# Group Testing:
# An Information Theory Perspective

Matthew Aldridge[1], Oliver Johnson[2], and Jonathan Scarlett[3]

[1]University of Leeds, m.aldridge@leeds.ac.uk
[2]University of Bristol, O.Johnson@bristol.ac.uk
[3]National University of Singapore, scarlett@comp.nus.edu.sg

October 1, 2019

# Contents

**Abstract**

The group testing problem concerns discovering a small number of defective items within a large population by performing tests on pools of items. A test is positive if the pool contains at least one defective, and negative if it contains no defectives. This is a sparse inference problem with a combinatorial flavour, with applications in medical testing, biology, telecommunications, information technology, data science, and more.

In this monograph, we survey recent developments in the group testing problem from an information-theoretic perspective. We cover several related developments: efficient algorithms with practical storage and computation requirements, achievability bounds for optimal decoding methods, and algorithm-independent converse bounds. We assess the theoretical guarantees not only in terms of scaling laws, but also in terms of the constant factors, leading to the notion of the *rate* of group testing, indicating the amount of information learned per test. Considering both noiseless and noisy settings, we identify several regimes where existing algorithms are provably optimal or near-optimal, as well as regimes where there remains greater potential for improvement.

In addition, we survey results concerning a number of variations on the standard group testing problem, including partial recovery criteria, adaptive algorithms with a limited number of stages, constrained test designs, and sublinear-time algorithms.

# Acknowledgments

We are very grateful to Sidharth Jaggi, who made significant contributions to this project in its early stages and has provided helpful advice and feedback on its later drafts.

We also thank our co-authors on various group testing papers whose results are surveyed in this monograph: Leonardo Baldassini, Volkan Cevher, Karen Gunderson, Thomas Kealy, and Dino Sejdinovic.

We have discussed this project and received helpful feedback on it from many people, including George Atia, Mahdi Cheraghchi, Teddy Furon, Venkatesh Saligrama, and Ashley Montanaro. We received many helpful comments from two anonymous reviewers. Of course, any remaining errors and omissions are our responsibility.

M. Aldridge was supported in part by the Heilbronn Institute for Mathematical Research. J. Scarlett was supported by an NUS Early Career Research Award. O. Johnson would like to thank Maria, Emily, and Becca for their support.

Finally, we would like to thank the editorial board and staff of *Foundations and Trends in Communications and Information Theory* for their support with this monograph.

# Notation

| | |
|---|---|
| $n$ | number of items (Definition 1.1) |
| $k$ | number of defective items (Definition 1.1) |
| $\mathcal{K}$ | defective set (Definition 1.1) |
| $\mathbf{u} = (u_i)$ | defectivity vector: $u_i = \mathbf{1}(i \in \mathcal{K})$, shows if item $i$ is defective (Definition 1.2) |
| $\alpha$ | sparsity parameter in the sparse regime $k = \Theta(n^\alpha)$ (Remark 1.1) |
| $\beta$ | sparsity parameter in the linear regime $k = \beta n$ (Remark 1.1) |
| $T$ | number of tests (Definition 1.3) |
| $\mathsf{X} = (x_{ti})$ | test design matrix: $x_{ti} = \mathtt{1}$ if item $i$ is in test $t$; $x_{ti} = \mathtt{0}$ otherwise (Definition 1.3) |
| $\mathbf{y} = (y_t)$ | test outcomes (Definition 1.4) |
| $\vee$ | Boolean inclusive `OR` (Remark 1.2) |
| $\widehat{\mathcal{K}}$ | estimate of the defective set (Definition 1.5) |
| $\mathbb{P}(\text{err})$ | average error probability (Definition 1.6) |
| $\mathbb{P}(\text{suc})$ | success probability $= 1 - \mathbb{P}(\text{err})$ (Definition 1.6) |
| rate | $\log_2 \binom{n}{k}/T$ (Definition 1.7) |
| $O$, $o$, $\Theta$ | asymptotic 'Big O' notation |
| $R$ | an achievable rate (Definition 1.8) |
| $\overline{R}$ | maximum achievable rate (Definition 1.8) |
| $S(i)$ | the support of column $i$ (Definition 1.9) |
| $S(\mathcal{L})$ | the union of supports $\bigcup_{i \in \mathcal{L}} S(i)$ (Definition 1.9) |
| $p$ | parameter for Bernoulli designs: each item is in each test independently with probability $p$ (Definition 2.2) |
| $L$ | parameter for near-constant tests-per-item designs: each item is in $L$ tests sampled randomly with replacement (Definition 2.3) |
| $\nu$ | test design parameter: for Bernoulli designs, $p = \nu/k$ (Definition 2.2); for near-constant tests-per-item designs, $L = \nu T/k$ (Definition 2.3) |

| | |
|---|---|
| $h(x)$ | binary entropy function: $h(x) = -x\log_2 x - (1-x)\log_2(1-x)$ (Theorem 2.2) |
| $q$ | proportion of defectives (Appendix to Chapter 1) |
| $\overline{k}$ | average number of defectives (Appendix to Chapter 1) |
| $p(y \mid m, \ell)$ | probability of observing outcome $y$ from a test containing $\ell$ defective items and $m$ items in total (Definition 3.1). |
| $\rho$, $\varphi$, $\vartheta$, $\xi$ | noise parameters in binary symmetric (Example 3.1), addition (Example 3.2), dilution/Z channel (Example 3.3, 3.4), and erasure (Example 3.5) models |
| $\overline{\theta}$, $\underline{\theta}$ | threshold parameters in threshold group testing model (Example 3.6) |
| $\Delta$ | decoding parameter for NCOMP (Section 3.4) |
| $\gamma$ | decoding parameter for separate decoding of items (Section 3.5) and information-theoretic decoder (Section 4.2) |
| $C_{\text{chan}}$ | Shannon capacity of communication channel (Theorem 3.1) |
| $m_{i\to t}^{(r)}(u_i)$, $\widehat{m}_{t\to i}^{(r)}(u_i)$ | item-to-test and test-to-item messages (Section 3.3) |
| $\mathcal{N}(i)$, $\mathcal{N}(t)$ | neighbours of an item node and test node (Section 3.3) |
| $\mathsf{X}_{\mathcal{K}}$ | submatrix of columns of $\mathsf{X}$ indexed by $\mathcal{K}$ (Section 4.2.2) |
| $\mathbf{X}_{\mathcal{K}}$ | a single row of $\mathsf{X}_{\mathcal{K}}$ (Section 4.2.2) |
| $V = V(\mathbf{X}_{\mathcal{K}})$ | random number of defective items in the test indicated by $\mathbf{X}$ (Section 4.2.2) |
| $P_{Y\mid V}$ | observation distribution depending on the test design only through $V$ (Equation (4.3)) |
| $S_0$, $S_1$ | partition of the defective set (Equation (4.4)) |
| $\imath$ | information density (Equation (4.6)) |
| $\mathsf{X}_{0,\tau}$, $\mathsf{X}_{1,\tau}$ | sub-matrices of $\mathsf{X}$ corresponding to $(S_0, S_1)$ with $\lvert S_0\rvert = \tau$ (Equation (4.14)) |
| $\mathbf{X}_{0,\tau}$, $\mathbf{X}_{1,\tau}$ | sub-vectors of $\mathbf{X}_{\mathcal{K}}$ corresponding to $(S_0, S_1)$ with $\lvert S_0\rvert = \tau$ |
| $I_\tau$ | conditional mutual information $I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau})$ (Equation (4.16)) |

# Chapter 1

# Introduction to Group Testing

## 1.1 What is group testing?

The 'group testing' problem arose in the United States in the 1940s, when large numbers of men were being conscripted into the army and needed to be screened for syphilis. Since an accurate blood test (the Wassermann test) exists for syphilis, one can take a sample of blood from each soldier, and test it. However, since it is a rare disease, the vast majority of such tests will come back negative. From an information-theoretic point of view, this testing procedure seems inefficient, because each test is not particularly informative.

Robert Dorfman, in his seminal paper of 1943 [61], founded the subject of group testing by noting that, for syphilis testing, the total number of tests needed could be dramatically reduced by pooling samples. That is, one can take blood samples from a 'pool' (or 'group') of many soldiers, mix the samples, and perform the syphilis test on the pooled sample. If the test is sufficiently precise, it should report whether or not any syphilis antigens are present in the combined sample. If the test comes back negative, one learns that all the soldiers in the pool are free of syphilis, whereas if the test comes back positive, one learns that *at least one* of the soldiers in the pool must have syphilis. One can use several such tests to discover which soldiers have syphilis, using fewer tests than the number of soldiers. (The origins of the problem, including the contributions of David Rosenblatt, are described in detail in [62, Section 1.1].)

Of course, this idealized testing model is a mathematical convenience; in practice, a more realistic model could account for sources of error – for example, that a large number of samples of negative blood could dilute syphilis antigens below a detectable level. However, the idealization results in a useful and interesting problem, which we will refer to as *standard noiseless group testing*.

Generally, we say we have $n$ *items* (in the above example, soldiers) of which $k$ are *defective* (have syphilis). A test on a subset of items is returned positive if at least one of the items is defective, and is returned negative is all of the items are nondefective. The central problem of group testing is then the following: Given the number of items $n$ and the number of defectives $k$, how many such

tests $T$ are required to accurately discover the defective items, and how can this be achieved?

As we shall see, the number of tests required depends on various assumptions on the mathematical model used. An important distinction is the following:

**Adaptive vs. nonadaptive** Under adaptive testing, the test pools are designed sequentially, and each one can depend on the previous test outcomes. Under nonadaptive testing, all the test pools are designed in advance, making them amenable to being implemented in parallel. Nearly all the focus of this survey is on nonadaptive testing, though we present adaptive results in Section 1.5 for comparison purposes, and consider the intermediate case of algorithms with a fixed number of stages in Section 5.2.

Within nonadaptive testing, it is often useful to separate the *design* and *decoding* parts of the group testing problem. The design problem concerns the question of how to choose the testing strategy – that is, which items should be placed in which pools. The decoding (or *detection*) problem consists of determining which items are defective given the test designs and outcomes, ideally in a computationally efficient manner.

The number of tests required to achieve 'success' depends on our criteria for declaring success:

**Zero error probability vs. small error probability** With a zero error probability criterion, we want to be certain we will recover the defective set. With a small error probability criterion, we it suffices to recover the defective set with high probability. For example, we may treat the $k$ defective items as being generated uniformly at random without replacement, and provide a tolerance $\epsilon > 0$ on the error probability with respect to this randomness. (In other sparse recovery problems, a similar distinction is sometimes made using the terminology 'for-each setting' and 'for-all setting' [94].) We will mostly focus on the small error probability case, but give zero error results in Section 1.6 for comparison purposes.

