closed-form

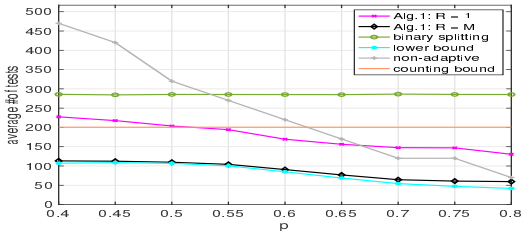

Figure 1: Noiseless case: Average number of tests.

expressions for the sum-product update rules (akin to equation (6) in [Kschischang et al., 2001]). The exact messages are described in Appendix D.

6 Numerical evaluation

In this section, we evaluate the benefits (in terms of number of tests and error rate) from taking the community structure into account in practical scenarios, where noiseless or noisy tests are used.

Experimental setup I: Symmetric. In our simulations, we consider 2 different use cases about the community structure: (Community 1) a neighborhood with $F = 200$ families of $M = 5$ members each, and (Community 2) a university department with $F = 20$ classes of $M = 50$ students each. In each use case, we also examine 2 different infection regimes: (a) a linear regime, where $\bar{k}/n = 0.1$; and (b) a sparse regime, where $\bar{k} = \sqrt{n} = 32$. Finally, we consider both noiseless tests that have perfect accuracy and noisy tests that follow the Z-channel model from Section 2.3. For each scenario, we average over 500 randomly generated community structures, in which the members/students are infected according to the symmetric probabilistic model (II): first a family/class is chosen at random w.p. q to be infected and then each of its members/students gets randomly infected w.p. p.

Results. Our results were similar in all scenarios; for brevity, we show here only the sparse regime. Further results can be found in the Appendix of the supplementary submitted document.

(i) *Noiseless testing – Average number of tests:* In this experiment, 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 nonadaptive algorithm (Section 4.3) that uses $T_1 = F$ tests for $\mathbf{G}_1$ and has FP rate around 0.5%. Alg. 1 assumes no prior knowledge of the number of infected families/classes or members/students, hence uses BSA for the *AdaptiveTest*().

Fig. 1 depicts our results about Community 2 and for $p \in [0.4, 0.8]$. Both versions of Alg. 1 need significantly fewer tests compared to classic BSA, 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 relevant result in the linear regime, was slightly worse: 50-70 tests above the lower bound. Last, the grey line shows number of tests needed by our nonadaptive algorithm; we observe that even that algorithm can perform better than BSA, when $p > 0.55$ and small FP rates are tolerated.

(ii) *Noiseless testing – Average error rate:* We here 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. In each run, we first run our two-stage algorithm (Section 4.2) that uses a classic constant-column-weight test design at each stage and measure the number of tests it requires to achieve zero errors. Then, we use the *same* number of tests to infer the members' infection status through 2 nonadaptive algorithms that account for the community structure either at the test matrix (encoding) part or the decoding and a traditional one that does not consider it at all: "COMP with C-encoder" is our nonadaptive algorithm that uses a COMP decoder as described in Section 4.3; "C-LBP with NC-encoder" is an algorithm that uses classic constant-column-weight test design combined with our LBP decoder form Section 5; and "COMP with NC-encoder" is a traditional nonadaptive algorithm, that we use as a benchmark and uses a constant-column-weight test matrix with a COMP decoder. "C" denotes that the community is taken into account, while "NC" denotes that it is ignored. It is important to note that the number of tests needed by the two-stage algorithm (and therefore all other algorithms) gets lower as p gets large, something that affects the results (as discussed further below).

Fig. 2 depicts the FP and FN error rates[3] (averaged over 500 runs) as a function of $p \in [0.3, 0.9]$ for Community 1. We observe that any community-aware nonadaptive algorithm performs better than traditional nonadaptive group testing (red line) when $p > 0.4$—the absolute performance gap ranges from 0.4% (when $p = 0.3$) to 5.5% (when $p = 0.9$). "COMP with C-encoder" has a stable FP rate across for all p values that was close to 1%, and a zero FN rate by construction. Our LBP decoder, may yield both FN and FP errors. Also, being an approximate inference algorithm, it may produce worse results than COMP when $p \in [0.42, 0.67]$, but performs better when the infection rate is higher.

Fig. 3 examines the effect of the number of tests. Start-

[3]FN rate is the percentage of *infected* individuals identified as negative and vice versa for FP.

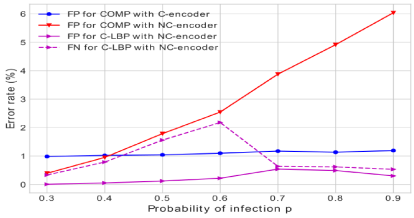

Figure 2: Noiseless case: Average error rate with few tests.

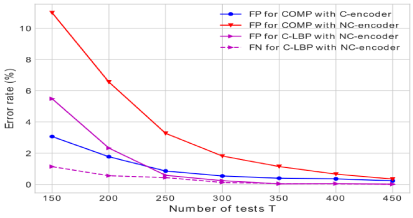

Figure 3: Noiseless case: Average error rate ($p = 0.6$).

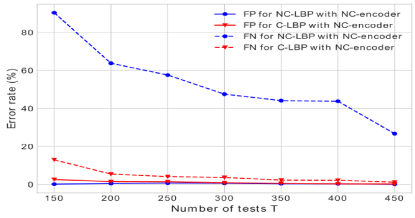

Figure 4: Noisy case: Average error rate ($p = 0.8$).

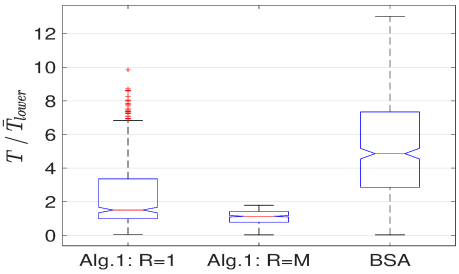

Figure 5: Asymmetric case: Ratio of the number of tests needed to the lower bound (2).

ing from the average number of tests used by the two stage algorithm when $p = 0.6$, we compute the FP and FN rates for larger numbers of tests. Our experiment shows a transition around $T = 240$, after which point "C-LBP with NC-encoder" performs better than "COMP with C-encoder". In fact, "COMP with C-encoder" seems to converges to zero FP errors much slower. This result was common for other p values, the transition just occurred at different T. We therefore conclude that one may use our "COMP with C-encoder" when the number of tests available is limited or they just want to use a simple decoder; otherwise if the testing budget is larger, one should better go with "C-LBP with NC-encoder".

(iii) *Noisy testing:* Assuming the Z-channel noise of Section 2.3 with parameter $z = 0.15$, 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. 4 depicts our results for Community 1 and for a selected $p = 0.8$ and a number of tests as given from the two-stage algorithm of the previous experiments. We observe that the knowledge of the community structure (in C-LBP) reduces both FP and FN rates achieved community-unaware NC-LBP. Especially, FN error rates drop significantly (up to 80% when tests are few), which is important in our context since FN errors lead to further infections. Our results were similar for other p values as well.

Experimental setup II: Asymmetric. In our asymmetric setup, infections follow again the probabilistic model (II), but this time for each family j, M_j and p_j are selected uniformly at random from the intervals $[5, 50]$ and $[0.4, 0.8]$, respectively.

Fig. 5 is a box plot depicting our results for the sparse regime ($q = 3\%$) over 500 randomly generated instances, as described above. The middle line in the box represents the mean and the ends of the box represent the lower and upper quartiles respectively. The crosses represent outlier points. BSA needs on average 5.23× (that can reach up to 13×) more tests compared to the probabilistic bound, while the two versions of Algorithm 1 with $R = 1$ and $R = M$ need only 2.4× and 1.11× (that can reach up to 9.85× and 1.8×) more tests, respectively. Also, the significantly smaller range between the 25-th and 75-th percentiles of the boxplots related to Algorithm 1 indicate a more predictable performance w.r.t. BSA.

7 Conclusions

The new observation we make in this paper is that taking into account infection correlations, as dictated by a known community structure, enables to reduce the number of group tests required to identify the infected members of a population and can improve the identification accuracy when the number of tests is fixed.