**Exact recovery vs. partial recovery** With an exact recovery criterion, we require that every defective item is correctly classified as defective, and every nondefective item is correctly classified as nondefective. With partial recovery, we may tolerate having some small number of incorrectly classified items – perhaps with different demands for false positives (nondefective items incorrectly classed as defective) and false negatives (defective items incorrectly classed as nondefective). For the most part, this survey focuses on the exact recovery criterion; some variants of partial recovery are discussed in Section 5.1.

When considering more realistic settings, it is important to consider group testing models that do not fit into the standard noiseless group testing idealization we began by discussing. Important considerations include:

**Noiseless vs. noisy testing** Under noiseless testing, we are guaranteed that the test procedure works perfectly: We get a negative test outcome if all items in the testing pool are nondefective, and a positive outcome if at least one item in the pool is defective. Under noisy testing, errors can

occur, either according to some specified random model or in an adversarial manner. The first two chapters of this survey describe results in the noiseless case for simplicity, before the noisy case is introduced and discussed from Chapter 3 onwards.

**Binary vs. nonbinary outcomes** Our description of standard group testing involves tests with binary outcomes – that is, positive or negative results. In practice, we may find it useful to consider tests with a wider range of outcomes, perhaps corresponding to some idea of weak and strong positivity, according to the numbers of defective and nondefective items in the test, or even the strength of defectivity of individual items. In such settings, the test matrix may even be non-binary to indicate the 'amount' of each item included in each test. We discuss these matters further in Sections 3.1 and 4.5.

Further to the above distinctions, group testing results can also depend on the assumed distribution of the defective items among all items, and the decoder's knowledge (or lack of knowledge) of this distribution. In general, the true defective set could have an arbitrary prior distribution over all subsets of items. However, the following are important distinctions:

**Combinatorial vs. i.i.d. prior** For mathematical convenience, we will usually consider the scenario where there is a fixed number of defectives, and the defective set is uniformly random among all sets of this size. We refer to this as the *combinatorial* prior (the terminology *hypergeometric group testing* is also used). Alternatively (and perhaps more realistically), one might imagine that each item is defective independently with the same fixed probability $q$, which we call the *i.i.d. prior*. In an Appendix to this chapter, we discuss how results can be transferred from one prior to the other under suitable assumptions. Furthermore, in Section 5.6, we discuss a variant of the i.i.d. prior in which each item has a different prior probability of being defective.

**Known vs. unknown number of defectives** We may wish to distinguish between algorithms that require knowledge of the true number of defectives, and those that do not. An intermediate class of algorithms may be given bounds or approximations to the true number of defectives (see Remark 2.3). In Section 5.3, we discuss procedures that use pooled tests to estimate the number of defective items.

In this survey, we primarily consider the combinatorial prior. Further, for the purpose of proving mathematical results, we will sometimes make the convenient assumption that $k$ is known. However, in our consideration of practical decoding methods in Chapters 2 and 3, we focus on algorithms that do not require knowledge of $k$.

## 1.2 About this survey

The existing literature contains several excellent surveys of various aspects of group testing. The paper by Wolf [197] gives an overview of the early history,

following Dorfman's original work [61], with a particular focus on adaptive algorithms. The textbooks of Du and Hwang [62, 63] give extensive background on group testing, especially on adaptive testing, zero-error nonadaptive testing, and applications. The lecture notes of D'yachkov [64] focus primarily on the zero-error nonadaptive setting, considering both fundamental limits and explicit constructions. For the small-error setting, significant progress was made in the Russian literature in the 1970s and 80s, particularly for nonadaptive testing in the very sparse regime where $k$ is constant as $n$ tends to infinity – the review paper of Malyutov [144] is a very useful guide to this work (see also [64, Ch. 6]).

The focus of this survey is distinct from these previous works. In contrast with the adaptive and zero-error settings covered in [62, 64, 197], here we concentrate on the fundamentally distinct nonadaptive setting with a small (nonzero) error probability. While this setting was also the focus of [144], we survey a wide variety of recent algorithmic and information-theoretic developments not covered there, as well as considering a much wider range of sparsity regimes (that is, scaling behaviour of $k$ as $n \to \infty$). We focus in particular on the more general 'sparse regime' where $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, which comes with a variety challenges compared to the 'very sparse regime' in which $k$ is constant. Another key feature of our survey is that we consider not only order-optimality results, but also quantify the performance of algorithms in terms of the precise constant factors, as captured by the *rate* of group testing. (While much of [63] focuses on the zero-error setting, a variety of probabilistic constructions are discussed in its Chapter 5, although the concept of a rate is not explicitly considered.)

Much of the work that we survey was inspired by the re-emergence of group testing in the information theory community following the paper of Atia and Saligrama [17]. However, to the best of our knowledge, the connection between group testing and information theory was first formally made by Sobel and Groll [180, Appendix A], and was used frequently in the works surveyed in [144].

An outline of the rest of the survey is as follows. In the remainder of this chapter, we give basic definitions and fix notation (Section 1.3), introduce the information-theoretic terminology of rate and capacity that we will use throughout the monograph (Section 1.4), and briefly review results for adaptive (Section 1.5) and zero-error (Section 1.6) group testing algorithms, to provide a benchmark for other subsequent results. In Section 1.7, we discuss some applications of group testing in biology, communications, information technology, and data science. In a technical appendix to the current chapter, we discuss the relationship between two common models for the defective set.

In Chapter 2, we introduce a variety of nonadaptive algorithms for noiseless group testing, and discuss their performance. Chapter 3 shows how these ideas can be extended to various noisy group testing models.

Chapter 4 reviews the fundamental information-theoretic limits of group testing. This material is mostly independent of Chapters 2 and 3 and could be read before them, although readers may find that the more concrete algorithmic approaches of the earlier chapters provides helpful intuition.

In Chapter 5, we discuss a range of variations and extensions of the standard group testing problem. The topics considered are partial recovery of the defective set (Section 5.1), adaptive testing with limited stages (Section 5.2), counting the number of defective items (Section 5.3), decoding algorithms that run in sublinear time (Section 5.4), the linear sparsity regime $k = \Theta(n)$ (Sec-

tion 5.5), group testing with more general prior distributions on the defective set (Section 5.6), explicit constructions of test designs (Section 5.7), group testing under constraints on the design (Section 5.8), and more general group testing models (Section 5.9). Each of these sections gives a brief outline of the topic with references to more detailed work, and can mostly be read independently of one another. Finally, we conclude in Chapter 6 with a partial list of interesting open problems.

For key results in the survey, we include either full proofs or proof sketches (for brevity). For other results that are not our main focus, we may omit proofs and instead provide pointers to the relevant references.

## 1.3 Basic definitions and notation

We now describe the group testing problem in more formal mathematical language.

**Definition 1.1.** We write $n$ for the number of items, which we label as $\{1, 2, \dots, n\}$. We write $\mathcal{K} \subset \{1, 2, \dots, n\}$ for the set of defective items (the *defective set*), and write $k = |\mathcal{K}|$ for the number of defectives.

**Definition 1.2.** We write $u_i = \mathtt{1}$ to denote that item $i \in \mathcal{K}$ is defective, and $u_i = \mathtt{0}$ to denote that $i \notin \mathcal{K}$ is nondefective. In other words, we define $u_i$ as an indicator function via $u_i = \mathbf{1}\{i \in \mathcal{K}\}$. We then write $\mathbf{u} = (u_i) \in \{\mathtt{0}, \mathtt{1}\}^n$ for the *defectivity vector*. (In some contexts, an uppercase $\mathbf{U}$ will denote a *random* defectivity vector.)

*Remark* 1.1. We are interested in the case that the number of items $n$ is large, and accordingly consider asymptotic results as $n \to \infty$. We use the standard 'Big O' notation $O(\cdot)$, $o(\cdot)$, and $\Theta(\cdot)$ to denote asymptotic behaviour in this limit, and we write $f_n \sim g_n$ to mean that $\lim_{n\to\infty} \frac{f_n}{g_n} = 1$. We consider three scaling regimes for the number of defectives $k$:

**The very sparse regime:** $k$ is constant (or bounded as $k = O(1)$) as $n \to \infty$;

**The sparse regime:** $k$ scales sublinearly as $k = \Theta(n^\alpha)$ for some *sparsity parameter* $\alpha \in [0, 1)$ as $n \to \infty$.

**The linear regime:** $k$ scales linearly as $k \sim \beta n$ for $\beta \in (0, 1)$ as $n \to \infty$.

We are primarily interested in the case that defectivity is rare, where group testing has the greatest gains, so we will only briefly review the linear regime in Section 5.5. A lot of early work considered only the very sparse regime, which is now quite well understood – see, for example, [144] and the references therein – so we shall concentrate primarily on the wider sparse regime.

The case $\alpha = 0$, which covers the very sparse regime, usually behaves the same as small $\alpha$ but sometimes requires slightly different analysis to allow for the fact that $k$ may not tend to $\infty$. Hence, for reasons of brevity, we typically only explicitly deal with the cases $\alpha \in (0, 1)$.

We assume for the most part that the true defective set $\mathcal{K}$ is uniformly random from the $\binom{n}{k}$ sets of items of size $k$ (the 'combinatorial prior'). The assumption that $k$ is known exactly is often mathematically convenient, but unrealistic in most applications. For this reason, in Chapters 2 and 3 we focus

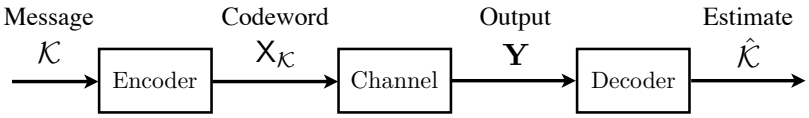



Figure 1.1: Group testing interpreted as a channel coding problem. The notation $\mathsf{X}_{\mathcal{K}}$ denotes the $T \times k$ sub-matrix of $\mathsf{X}$ obtained by keeping only the $k$ columns indexed by $\mathcal{K}$, and the output $\mathbf{Y}$ is the 'OR' of these $k$ columns.

primarily on decoding algorithms that do not require knowledge of $k$. However, there exist nonadaptive algorithms that can estimate the number of defectives using $O(\log n)$ tests (see Section 5.3 below), which could form the first part of a two-stage algorithm, if permitted.

**Definition 1.3.** We write $T = T(n)$ for the *number of tests* performed, and label the tests $\{1, 2, \dots, T\}$. To keep track of the design of the test pools, we write $x_{ti} = \mathtt{1}$ to denote that item $i \in \{1, 2, \dots, n\}$ is in the pool for test $t \in \{1, 2, \dots, T\}$, and $x_{ti} = \mathtt{0}$ to denote that item $i$ is not in the pool for test $t$. We gather these into a matrix $\mathsf{X} \in \{\mathtt{0}, \mathtt{1}\}^{T \times n}$, which we shall refer to as the *test matrix* or *test design*.

To our knowledge, this matrix representation was introduced by Katona [118]. It can be helpful to think of group testing in a channel coding framework, where the particular defective set $\mathcal{K}$ acts like the source message, finding the defective set can be thought of as decoding, and the matrix $\mathsf{X}$ acts like the codebook. See Figure 1.1 for an illustration. See also [144] for a related interpretation of group testing as a type of *multiple-access channel*.