In this paper we make this point assuming a nonoverlaping community structure, a specific noise model and binary group testing. We considered a combinatorial and probabilistic model, derived lower bounds on the number of tests needed, explored adaptive, two-stage and non-adaptive algorithms for the noiseless case, as well as algorithms for the noisy case. Our algorithms are not always optimal w.r.t. the lower bounds, but perform significantly better than community-agnostic group testing; per our experiments, they need upto $55 - 75\%$ fewer tests (on average) to achieve the same identification accuracy.

We posit that such benefits are possible in a number of other community or noise or group test models; as an example, the followup work in [Nikolopoulos et al., 2021] illustrates benefits when the families overlap. Understanding what are benefits in more sophisticated models remains as an open question.

Acknowledgments

This work was supported in part by NSF grants #2007714, #1705077 and UC-NL grant LFR-18-548554. We would also like to thank Katerina Argyraki for her ongoing support and the valuable discussions we have had about this project, as well as the anonymous reviewers (especially Reviewer 1) for their constructive comments that helped improve our paper.

References

[Aldridge, 2019] Aldridge, M. (2019). Individual testing is optimal for nonadaptive group testing in the linear regime. *IEEE Trans. Inf. Theory*, 65(4).

[Aldridge et al., 2019] Aldridge, M., Johnson, O., and Scarlett, J. (2019). Group testing: an information theory perspective. *CoRR*, abs/1902.06002.

[Ash, 1990] Ash, R. (1990). *Information theory*. Dover Publications Inc., New York, NY.

[Baldassini et al., 2013] Baldassini, L., Johnson, O., and Aldridge, M. (2013). The capacity of adaptive group testing. In *2013 IEEE International Symposium on Information Theory*, pages 2676–2680.

[Broadfoot, 2020] Broadfoot, M. (2020). Coronavirus test shortages trigger a new strategy: Group screening. See `https://www.scientificamerican.com/article/coronavirus-test-shortages-trigger-a-new-strategy-group-screening2/`.

[Cai et al., 2017] Cai, S., Jahangoshahi, M., Bakshi, M., and Jaggi, S. (2017). Efficient algorithms for noisy group testing. *IEEE Trans. Inf. Theory*, 63(4):2113–2136.

[Chan et al., 2014] Chan, C. L., Jaggi, S., Saligrama, V., and Agnihotri, S. (2014). Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Trans. Inf. Theory*, 60(5):3019–3035.

[Cheraghchi et al., 2012] Cheraghchi, M., Karbasi, A., Mohajer, S., and Saligrama, V. (2012). Graph-constrained group testing. *IEEE Transactions on Information Theory*, 58(1):248–262.

[Coja-Oghlan et al., 2020] Coja-Oghlan, A., Gebhard, O., Hahn-Klimroth, M., and Loick, P. (2020). Information-theoretic and algorithmic thresholds for group testing. *IEEE Trans. Inf. Theory*.

[Coja-Oghlan et al., 2020] Coja-Oghlan, A., Gebhard, O., Hahn-Klimroth, M., and Loick, P. (2020). Optimal group testing. volume 125 of *Proceedings of Machine Learning Research*, pages 1374–1388.

[Cuturi et al., 2020] Cuturi, M., Teboul, O., and Vert, J.-P. (2020). Noisy adaptive group testing using bayesian sequential experimental design. *arXiv preprint arXiv:2004.12508*.

[Dorfman, 1943] Dorfman, R. (1943). The detection of defective members of large population. *The Annals of Mathematical Statistics*, 14:436–440.

[Du and Hwang, 1993] Du, D.-Z. and Hwang, F. (1993). *Combinatorial Group Testing and Its Applications*. Series on Applied Mathematics.

[Ellenberg, 2020] Ellenberg, J. (2020). Five people. one test. this is how you get there. *NYtimes*.

[Ghosh et al., 2020] Ghosh, S. et al. (2020). Tapestry: A single-round smart pooling technique for covid-19 testing. *medRxiv*.

[Goenka et al., 2020] Goenka, R., Cao, S.-J., Wong, C.-W., Rajwade, A., and Baron, D. (2020). Contact tracing enhances the efficiency of covid-19 group testing. *arXiv preprint arXiv:2011.14186*.