Similarly to the channel coding problem, explicit deterministic matrix designs often fail to achieve even order-optimal performance (see Sections 1.6 and 5.7 for discussion). Following the development and successes of randomized channel codes (discussed further below), it is therefore a natural development to consider randomized test designs. We will use a capital $X_{ti}$ to denote the random entries of a random testing matrix. Some designs of interest include the following:

**Bernoulli design** In a Bernoulli design, each item is included in each test independently at random with some fixed probability $p$. That is, we have $\mathbb{P}(X_{ti} = \mathtt{1}) = p$ and $\mathbb{P}(X_{ti} = \mathtt{0}) = 1 - p$, i.i.d. over $i \in \{1, 2, \dots, n\}$ and $t \in \{1, 2, \dots, T\}$. Typically the parameter $p$ is chosen to scale as $p = \Theta(1/k)$. See Definition 2.2 for more details.

**Constant tests-per-item** In a constant tests-per-item design, each item is included in some fixed number $L$ of tests, with the $L$ tests for each item chosen uniformly at random, independent from the choices for all other items. In terms of the testing matrix $\mathsf{X}$, we have independent columns of constant weight. Typically the parameter $L$ is chosen to scale as $L = \Theta(T/k)$. In fact, it is often more mathematically convenient to analyse the similar *near-constant tests-per-item* (near-constant column weight) design, where the $L$ tests for each item are chosen uniformly at random *with replacement* – see Definition 2.3 for more details.

**Doubly regular design** One can also consider designs with both a constant number $L$ of tests-per-item (column weight) and also a constant number $m$ of items-per-test (row weight), with $nL = mT$. The matrix is picked uniformly at random according to these constraints. One can again consider 'near-constant' versions, where sampling is with replacement. Again, $L = \Theta(T/k)$ (or equivalently, $m = \Theta(n/k)$) is a useful scaling. We will not focus on these designs in this monograph, but mention that they were studied in the papers [148, 192], among others.

As hinted above, these random constructions can be viewed as analogous to random coding in channel coding, which is ubiquitous for proving achievability bounds. However, while standard random coding designs in channel coding are impractical due to the exponential storage and computation required, the above designs can still be practical, since the random matrix only contains $T \times n$ entries. In this sense, the constructions are in fact more akin to random linear codes, random LDPC codes, and so on.

We write $y_t \in \{\mathtt{0}, \mathtt{1}\}$ for the *outcome* of test $t \in \{1, 2, \dots, T\}$, where $y_t = \mathtt{1}$ denotes a positive outcome and $y_t = \mathtt{0}$ a negative outcome. Recall that in the standard noiseless model, we have $y_t = \mathtt{0}$ if all items in the test are nondefective, and $y_t = \mathtt{1}$ if at least one item in the test is defective. Formally, we have the following.

**Definition 1.4.** Fix $n$ and $T$. Given a defective set $\mathcal{K} \subset \{1, 2, \dots, n\}$ and a test design $\mathsf{X} \in \{\mathtt{0}, \mathtt{1}\}^{T \times n}$, the *standard noiseless group testing model* is defined by the outcomes

$$y_t = \begin{cases} \mathtt{1} & \text{if there exists } i \in \mathcal{K} \text{ with } x_{ti} = \mathtt{1}, \\ \mathtt{0} & \text{if for all } i \in \mathcal{K} \text{ we have } x_{ti} = \mathtt{0}. \end{cases} \tag{1.1}$$

We write $\mathbf{y} = (y_t) \in \{\mathtt{0}, \mathtt{1}\}^T$ for the vector of test outcomes.

*Remark* 1.2. A concise way to write (1.1) is using the Boolean inclusive `OR` (or disjunction) operator $\vee$, where $\mathtt{0} \vee \mathtt{0} = \mathtt{0}$ and $\mathtt{0} \vee \mathtt{1} = \mathtt{1} \vee \mathtt{0} = \mathtt{1} \vee \mathtt{1} = \mathtt{1}$. Then,

$$y_t = \bigvee_{i \in \mathcal{K}} x_{ti}; \tag{1.2}$$

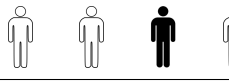
or, with the understanding that `OR` is taken component-wise,

$$\mathbf{y} = \bigvee_{i \in \mathcal{K}} \mathbf{x}_i. \tag{1.3}$$

Using the defectivity vector notation of Definition 1.2, we can rewrite (1.2) in analogy with matrix multiplication as

$$y_t = \bigvee_{i} x_{ti} u_i. \tag{1.4}$$

Note that the nonlinearity of the $\vee$ operation is what gives group testing its specific character, as opposed to models based on exclusive `OR`, or mod-2 addition. Indeed, we can consider (1.4) to be a nonlinear 'Boolean counterpart' to the well-known compressed sensing problem [92].



| | | | | | | | | Outcome |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | ① | 1 | 0 | 0 | 0 | 0 | Positive |
| 0 | 0 | 0 | 0 | ① | 1 | 1 | 1 | Positive |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Negative |
| 0 | 0 | ① | 0 | 0 | 0 | 0 | 0 | Positive |
| 0 | 0 | ① | 0 | ① | 1 | 0 | 0 | Positive |
| 0 | 0 | 0 | 0 | ① | 0 | 0 | 0 | Positive |

Figure 1.2: Example of a group testing procedure and its outcomes. Icons for defective individuals (items 3 and 5) are filled, and icons for nondefective individuals are unfilled. The testing matrix $\mathsf{X}$ is shown beneath the individuals, where elements $x_{ti}$ are circled for emphasis if $x_{ti} = 1$ and individual $i$ is defective. Hence, a test is positive if and only if it contains at least one circled 1.

| ? | ? | ? | ? | ? | ? | ? | ? | $\mathbf{y}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Figure 1.3: Group testing inference problem. We write 1 for a positive test and 0 for a negative test, but otherwise the matrix $\mathsf{X}$ is exactly as in Figure 1.2 above. The defectivity status of the individuals is now unknown, and we hope to infer it from the outcomes $\mathbf{y}$ and matrix $\mathsf{X}$.

We illustrate a simple group testing procedure in Figure 1.2, where the defective items are represented by filled icons, and so it is clear that the positive tests are those containing at least one defective item.

Given the test design $\mathsf{X}$ and the outcomes $\mathbf{y}$, we wish to find the defective set. Figure 1.3 represents the inference problem we are required to solve – the defectivity status of particular individuals is hidden, and we are required to infer it from the matrix $\mathsf{X}$ and the vector of outcomes $\mathbf{y}$. In Figure 1.3, we write $\mathtt{1}$ for a positive test and $\mathtt{0}$ for a negative test. In general we write $\hat{\mathcal{K}} = \hat{\mathcal{K}}(\mathsf{X}, \mathbf{y})$ for our estimate of the defective set.

**Definition 1.5.** A *decoding* (or detection) *algorithm* is a (possibly randomized) function $\hat{\mathcal{K}} : \{\mathtt{0},\mathtt{1}\}^{T\times n} \times \{\mathtt{0},\mathtt{1}\}^{T} \to \mathcal{P}(\{1,2,\dots,n\})$, where the power-set $\mathcal{P}(\{1,2,\dots,n\})$ is the collection of subsets of items.

Under the exact recovery criterion, we succeed when $\hat{\mathcal{K}} = \mathcal{K}$, while under partial recovery, we succeed if $\hat{\mathcal{K}}$ is close to $\mathcal{K}$ in some predefined sense (see Section 5.1). Since we focus our attention on the former, we provide its formal definition as follows.

**Definition 1.6.** Under the exact recovery criterion, the (*average*) *error probability* for noiseless group testing with a combinatorial prior is

$$\mathbb{P}(\text{err}) := \frac{1}{\binom{n}{k}} \sum_{\mathcal{K}\,:\,|\mathcal{K}|=n} \mathbb{P}\big(\hat{\mathcal{K}}(\mathsf{X},\mathbf{y}) \neq \mathcal{K}\big), \tag{1.5}$$

where $\mathbf{y}$ is related to $\mathsf{X}$ and $\mathcal{K}$ via the group testing model and the probability $\mathbb{P}$ is over the randomness in the test design $\mathsf{X}$ (if randomized), the group testing model (if random noise is present), and the decoding algorithm $\hat{\mathcal{K}}$ (if randomized). We call $\mathbb{P}(\text{suc}) := 1 - \mathbb{P}(\text{err})$ the *success probability*.

We note that this average error probability refers to an average over a uniformly distributed choice of defective set $\mathcal{K}$, where we can think of this randomness as being introduced by nature. Even in a setting where the true defective set $\mathcal{K}$ is actually deterministic, this can be a useful way to think of randomness in the model. Since the outcomes of the tests only depend on the columns of the test matrix $\mathsf{X}$ corresponding to $\mathcal{K}$, the same average error probability is achieved even for a fixed $\mathcal{K}$ by any exchangeable matrix design (that is, one where the distribution of $\mathsf{X}$ is invariant under uniformly-chosen column permutations). This includes Bernoulli, near-constant tests-per-item, and doubly regular designs, as well as any deterministic matrix construction acted on by uniformly random column permutations.

## 1.4 Counting bound and rate

Recall that the goal is, given $n$ and $k$, to choose $\mathsf{X}$ and $\hat{\mathcal{K}}$ such that $T$ is as small as possible, while keeping the error probability $\mathbb{P}(\text{err})$ small.

Supposing momentarily that we were to require an error probability of *exactly* zero, a simple counting argument based on the pigeonhole principle reveals that we require $T \geq \log_2 \binom{n}{k}$: There are only $2^T$ combinations of test results, but there are $\binom{n}{k}$ possible defective sets that each must give a different set of

results. This argument is valid regardless of whether the test design is adaptive or nonadaptive.

The preceding argument extends without too much difficulty to the nonzero error probability case. For example, Chan *et al.* [33] used an argument based on Fano's inequality to prove that

$$\mathbb{P}(\text{suc}) \leq \frac{T}{\log_2 \binom{n}{k}}, \tag{1.6}$$

which they refer to as 'folklore', while Baldassini *et al.* gave the following tighter bound on the success probability [20, Theorem 3.1] (see also [112])

**Theorem 1.1** (Counting bound)**.** *Any algorithm (adaptive or nonadaptive) for recovering the defective set with $T$ tests has success probability satisfying*

$$\mathbb{P}(\text{suc}) \leq \frac{2^T}{\binom{n}{k}}. \tag{1.7}$$

*In particular, $\mathbb{P}(\text{suc}) \to 0$ as $n \to \infty$ whenever $T \leq (1-\eta)\log_2 \binom{n}{k}$ for arbitrarily small $\eta > 0$.*

From an information-theoretic viewpoint, this result essentially states that since the prior uncertainty is $\log_2 \binom{n}{k}$ for a uniformly random defective set, and each test is a yes/no answer revealing at most 1 bit of information, we require at least $\log_2 \binom{n}{k}$ tests. Because the result is based on counting the number of defective sets, we refer to it as the *counting bound*, often using this terminology for both the asymptotic and nonasymptotic versions when the distinction is clear from the context.

With this mind, it will be useful to think about how many bits of information we learn (on average) per test. Using an analogy with channel coding, we shall call this the *rate* of group testing. In general, if the defective set $\mathcal{K}$ is chosen from some underlying random process with entropy $H$, then for a group testing strategy with $T$ tests, we define the rate to be $H/T$. In particular, under a combinatorial prior, where the defective set is chosen uniformly from the $\binom{n}{k}$ possible sets, the entropy is $H = \log_2 \binom{n}{k}$, leading to the following definition.

**Definition 1.7.** Given a group testing strategy under a combinatorial prior with $n$ items, $k$ defective items, and $T$ tests, we define the *rate* to be

$$\text{rate} := \frac{\log_2 \binom{n}{k}}{T}. \tag{1.8}$$

This definition was first proposed for the combinatorial case by Baldassini, Aldridge and Johnson [20], and extended to the general case (see Definition 5.2) in [120]. This definition generalizes a similar earlier definition of rate by Malyutov [143, 144], which applied only in the very sparse ($k$ constant) regime.

We note the following well-known bounds on the binomial coefficient (see for example [47, p. 1186]):

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{\mathrm{e}n}{k}\right)^k. \tag{1.9}$$

Thus, we have the asymptotic expression

$$\log_2 \binom{n}{k} = k \log_2 \frac{n}{k} + O(k), \tag{1.10}$$

and in the sparse regime $k = \Theta(n^\alpha)$ for $\alpha \in [0, 1)$, we have the asymptotic equivalence

$$\log_2 \binom{n}{k} \sim k \log_2 \frac{n}{k} \sim (1-\alpha) k \log_2 n = \frac{(1-\alpha)}{\ln 2} k \ln n. \tag{1.11}$$

Thus, to achieve a positive rate in this regime, we seek group testing strategies with $T = O(k \log n)$ tests. In contrast, in Section 5.5, we will observe contrasting behaviour of the binomial coefficient in the linear regime $k \sim \beta n$, expressed in (5.8).

**Definition 1.8.** Consider a group testing problem, possibly with some aspects fixed (for example, the random test design or the decoding algorithm), in a setting where the number of defectives scales as $k = k(n)$ according to some function (e.g., $k(n) = \Theta(n^\alpha)$ with $\alpha \in (0, 1)$).

1. We say a rate $R$ is *achievable* if, for any $\delta, \epsilon > 0$, for $n$ sufficiently large there exists a group testing strategies with a number of tests $T = T(n)$ such that the rate satisfies

   $$\text{rate} = \frac{\log_2 \binom{n}{k}}{T} > R - \delta, \tag{1.12}$$

   and the error probability $\mathbb{P}(\text{err})$ is at most $\epsilon$.

2. We say a rate $R$ is *zero-error achievable* if, for any $\delta > 0$, for $n$ sufficiently large, there exists a group testing strategy with a number of tests $T = T(n)$ such that the rate exceeds $R - \delta$, and $\mathbb{P}(\text{err}) = 0$.

3. Given a random or deterministic test matrix construction (design), we define the *maximum achievable rate* to be the supremum of all achievable rates that can be achieved by any decoding algorithm. We sometimes also use this terminology when the decoding algorithm is fixed. For example, we write $\overline{R}_{\text{Bern}}$ for the maximum rate achieved by Bernoulli designs and any decoding algorithm, and $\overline{R}_{\text{Bern}}^{\text{COMP}}$ for the maximum rate achieved by Bernoulli rates using the COMP algorithm (to be described in Section 2.3).

4. Similarly, the *maximum zero-error achievable rate* is the supremum of all zero-error achievable rates for a particular design.

5. We define the *capacity* $C$ to be the supremum of all achievable rates, and the *zero-error capacity* $C_0$ to be the supremum of all zero-error achievable rates. Whereas the notion of *maximum achievable rate* allows test design and/or decoding algorithm to be fixed, the definition of capacity optimizes over both.

Note that these notions of rate and capacity may depend on the scaling of $k(n)$. In our achievability and converse bounds for the sparse regime $k = \Theta(n^\alpha)$, the maximum rate will typically vary with $\alpha$, but will not depend on the implied constant in the $\Theta(\cdot)$ notation.

*Remark* 1.3. Note that the counting bound (Theorem 1.1) gives us a universal upper bound $C \leq 1$ on capacity. In fact, it also implies the so-called strong converse: The error probability $\mathbb{P}(\text{err})$ tends to 1 when $T \leq (1-\eta)\log_2 \binom{n}{k}$ for arbitrarily small $\eta > 0$, which corresponds to a rate $R \geq 1/(1-\eta) > 1$.

We are interested in determining when the upper bound $C = 1$ can or cannot be achieved, as well as determining how close practical algorithms can come to achieving it. (We discuss what we mean by 'practical' in this context in Section 2.1.)

We will observe the following results for noiseless group testing in the sparse regime $k = \Theta(n^\alpha)$, which are illustrated in Figure 1.4:

**Adaptive testing** is very powerful, in that both the zero-error and small-error capacity equal $C_0 = C = 1$ for all $\alpha \in [0, 1)$ (see Section 1.5).

**Zero-error nonadaptive testing** is much weaker, in the sense that the zero-error capacity is $C_0 = 0$ for all $\alpha \in (0, 1)$ (see Section 1.6).

**Small-error nonadaptive testing** is more complicated. The capacity is $C = 1$ for $\alpha \in [0, 0.409]$; this is achievable with a Bernoulli design for $\alpha < 1/3$ (Theorem 4.1), and with a (near-)constant column weight design for the full interval (Theorem 4.2). The capacity is unknown for $\alpha \in (0.409, 1)$, for which the best known achievable rate is $(\ln 2)\frac{1-\alpha}{\alpha}$ (Theorem 4.2). Finding the capacity of small-error nonadaptive group testing for $\alpha \in (0.409, 1)$ is a significant open problem. We discuss these results further in Chapter 4, and discuss rates for practical algorithms in Chapter 2.

This survey is mostly concerned with nonadaptive group testing with small error probability, starting with the noiseless setting (Chapter 2). Later in the monograph, we will expand our attention to the noisy nonadaptive setting (Chapter 3), partial recovery criteria (Section 5.1), 'semi-adaptive' testing with limited stages (Section 5.2), and the linear regime $k = \Theta(n)$ (Section 5.5), among others.

It will be useful to compare the results to come with various well-established results for adaptive testing and for zero-error nonadaptive testing (in the noiseless setting). The next two sections provide a brief review of these two models.

## 1.5 A brief review of noiseless adaptive group testing

Much of the early group testing literature focused on adaptive procedures. Dorfman's original paper [61] proposed a simple procedure where items were partitioned into sets that undergo primary testing: A negative test indicates that all the items in that set are definitely nondefective, whereas for within the positive tests, all items are subsequently tested individually. It is easily checked (see, for example, [127], [77, Ex. 26, Section IX.9]) that the optimal partition (assuming that $k$ is known) comprises $\sqrt{nk}$ subsets, each of size $\sqrt{n/k}$. Dorfman's procedure therefore requires requires at most

$$T = \sqrt{nk} + k\sqrt{\frac{n}{k}} = 2\sqrt{nk} \tag{1.13}$$

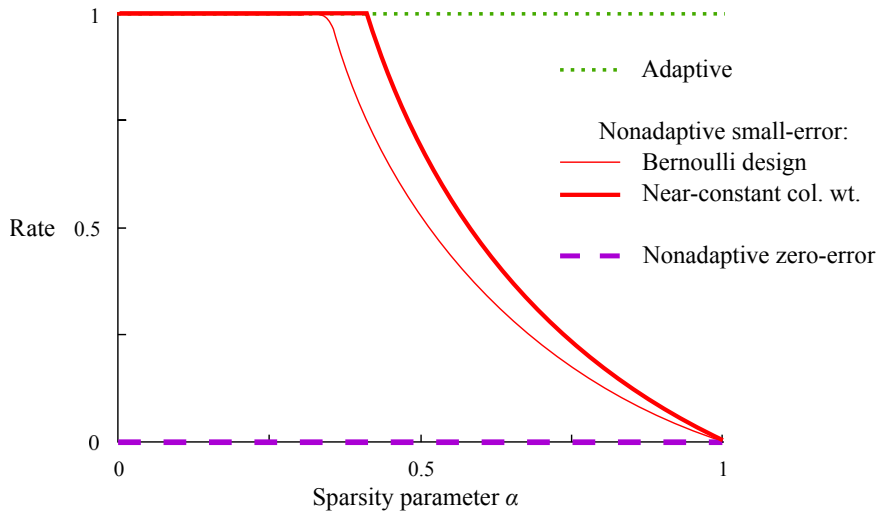



Figure 1.4: Achievable rates for noiseless group testing with $k = \Theta(n^\alpha)$ for a sparsity parameter $\alpha \in (0, 1)$: the adaptive capacity $C = 1$; the nonadaptive zero-error capacity is $C_0 = 0$; and the achievable rates for nonadaptive small-error group testing are given in Theorem 4.1 for Bernoulli designs and Theorem 4.2 for near-constant column weight designs. These achieve the capacity $C = 1$ for $\alpha \leq 1/3$ and $\alpha < 0.409$ respectively.

tests.

Sterrett [184] showed that improvements arise by testing items in a positive test individually until a defective item is found, and then re-testing all remaining items in the set together. Li [127] and Finucan [78] provided variants of Dorfman's scheme based on multi-stage adaptive designs.

The work of Sobel and Groll [180] introduced the crucial idea of recursively splitting the set, with their later paper [181] showing that such a procedure performs well even if the number of defectives is unknown. We will describe the procedure of *binary splitting*, which lies at the heart of many adaptive algorithms. Suppose we have a set $A$ of items. We can test whether $A$ contains any defectives, and, if it does, discover a defective item through binary splitting as follows.

**Algorithm 1.1** (Binary splitting)**.** Given a set $A$:

1. Initialize the algorithm with set $A$. Perform a single test containing every item in $A$.
2. If the preceding test is negative, $A$ contains no defective items, and we halt. If the test is positive, continue.
3. If $A$ consists of a single item, then that item is defective, and we halt. Otherwise, pick half of the items in $A$, and call this set $B$. Perform a single test of the pool $B$.
4. If the test is positive, set $A := B$. If the test is negative, set $A := A \setminus B$. Return to Step 3.

The key idea is to observe that even if the test in Step 3 is negative, we still gain information from it; since $A$ contained at least one defective (as confirmed by Steps 1 and 2), and $B$ contained no defective, we can be certain that $A \setminus B$ contains at least one defective.