[Gollier and Gossner, 2020] Gollier, C. and Gossner, O. (2020). Group testing against covid-19. See `https://www.tse-fr.eu/articles/group-testing-against-covid-19`.

[Hu et al., 1981] Hu, M. C., Hwang, F. K., and Wang, J. K. (1981). A boundary problem for group testing. *SIAM Jour. on Algebraic Discrete Methods*.

[Hwang, 1972] Hwang, F. K. (1972). A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608.

[Johnson et al., 2019] Johnson, O., Aldridge, M., and Scarlett, J. (2019). Performance of group testing algorithms with near-constant tests per item. *IEEE Trans. Inf. Theory*, 65(2):707–723.

[Johnson, 2017] Johnson, O. T. (2017). Strong converses for group testing from finite block- length results. *IEEE Trans. Inf. Theory*, 63(9).

[Karbasi and Zadimoghaddam, 2012] Karbasi, A. and Zadimoghaddam, M. (2012). Sequential group testing with graph constraints. In *2012 IEEE information theory workshop*, pages 292–296. Ieee.

[Kschischang et al., 2001] Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519.

[Kucirka et al., 2020] Kucirka, L. M., Lauer, S. A., Laeyendecker, O., Boon, D., and Lessler, J. (2020). Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based sars-cov-2 tests by time since exposure. *Annals of Internal Medicine*, 173:262–267.

[Li et al., 2014] Li, T., Chan, C. L., Huang, W., Kaced, T., and Jaggi, S. (2014). Group testing with prior statistics. In *2014 IEEE International Symposium on Information Theory*, pages 2346–2350.

[Luo et al., 2019] Luo, S., Matsuura, Y., Miao, Y., and Shigeno, M. (2019). Non-adaptive group testing on graphs with connectivity. *Journal of Combinatorial Optimization*, 38(1):278–291.

[Nikolopoulos et al., 2020] Nikolopoulos, P., Srinivasavaradhan, S. R., Guo, T., Fragouli, C., and Diggavi, S. (2020). Community aware group testing. *arXiv preprint arXiv:2007.08111*.

[Nikolopoulos et al., 2021] Nikolopoulos, P., Srinivasavaradhan, S. R., Guo, T., Fragouli, C., and Diggavi, S. (2021). Group testing for overlapping communities. In *Proc. of the IEEE International Conference on Communications, ICC 2021*.

[Riccio and Colbourn, 2000] Riccio, L. and Colbourn, C. J. (2000). Sharper bounds in adaptive group testing. *Taiwanese Journal of Mathematics*, page 669–673.

[Saiki et al., 1985] Saiki, R. et al. (1985). Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science*, 230(4732):1350–1354.

[Scarlett and Cevher, 2016] Scarlett, J. and Cevher, V. (2016). Phase transitions in group testing. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 40–53. SIAM.

[Sobel and Elashoff, 1975] Sobel, M. and Elashoff, R. (1975). Group testing with a new goal, estimation. *Biometrika*, 62(1):181–193.

[Sobel and Groll, 1959] Sobel, M. and Groll, P. A. (1959). Group testing to eliminate efficiently all defectives in a binomial sample. *The Bell System Technical Journal*, 38(5):1179–1252.

[Ungar, 1960] Ungar, P. (1960). Cutoff points in group testing. *Comm. Pure Appl. Math*, 13:49–54.

[Verdun et al., 2020] Verdun, C. et al. (2020). Group testing for sars-cov-2 allows up to 10-fold efficiency increase across realistic scenarios and testing strategies. *medRxiv*.

[Walter et al., 1980] Walter, S. D., Hildreth, S. W., and Beaty, B. J. (1980). Estimation of infection rates in populations of organisms using pools of variable size. *American Journal of Epidemiology*, 112(1):124–128.

[Yaakov Malinovsky, 2016] Yaakov Malinovsky, P. S. A. (2016). Revisiting nested group testing procedures: new results, comparisons, and robustness. *American Statistician*. See also `https://arxiv.org/abs/1608.06330`.

[Zhu et al., 2020] Zhu, J., Rivera, K., and Baron, D. (2020). Noisy pooled pcr for virus testing. *arXiv preprint arXiv:2004.02689*.