In Step 3, when picking the set $B$ to be half the size of $A$, we can round $|A|/2$ in either direction. Since the size of the set $A$ essentially halves on each loop through the algorithm, we see that binary splitting finds a defective item in at most $\lceil \log_2 |A| \rceil$ adaptive tests, or confirms there are no defective items in a single test. We conclude the following.

**Theorem 1.2.** *We can find all $k$ defectives in a set of $n$ items by repeated rounds of Algorithm 1.1, using a total of $k \log_2 n + O(k)$ adaptive tests, even when $k$ is unknown. In the sparse regime $k = \Theta(n^\alpha)$ with $\alpha \in [0, 1)$, this gives an achievable rate of $1 - \alpha$.*

*Proof.* In the first round, we initialize the binary splitting algorithm using $A = \{1, 2, \dots, n\}$, and find the first defective (denoted by $d_1$) using at most $\lceil \log_2 n \rceil$ tests.

In subsequent rounds, if we have found defectives $\{d_1, \dots, d_r\}$ in the first $r$ rounds, then the $(r+1)$-th round of Algorithm 1.1 is initialized with $A = \{1, 2, \dots, n\} \setminus \{d_1, d_2, \dots, d_r\}$. We perform one further test to determine whether $\{1, 2, \dots, n\} \setminus \{d_1, d_2, \dots, d_r\}$ contains at least one defective. If not, we are done. If it does, we find the next defective item using at most $\lceil \log_2(n-r) \rceil \leq \lceil \log_2 n \rceil$ tests. We repeat the procedure until no defective items remain, and the result follows. □

Note that for $\alpha > 0$, this rate $1 - \alpha$ fails to match the counting bound $C \leq 1$. However, we can reduce the number of tests required to $k \log_2(n/k) + O(k)$, thus raising the rate to 1 for all $\alpha \in [0, 1)$, by using a variant of Hwang's generalized binary splitting algorithm [106]. The key idea is to notice that, unless there are very few defectives remaining, the first tests in each round of the repeated binary splitting algorithm are overwhelmingly likely to be positive, and are therefore very uninformative. A better procedure is as follows:

**Algorithm 1.2.** Divide the $n$ items into $k$ subsets of size $n/k$ (rounding if necessary), and apply Algorithm 1.1 to each subset in turn.

Note that each of these subsets contains an average of one defective. Using the procedure above, if the $i$-th subset contains $k_i$ defectives, taking $k = k_i$ and $n = n/k$ in Theorem 1.2, we can find them all using $k_i \log_2(n/k) + O(k_i)$ tests, or confirm the absence of any defectives with one test if $k_i = 0$. Adding together the number of tests over each subset, we deduce the result.

Combining this analysis with the upper bound $C \leq 1$ (Remark 1.3), we deduce the following.

**Theorem 1.3.** *Using Algorithm 1.2, we can find the defective set with certainty using $k \log_2(n/k) + O(k)$ adaptive tests. Thus, the capacity of adaptive group testing in the sparse regime $k = \Theta(n^\alpha)$ is $C_0 = C = 1$ for all $\alpha \in [0, 1)$.*

This theorem follows directly from the work of Hwang [106], and it was explicitly noted that such an algorithm attains the capacity of adaptive group testing by Baldassini *et al.* [20].

The precise form of Hwang's generalized binary splitting algorithm [106] used a variant of this method, with various tweaks to reduce the $O(k)$ term. For example, the set sizes are chosen to be powers of 2 at each stage, so the splitting step in Algorithm 1.1 is always exact. Further, items appearing at any stage in a negative test are removed completely, and the values $n$ and $k$ of remaining items are updated as the algorithm progresses. Some subsequent work further reduced the implied constant in the $O(k)$ term in the expression $k \log_2(n/k) + O(k)$ above; for example, Allemann [13] reduced it to $0.255k$ plus lower order terms.

We see that algorithms based on binary splitting are very effective when the problem is sparse, with $k$ much smaller than $n$. For denser problems, the advantage may be diminished; for instance, when $k$ is a large enough fraction of $n$, it turns out that adaptive group testing offers no performance advantage over the simple strategy of individually testing every item once. For example, for adaptive zero-error combinatorial testing, Riccio and Colbourn [157] proved that no algorithm can outperform individual testing if $k \geq 0.369n$, while the Hu–Hwang–Wang conjecture [104] suggests that such a result remains true for $k \geq n/3$. We further discuss adaptive (and nonadaptive) group testing in the linear regime $k = \Theta(n)$ in Section 5.5. Meanwhile, the focus of this survey remains the sparse regime, $k = \Theta(n^\alpha)$ with $\alpha \in [0, 1)$, where group testing techniques have their greatest effect.

## 1.6 A brief review of zero-error nonadaptive group testing

In this section, we discuss nonadaptive group testing with a zero error criterion – that is, we must be certain that any defective set of a given size can be accurately decoded. In particular, we examine the important concepts of separable and disjunct matrices. The literature in this area is deep and wide-ranging, and we shall barely scratch the surface here. The papers of Kautz and Singleton [119] and D'yachkov and Rykov [65] are classic early works in this area, while the textbook of Du and Hwang [62] provides a comprehensive survey.

The following definitions for test matrices are well known – see for example [62, Chapter 7] – and are important for studying zero-error nonadaptive group testing.

**Definition 1.9.** Given a test matrix $\mathsf{X} = (x_{ti}) \in \{0,1\}^{T \times n}$, we write $S(i) := \{t : x_{ti} = 1\}$ for the support of column $i$. Further, for any subset $\mathcal{L} \subseteq \{1, 2, \ldots, n\}$ of columns, we write $S(\mathcal{L}) = \bigcup_{i \in \mathcal{L}} S(i)$ for the union of their supports. (By convention, $S(\varnothing) = \varnothing$.)

Observe that $S(i)$ is the set of tests containing item $i$, while $S(\mathcal{K})$ is the set of positive tests when the defective set is $\mathcal{K}$.

**Definition 1.10.** A matrix $\mathsf{X}$ is called *$k$-separable* if the support unions $S(\mathcal{L})$ are distinct over all subsets $\mathcal{L} \subseteq \{1, 2, \ldots, n\}$ of size $|\mathcal{L}| = k$.

A matrix $\mathsf{X}$ is called *$\bar{k}$-separable* if the support unions $S(\mathcal{L})$ are distinct over all subsets $\mathcal{L} \subseteq \{1, 2, \ldots, n\}$ of size $|\mathcal{L}| \leq k$.

Clearly, using a $k$-separable matrix as a test design ensures that group testing will provide different outcomes for each possible defective set of size $k$; thus,

provided that there are exactly $k$ defectives, it is certain that the true defective set can be found (at least in theory – we discuss what it means for an algorithm to be 'practical' in Section 2.1). In fact, it is clear that $k$-separability of the test matrix is also a necessary condition for zero-error group testing to be possible: If the matrix is not separable, then there must be two sets $\mathcal{L}_1$ and $\mathcal{L}_2$ with $S(\mathcal{L}_1) = S(\mathcal{L}_2)$ which cannot be distinguished from the test outcomes. Similarly, a $\bar{k}$-separable test design ensures finding the defective set provided that there are *at most* $k$ defectives.

Thus, given $n$ and $k$, we want to know how large $T$ must be for a $k$-separable $(T \times n)$-matrix to exist.

An important related definition is that of a disjunct matrix.

**Definition 1.11.** A matrix $\mathsf{X}$ is called *$k$-disjunct* if for any subset $\mathcal{L} \subseteq \{1, 2, \dots, n\}$ of size $|\mathcal{L}| = k$ and any $i \notin \mathcal{L}$, we never have $S(i) \subseteq S(\mathcal{L})$.

In group testing language, this ensures that no nondefective item appears only in positive tests. This not only guarantees that the defective set can be found, but also reveals how to do so easily: Any item that appears in a negative test is nondefective, while an item that appears solely in positive tests is defective. (We will study this simple algorithm under the name COMP in Chapter 2.)

We briefly mention that the notions of $k$-separability, $\bar{k}$-separability, and $k$-disjunctness often appear in the literature under different names. In particular, the columns of a $k$-disjunct matrix are often said to form a *$k$-cover free family*, and the terminology *superimposed code* is often used to refer to the columns of either a $\bar{k}$-separable matrix or a $k$-disjunct matrix (see, for example, [119, 62, 66]).

It is clear that the implications

$$k\text{-disjunct} \;\Rightarrow\; \bar{k}\text{-separable} \;\Rightarrow\; k\text{-separable} \tag{1.14}$$

hold. Furthermore, Chen and Hwang [36] showned that the number of tests $T$ required for separability and disjunctness in fact have the same order-wise scaling, proving the following.

**Theorem 1.4.** *Let $\mathsf{X}$ be $2k$-separable. Then there exists a $k$-disjunct matrix formed by adding at most one row to $\mathsf{X}$.*

Because of this, attention is often focused on bounds for disjunct matrices, since such bounds are typically easier to derive, and these results can be easily converted to statements on separable matrices using (1.14) and Theorem 1.4.

The following result, which D'yachkov and Rykov [65] attribute to Bassalygo, was an important early lower bound on the size of disjunct matrices.

**Theorem 1.5.** *Suppose there exists a $k$-disjunct $(T \times n)$-matrix. Then*

$$T \geq \min\left\{\frac{1}{2}(k+1)(k+2), n\right\}. \tag{1.15}$$

There have been many improvements to this result on bounds for disjunct matrices to exist, of which we mention a few examples. Shangguan and Ge [173] improve the constant $1/2$ in front of the $k^2$ term of Theorem 1.5 with the bound

$$T \geq \min\left\{\frac{15+\sqrt{33}}{24}(k+1)^2, n\right\} \approx \min\left\{0.864(k+1)^2, n\right\}. \tag{1.16}$$

Ruszinkó [159] proves the bound

$$T \geq \frac{1}{8}\, k^2\, \frac{\log n}{\log k} \tag{1.17}$$

for $n$ sufficiently large, provided that $k$ grows slower than $\sqrt{n}$,[1] while Füredi [82] proves a similar bound with $1/8$ improved to $1/4$. In the sparse regime $k = \Theta(n^\alpha)$ with $\alpha \in (0,1)$, we have $\log n / \log k \to 1/\alpha$, which means that (1.17) and Füredi's improvement give improved constants compared to Theorem 1.5 and (1.16) for sufficiently small $\alpha$. In the very sparse regime $k = O(1)$, (1.17) gives roughly a $\log n$ factor improvement, which D'yachkov and Rykov [65] improve further, replacing $1/8$ by a complicated expression that is approximately $1/2$ for large (but constant) values of $k$.

In the case that $k = \Theta(n^\alpha)$ with $\alpha > 1/2$, the bound $T \geq n$ of Theorem 1.5 can be achieved by the identity matrix (that is, testing each item individually), and the resulting number of tests $T = n$ is optimal.

For $\alpha < 1/2$, the $T \geq \Omega(k^2)$ lower bounds of Theorem 1.5 and related results are complemented by achievability results of the form $T \leq O(k^2 \log n)$, just a logarithmic factor larger. For example, using a Bernoulli random design with $p = 1/(k+1)$, one can prove the existence of a $k$-disjunct $(T \times n)$-matrix with

$$T \leq (1+\delta)\mathrm{e}(k+1)\ln\left((k+1)\binom{n}{k+1}\right) \sim (1+\delta)\mathrm{e}(k+1)^2 \ln n$$

for any $\delta > 0$ [62, Theorem 8.1.3]. (Du and Hwang [62, Section 8.1] attribute this result to unpublished work by Busschbach [31].) Kautz and Singleton [119] give a number of constructions of separable and disjunct matrices, notably including a construction based on Reed–Solomon codes that we discuss further in Section 5.7. Porat and Rothschild [155] give a construction with $T = O(k^2 \log n)$ using linear codes.

Note that in the sparse regime, the lower bound from Theorem 1.5 is on the order of $\min\{\Omega(k^2), n\}$ which is much larger than the order $k \log n$ of the counting bound. Thus, nonadaptive zero-error group testing has rate 0 according to Definition 1.7.

**Theorem 1.6.** *The capacity of nonadaptive group testing with the zero-error criterion is $C_0 = 0$ in the case that $k = \Theta(n^\alpha)$ with $\alpha \in (0,1)$.*

*Remark* 1.4. In the context of zero-error communication [174], a memoryless channel having a zero-error capacity of zero is a very negative result, as it implies that not even two distinct codewords can be distinguished with zero error probability. We emphasize that when it comes to group testing, the picture is very different: A result stating that $C_0 = 0$ by no means implies that attaining zero error probability is a hopeless task; rather, it simply indicates that it is insufficient to take $O\big(k \log \frac{n}{k}\big)$ tests. As discussed above, there is an extensive amount of literature establishing highly valuable results in which the number of tests is $O(k^2 \log n)$ or similar.

In contrast with Theorem 1.6, in Chapters 2 and 4 of this survey, we will see that under the small-error criterion (i.e., asymptotically vanishing but non-zero

[1] The first line of the proof in [159] assumes $k^2$ divides $n$; this is not true when $k^2 > n$, but can be accommodated with a negligible increase in $n$ if $k$ grows slower than $\sqrt{n}$.

error probability), we can achieve nonzero rates for all $\alpha \in [0, 1)$, and even reach the optimal rate of 1 for $\alpha \in [0, 0.409]$. This demonstrates the significant savings in the number of tests permitted by allowing a small nonzero error probability.

An interesting point of view is provided by Gilbert *et al.* [91], who argue that zero-error group testing can be viewed as corresponding to an adversarial model on the defective set; specifically, the adversary selects $\mathcal{K}$ as a function of $\mathsf{X}$ in order to make the decoder fail. Building on this viewpoint, [91] gives a range of models where the adversary's choice is limited by computation or other factors, effectively interpolating between the zero-error and small-error models.

## 1.7 Applications of group testing

Although group testing was first formulated in terms of testing for syphilis [61], it has been abstracted into a combinatorial and algorithmic problem, and subsequently been applied in many contexts. The early paper of Sobel and Groll [180] lists some basic applications to unit testing in industrial processes, such as the detection of faulty containers, capacitors, or Christmas tree lights. Indeed, solutions based on group testing have been proposed more recently for quality control in other manufacturing contexts, such as integrated circuits [114] and molecular electronics [183] (though the latter paper studies a scenario closer to the linear model discussed in Section 5.9).

We review some additional applications here; this list is certainly not exhaustive, and is only intended to give a flavour of the wide range of contexts in which group testing has been applied. Many of these applications motivate our focus on nonadaptive algorithms. This is because in many settings, adaptive algorithms are impractical, and it is preferable to fix the test design in advance – for example, to allow a large number of tests to be run in parallel.

### Biology

As group testing was devised with a biological application in mind, it is no surprise that it has found many more uses in this field, as summarised, for example, in [21, 37, 63]. We list some examples here:

**DNA testing** As described in [62, Chapter 9], [172] and [177], modern sequencing methods search for particular subsequences of the genome in relatively short fragments of DNA. As a result, since samples from individuals can easily be mixed, group testing can lead to significant reductions in the number of tests required to isolate individuals with rare genetic conditions – see, for example, [21, 51, 95]. In this context, it is typical to use nonadaptive methods (as in [63, 73, 72, 136, 177]), since it is preferable not to stop machines in order to rearrange the sequencing strategy. Furthermore, the physical design of modern DNA testing plates means that it can often be desirable to use exactly $T = 96$ tests (see [72]). Macula [136] describes combinatorial constructions that are robust to errors in testing.

**Counting defective items** Often we do not need to estimate the defective set itself, but rather wish to efficiently estimate the proportion of defective items. This may be because we have no need to distinguish individuals (for example,

when dealing with insects [186, 194]), or wish to preserve confidentiality of individuals (for example, monitoring prevalence of diseases). References [35, 179, 185] were early works showing that group testing offers an efficient way to estimate the proportion of defectives, particularly when defectivity is rare. This testing paradigm continues to be used in recent medical research, where pooling can provide significant reductions in the cost of DNA testing – see for example [121], [187].

Specific applications are found in works such as [117, 185, 186, 194], in which the proportion of insects carrying a disease is estimated; and in [89, 189], in which the proportion of the population with HIV/AIDS is estimated while preserving individual privacy. Many of these protocols require nonadaptive testing, since tests may be time-consuming – for example, one may need to place a group of possibly infected insects with a plant, and wait to see if the plant becomes infected. A recent paper [74] gives a detailed analysis of an adaptive algorithm that estimates the number of defectives. We review the question of counting defectives using group testing in more detail in Section 5.3.

**Other biological applications** We briefly remark that group testing has also been used in many other biological contexts – see [63, Section 1.3] for a review. For example, this includes the design of protein–protein interaction experiments [151], high-throughput drug screening [115], and efficient learning of the Immune–Defective graphs in drug design [86].

## Commmunications

Group testing has been applied in a number of communications scenarios, including the following:

**Multiple access channels** We refer to a channel where several users can communicate with a single receiver as a *multiple access channel*. Wolf [197] describes how this can be formulated in terms of group testing: At any one time, a small subset of users (active users) will have messages to transmit, and correspond to defective items in this context. Hayes [101] introduced adaptive protocols based on group testing to schedule transmissions, which were further developed by many authors (see [197] for a review). In fact, Berger *et al.* [23] argue for the consideration of a ternary group testing problem with outcomes 'idle', 'success' and 'collision' corresponding to no user, one user or multiple users broadcasting simultaneously, and develop an adaptive transmission protocol.

These adaptive group testing protocols for multiple access channels are complemented by corresponding nonadaptive protocols developed in works such as [124] (using random designs) and [59] (using designs based on superimposed code constructions). Variants of these schemes were further developed in works such as [60], [198] and [199]. The paper [190] uses a similar argument for the related problem of Code-Division Multiple Access (CDMA), where decoding can be performed for a group of users simultaneously transmitting from constellations of possible points.

**Cognitive radios** A related communication scenario is that of cognitive radio networks, where 'secondary users' can opportunistically transmit on frequency

bands which are unoccupied by primary users. We can scan combinations of several bands at the same time and detect if any signal is being transmitted across any of them, and use procedures based on group testing to determine which bands are unoccupied – see for example [16, 176].

**Network tomography and anomaly discovery** Group testing has been used to perform (loss) network tomography; that is, to detect faults in a computer network only using certain end-to-end measurements. In this scenario, users send a packet from one machine to another, and check whether it successfully arrives. For example, we can view the edges of the network as corresponding to items, with items in a test corresponding to the collection of edges along which the packet travelled. If (and only if) a packet arrives safely, we know that no edge on that route is faulty (no item is defective), which precisely corresponds to the OR operation of the standard noiseless group testing model.

As described in several works including [42, 100, 133, 200], and discussed in more detail in Section 5.8, this leads to a scenario where arbitrary choices of tests cannot be taken, since each test must correspond to a connected path in the graph topology. This motivates the study of graph-constrained group testing, which is an area of interest in its own right.

Goodrich and Hirschberg [98] describes how an adaptive algorithm for ternary group testing can be used to find faulty sensors in networks, and a nonadaptive algorithm (combining group testing with Kalman filters) is described in [131].

## Information technology

The discrete nature of the group testing problem makes it particularly useful for various problems in computing, such as the following:

**Data storage and compression** Kautz and Singleton [119] describe early applications of superimposed coding strategies to efficiently searching punch cards and properties of core memories. Hong and Ladner [102] describe an adaptive data compression algorithm for images, based on the wavelet coefficients. In particular, they show that the standard Golomb algorithm for data compression is equivalent to Hwang's group testing algorithm [106] (see Section 1.5). These ideas have been extended, for example by [103] in the context of compressing correlated data from sensor networks, using ideas related to the multiple access channel described above.

**Cybersecurity** An important cybersecurity problem is to efficiently determine which computer files have changed, based on a collection of hashes of various combinations of files (this is sometimes referred to as the 'file comparison problem'). Here the modified files correspond to defective items, with the combined hash acting as a testing pool. References [97] and [138] demonstrate methods to solve this problem using nonadaptive procedures based on group testing.

Khattab *et al.* [122] and Xuan *et al.* [201] describe how group testing can be used to detect denial-of-service attacks, by dividing the server into a number of virtual servers (each corresponding to a test), observing which ones receive large

amounts of traffic (test positive) and hence deducing which users are providing the greatest amount of traffic.

**Database systems** In order to manage databases efficiently, it can be useful to classify items as 'hot' (in high demand), corresponding to defectivity in group testing language. Cormode and Muthukrishnan [48] show that this can be achieved using both adaptive and nonadaptive group testing, even in the presence of noise. A related application is given in [195], which considers the problem of identifying 'heavy hitters' (high-traffic flows) in Internet traffic, and provides a solution using linear group testing, where each test gives the number of defective items in the testing pool (see Section 5.9).

**Bloom filters** A *Bloom filter* [25] is a data structure that allows one to test if a given item is in a special set of distinguished items extremely quickly, with no possibility of false negatives and very rare false positives.

The Bloom filter uses $L$ hash functions, each of which maps items to $\{1, 2, ..., T\}$. For each of the items in the distinguished set, one sets up the Bloom filter by hashing the item using each of the $L$ hash functions, and setting the corresponding bits in a $T$-bit array to `1`. (If the bit is already set to `1`, it is left as `1`.) To test if another item is in the distinguished set, one hashes the new item with each of the $L$ hash functions and looks up the corresponding bits in the array. If any of the bits are set to `0`, the item is not in the distinguished set; while if the bits are all set `1`, one assumes the item is in the set, although there is some chance of a false positive.

The problem of deciding how many hash functions $L$ to use, and how large the size of the array $T$ is, essentially amounts to a group testing problem. For instance, when $L$ is large enough for the outcomes to be essentially noiseless, the analysis is almost identical to that of the COMP algorithm with a near-constant tests-per-item design (see Section 2.7). We also mention that [203] makes a connection between Bloom filters and coding over an `OR` multiple-access channel, which is also closely related to group testing.

## Data science

Finally, group testing has been applied to a number of problems in statistics and theoretical computer science.

**Search problems** Du and Hwang [62, Part IV] give an extensive review of applications of group testing to a variety of search problems, including the famous problem of finding a counterfeit coin and membership problems. This can be seen as a generalization of group testing; a significant early contribution to establish order-optimal performance was made by Erdős and Rényi [71].

**Sparse inference and learning** Gilbert, Iwen and Strauss [92] discuss the relationship between group testing and compressed sensing, and show that group testing can be used in a variety of sparse inference problems, including streaming algorithms and learning sparse linear functions. Reference [140] builds on this idea by showing how group testing can be used to perform binary classification of objects, and [67] develops a framework for testing arrivals with decreasing

defectivity probability. Similar ideas can be used for classification by searching for similar items in high dimensional spaces [178].

In the work of Emad, Varshney, Malioutov and Dash [69], [141], group testing is used to learn classification rules that are interpretable by practitioners. For example, in medicine we may wish to develop a rule based on training data that can diagnose a condition or identify high-risk groups from a number of pieces of measured medical data (features). However, standard machine learning approaches such as support vector machines or neural networks can lead to classification rules that are complex, opaque and hard to interpret for a clinician. For reasons of simplicity, it can be preferable to use suboptimal classification rules based on a small collection of `AND` clauses or a small collection of `OR` clauses. In [69], [141], the authors show how such rules can be obtained using a relaxed noisy linear programming formulation of group testing (to be introduced in Section 3.2). They also use ideas based on threshold group testing (see, for example, Example 3.6 in Section 3.1) to develop a more general family of classifiers based on clinical scorecards, where a small number of integer values are added together to assess the risk.

**Theoretical computer science** Group testing has been applied to classical problems in theoretical computer science, including pattern matching [45, 109, 137] and the estimation of high degree vertices in hidden bipartite graphs [196].

In addition, generalizations of the group testing problem are studied in this community in their own right, including the '$k$-junta problem' (see for example [24, 30, 150]). A binary function $f$ is referred to as a $k$-junta if it depends on at most $k$ of its inputs, and we wish to investigate this property using a limited number of input–output pairs $(x, f(x))$.

It is worth noting that *testing* $k$-juntas only requires determining whether a given $f$ has this property or is far from having this property [24], which is distinct from *learning* $k$-juntas, i.e., either determining the $k$ inputs that $f$ depends on or estimating $f$ itself. Further studies of the $k$-junta problem vary according to whether the inputs $x$ are chosen by the tester ('membership queries') [29], uniformly at random by nature [150], or according to some quantum state [14, 19]. In this sense, group testing with a combinatorial prior is a special case of the $k$-junta learning problem, where we are sure that the function is an `OR` of the $k$ inputs.

# Appendix: Comparison of combinatorial and i.i.d. priors

In this technical appendix, we discuss the relationship between combinatorial and i.i.d. priors for the defective set. We tend to use the combinatorial prior throughout this survey, so new readers can safely skip this appendix on first reading.

Recall the two related prior distributions on the defective set $\mathcal{K}$:

- Under the *combinatorial prior*, there are exactly $k$ defective items, and the defective set $\mathcal{K}$ is uniformly random over the $\binom{n}{k}$ possible subsets of that size.

- Under the *i.i.d. prior*, each item is defective independently with a given probability $q \in (0,1)$, and hence the number of defectives $k = |\mathcal{K}|$ is distributed as $k \sim \text{Binomial}(n,q)$, with $\mathbb{E}[k] = nq$. For brevity, we adopt the notation $\overline{k} = nq$.

Intuitively, when the (average) number of defectives is large, one should expect the combinatorial prior with parameter $k$ to behave similarly to the i.i.d. prior with a matching choice of $\overline{k}$, since in the latter case we have $k = \overline{k}(1+o(1))$ with high probability, due to standard binomial concentration bounds.

To formalize this intuition, first consider the definition rate $:= \frac{1}{T}\log_2 \binom{n}{k}$ for the combinatorial prior (see Definition 1.7), along with the following analogous definition for the i.i.d. prior:

$$\text{rate} := \frac{nh(q)}{T}, \tag{1.18}$$

where $h(q) = -q\log_2 q - (1-q)\log_2(1-q)$ is the binary entropy function. Using standard estimates of the binomial coefficient [15, Sec. 4.7], the former is asymptotically equivalent to $\frac{1}{T}nh(k/n)$, which matches $\frac{1}{T}nh(q) = \frac{1}{T}nh(\overline{k}/n)$ with $k$ in place of $\overline{k}$. Consistent with our focus in this monograph, in this section we focus on scaling laws of the form $k \to \infty$ and $k = o(n)$ (or similarly with $\overline{k}$ in place of $k$), in which case the preceding rates are asymptotically equivalent to $\frac{1}{T}k\log_2\frac{n}{k}$ and $\frac{1}{T}\overline{k}\log_2\frac{n}{\overline{k}}$. With some minor modifications, the arguments that we present below also permit extensions to the linear regime $k = \Theta(n)$ (discussed in Section 5.9).

Having established that the rate expressions asymptotically coincide, further arguments are needed to transfer achievability and converse results from one prior to the other. In the following, we present two results for this purpose. In both results, any statement on the existence or non-existence of a decoding rule may refer to decoders that have perfect knowledge of the number of defectives $k$, or only partial knowledge (for example high-probability bounds), or no knowledge at all – but the assumed decoder knowledge must remain consistent throughout the entire theorem. Note that having exact knowledge of $k$ in the i.i.d. setting is a particularly unrealistic assumption, since in that setting it is a random quantity. We further discuss the issue of known vs. unknown $k$ at the end of the section.

The following theorem describes how to transfer achievability bounds from the combinatorial prior to the i.i.d. prior.

**Theorem 1.7.** *Consider a sequence of (possibly randomized or adaptive) test designs* $\mathsf{X}$ *(indexed by* $n$*) attaining* $\mathbb{P}(\text{err}) \to 0$ *under the combinatorial prior whenever* $k = k_0(1+o(1))$ *for some nominal number of defectives* $k_0$*, with* $k_0 \to \infty$ *and* $k_0 = o(n)$ *as* $n \to \infty$*. Then the same* $\mathsf{X}$ *and decoding rule also attains* $\mathbb{P}(\text{err}) \to 0$ *under the i.i.d. prior with* $q = k_0/n$ *(i.e.,* $\overline{k} = k_0$*). In particular, if a given rate* $R_0$ *is achievable under the combinatorial prior whenever* $k = k_0(1+o(1))$*, then it is also achievable under the i.i.d. prior with* $\overline{k} = k_0$*.*

*Proof.* Since $\overline{k} = k_0$ grows unbounded as $n \to \infty$ by assumption, we have by standard binomial concentration that $k = \overline{k}(1+o(1)) = k_0(1+o(1))$ with probability approaching one under the i.i.d. prior. Letting $\mathcal{I}$ denote the corresponding

set of 'typical' $k$ values, we deduce that

$$\mathbb{P}(\text{err}) = \sum_k \mathbb{P}(k)\mathbb{P}(\text{err} \mid k) \tag{1.19}$$

$$\leq \sum_{k \in \mathcal{I}} \mathbb{P}(k)\mathbb{P}(\text{err} \mid k) + \mathbb{P}(k \notin \mathcal{I}) \tag{1.20}$$

$$\to 0, \tag{1.21}$$

since $\mathbb{P}(\text{err} \mid k) \to 0$ for all $k \in \mathcal{I}$ by assumption, and we established that $\mathbb{P}(k \notin \mathcal{I}) \to 0$. This establishes the first claim. The additional claim on the rate follows since, as discussed above (and using $k = k_0(1 + o(1))$), the achievable rates for both priors are asymptotically equivalent to $\frac{1}{T} k_0 \log_2 \frac{n}{k_0}$. □

Analogously, the following theorem describes how to transfer converse bounds from the combinatorial prior to the i.i.d. prior.

**Theorem 1.8.** *Fix $R_0 > 0$, and suppose that under the combinatorial prior with some sequence of defective set sizes $k = k_0$ (indexed by $n$) satisfying $k_0 \to \infty$ and $k_0 = o(n)$, there does not exist any algorithm achieving rate $R_0$. Then for any arbitrarily small constant $\epsilon > 0$, under the the i.i.d. prior with $q = k_0(1 + \epsilon)/n$, there does not exist any algorithm achieving rate $R_0(1 + \epsilon)$.*

*Proof.* Since $k_0 \to \infty$ and the average number of defectives under the i.i.d. prior is $\overline{k} = k_0(1 + \epsilon)$, we deduce via binomial concentration that $k \in [k_0, k_0(1 + 2\epsilon)]$ with probability approaching one. For $k$ outside this range, any contribution to the overall error probability is asymptotically negligible.

On the other hand, when $k$ does fall in this range, we can consider a genie argument in which a uniformly random subset of $k_1 = k - k_0$ defectives is revealed to the decoder. The decoder is then left to identify $k_0$ defectives out of $n_1 = n - k_1$ items. Hence, the problem is reduced to the combinatorial prior with slightly fewer items.

The condition $k \leq k_0(1+2\epsilon)$ implies that $k_1 \leq 2\epsilon k_0$ and hence $n_1 \geq n - 2\epsilon k_0$, which behaves as $n(1 - o(1))$ since $k_0 = o(n)$. As discussed at the start of this section, in the sparse regime, the asymptotic rate is equal to the asymptotic value of $\frac{1}{T} k \log_2 \frac{n}{k}$ (combinatorial prior) or $\frac{1}{T} \overline{k} \log_2 \frac{n}{\overline{k}}$ (i.i.d. prior), and we observe that (i) replacing $n$ by $n(1 - o(1))$ does not impact the asymptotic rate; and (ii) replacing $\overline{k} = k_0(1 + \epsilon)$ by $k_0$ reduces the rate by a factor of $1/(1 + \epsilon)$ asymptotically.

We conclude the proof via a contradiction argument: If $R_0(1 + \epsilon)$ were achievable with $q = k_0(1 + \epsilon)/n$ under the i.i.d. prior, the preceding reduction would imply that $R_0$ is achievable with $k = k_0$ under the combinatorial prior, which was assumed to be impossible. □

Unlike Theorem 1.7, this result requires scaling the (average) number of defectives and the rate by $1 + \epsilon$. However, since $\epsilon$ is arbitrarily small, one can think of this scaling as being negligible. In fact, for all achievability and converse results that we are aware of, in the case that $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, the asymptotic rate depends only on $\alpha$ and not on the implied constant in the $\Theta(\cdot)$ notation.

**From i.i.d. to combinatorial** It is also of interest to transfer results in the opposite direction, i.e., to infer achievability or converse bounds for the combinatorial prior based on those for the i.i.d. prior. For this purpose, the contrapositive statements of Theorems 1.7 and 1.8 read as follows:

- (Theorem 1.7) If there does not exist any test design and decoder achieving $\mathbb{P}(\text{err}) \to 0$ under the i.i.d. prior when $q = k_0/n$ with $k_0 \to \infty$ and $k_0 = o(n)$, then there also does not exist any test design and decoder that simultaneously achieves $\mathbb{P}(\text{err}) \to 0$ under the combinatorial prior for all $k$ such that $k = k_0(1 + o(1))$.
- (Theorem 1.8) Again assuming $k_0 \to \infty$ and $k_0 = o(n)$, if the rate $R_0$ is achievable with $q = k_0/n$ under the i.i.d. prior, then for arbitrarily small $\epsilon > 0$ the rate $R_0/(1 + \epsilon)$ is achievable with $k = k_0/(1 + \epsilon)$ under the combinatorial prior.

It is worth noting that the former of these statements does not directly provide a converse result for any particular value of $k$, but rather, only does so for the case that several $k$ must be handled simultaneously.

**Discussion on knowledge of $k$** A somewhat more elusive challenge is to transfer results from the case that the decoder knows $k$ to the case that it does not (or the case that it only knows bounds on $k$), and vice versa. In particular, it is sometimes convenient to prove achievability results for the case that $k$ is known exactly (see for example Chapter 4), and to prove converse results for the case that $k$ is unknown (see for example Section 2.2). This potentially poses a 'gap' in the achievability and converse bounds even when the associated rates coincide.

While we are not aware of any general results allowing one to close such a gap, we briefly mention a technique that has succeeded in doing so in the noiseless setting; the details will be given in Section 4.3. Recall that we gave a lower bound on the rate of SSS for unknown $k$ in Section 2.2. To establish a converse for the case that $k$ is known, we argue that if both SSS and COMP fail, then there must exist some $\mathcal{K}' \neq \mathcal{K}$ with $|\mathcal{K}'| = |\mathcal{K}| = k$ such that $\mathcal{K}'$ is also consistent with the outcomes. (Note that neither COMP nor SSS require knowledge of $k$). Since the decoder cannot do better than randomly guess between these two consistent sets even when $k$ is known, we deduce that $\mathbb{P}(\text{err})$ cannot tend to zero.

Transferring converse results from the unknown-$k$ setting to the known-$k$ setting – or equivalently, transferring achievability results in the other direction – in greater generality (such as noisy settings) remains an interesting open problem. General results of this kind would reduce the need to study these distinct cases separately.

# Chapter 2

# Algorithms for Noiseless Group Testing

## 2.1 Summary of algorithms

In this chapter, we discuss decoding algorithms for noiseless nonadaptive group testing. That is, we are interested in methods for forming the estimate $\hat{\mathcal{K}}$ of the defective set given the test matrix $\mathsf{X}$ and the outcomes $\mathbf{y}$. In addition, we present performance bounds for these algorithms under random test designs.

We are particularly interested in algorithms that are practical in the following two senses:

1. The algorithm does not require knowledge of the number of defectives $k$, other than general imprecise knowledge such as 'defectivity is rare compared to nondefectivity', which we implicitly assume throughout.

2. The algorithm is computationally feasible. Specifically, we seek algorithms that run in time and space polynomial in $n$, and preferably no worse than the $O(nT)$ time and space needed to read and store an arbitrary test design $\mathsf{X}$. In fact, certain algorithms exist having even faster 'sublinear' decoding times, but their discussion is deferred to Section 5.4.

All algorithms in this chapter are practical in the first sense. One of the algorithms, called SSS, is not (likely to be) practical in the second sense, but we include it here as a benchmark. Since several algorithms are based on approximating the SSS algorithm in faster time, the analysis of SSS is useful for understanding these other more practical algorithms.

An important concept that we will use is that of a satisfying set.

**Definition 2.1.** Consider the noiseless group testing problem with $n$ items, using a test design $\mathsf{X}$ and producing outcomes $\mathbf{y}$. A set $\mathcal{L} \subset \{1, 2, \ldots, n\}$ is called a *satisfying set* if

- every positive test contains at least one item from $\mathcal{L}$;

- no negative test contains any item from $\mathcal{L}$.

Thus, when performing group testing with test design $\mathsf{X}$, had the true defective set been $\mathcal{L}$, we would have observed the outcome $\mathbf{y}$. Clearly the true defective set $\mathcal{K}$ is itself a satisfying set.

While we postpone complete definitions of algorithms until the relevant sections, it is worth providing a quick outline here:

**SSS (smallest satisfying set)** chooses the smallest satisfying set. This is based on the idea that the defective set $\mathcal{K}$ is a satisfying set, but since defectivity is rare, $\mathcal{K}$ is likely to be small. This algorithm appears unlikely to be implementable with a practical runtime in general for large $n$, as it is equivalent to solving an integer program. See Sections 2.2 and 2.6 for details.

**COMP (combinatorial orthogonal matching pursuit)** assumes that each item is defective unless there is a simple proof that it is nondefective, namely, that the item is in at least one negative test. See Section 2.3 for details.

**DD (definite defectives)** assumes that each item is nondefective unless there is a certain simple proof that it is defective. See Section 2.4 for details.

**SCOMP (sequential COMP)** attempts to find the smallest satisfying set by beginning with the DD set of definite defectives, and sequentially adding items to the estimated defective set until a satisfying set is obtained. See Section 2.5 for details.

**Linear programming relaxations** solve a relaxed form of the smallest satisfying set problem, and use this to attempt to find the smallest satisfying set. See Section 2.6 for details.

In Chapter 3, we will present additional algorithms that can be specialized to the noiseless case, particularly *belief propagation* (see Section 3.3) and *separate decoding of items* (Section 3.5). Since these algorithms are primarily of interest for noisy settings, we omit them from this chapter to avoid repetition.

We will shortly see numerical evidence that SSS performs the best when the problem is sufficiently small that it can be implemented, while the SCOMP and linear programming approaches perform best among the more practical algorithms. As another means to compare these algorithms, in the upcoming sections, we will mathematically analyse the rates achieved by these algorithms in the case that the matrix $\mathsf{X}$ has a Bernoulli design, formally stated as follows.

**Definition 2.2.** In a Bernoulli design, each item is included in each test independently at random with some fixed probability $p = \nu/k$. In other words, independently over $i \in \{1, 2, \dots, n\}$ and $t \in \{1, 2, \dots, T\}$, we have $\mathbb{P}(X_{ti} = \mathsf{1}) = p = \nu/k$ and $\mathbb{P}(X_{ti} = \mathsf{0}) = 1 - p = 1 - \nu/k$.

The parametrization $p = \nu/k$ is chosen because such scaling with constant $\nu$ will be seen to be optimal as $k$ grows large. Intuitively, since the average number of defectives in each test is $\nu$, one should avoid the cases $\nu \to 0$ or $\nu \to \infty$ because they lead to uninformative tests (that is, tests returning a given outcome with high probability). In Section 2.7, we turn to the near-constant tests-per-item design, formally introduced in Definition 2.3 below. We

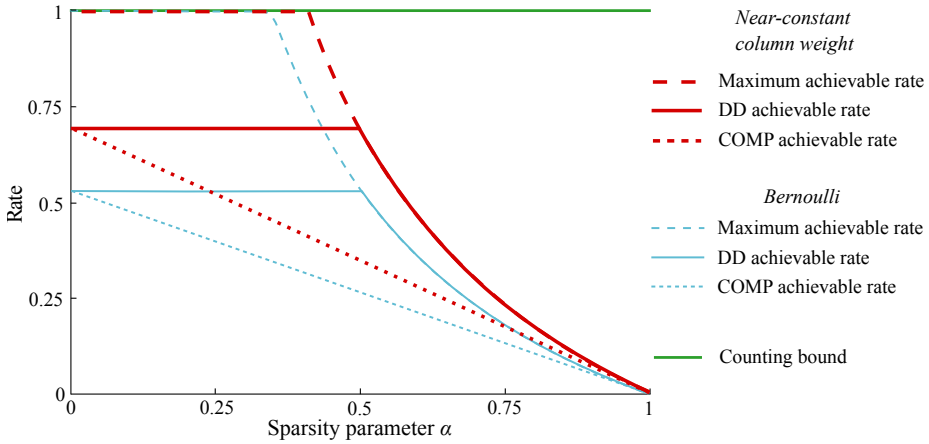



Figure 2.1: Rates of various algorithms for nonadaptive group testing in the sparse regime with a Bernoulli design and with a near-constant column weight design.

will see that the rates of all of the above algorithms are slightly improved when used with this alternative design.

Later, in Section 4.3, we will see how the analysis of the SSS and COMP algorithms allows us to establish algorithm-independent lower bounds on the number of tests required under Bernoulli and near-constant test-per-item designs.

The result of this chapter are further complemented by those of Chapter 4, which looks further into information-theoretic achievability and converse bounds. A collective highlight of these two chapters is the following result on the information-theoretic optimality of some practical algorithms when certain random test designs are adopted.

**Theorem 2.1.** *Consider group testing with $n$ items and $k = \Theta(n^\alpha)$ defectives, for $\alpha > 1/2$. Suppose that the test design is Bernoulli (with the parameter $p$ chosen optimally) or constant tests-per-item (with the parameter $L$ chosen optimally), and suppose that the decoding algorithm is SSS, DD, SCOMP, or linear programming. Then we can achieve the rate*

$$R^* = \begin{cases} \dfrac{1}{\mathrm{e}\ln 2}\dfrac{1-\alpha}{\alpha} \approx 0.531\dfrac{1-\alpha}{\alpha} & \text{for Bernoulli design,} \\ \ln 2\,\dfrac{1-\alpha}{\alpha} \approx 0.693\dfrac{1-\alpha}{\alpha} & \text{for near-constant tests-per-item,} \end{cases}$$

*which is the optimal rate for these random test designs regardless of the decoding algorithm used.*

This and the other main results from this chapter are illustrated in Figure 2.1.

Table 2.1 summarizes the algorithms. Whether or not the linear programming approach is guaranteed to give a satisfying set depends on the rule used to convert the LP solution to an estimate of the defective set – see Section 2.6 for further discussion.

| | Optimal rate | Fast | SS | No false + | No false − |
|---|---|---|---|---|---|
| **SSS** | $\alpha \in [0, 1)$ | no | yes | no | no |
| **COMP** | no | yes | no | no | yes |
| **DD** | $\alpha \in [1/2, 1)$ | yes | no | yes | no |
| **SCOMP** | $\alpha \in [1/2, 1)$ | yes | yes | no | no |
| **LP** | $\alpha \in [1/2, 1)$ | yes | maybe | no | no |

Table 2.1: Summary of features of algorithms: (i) range of $\alpha$ (if any) for which the optimal rate is attained under randomized testing; (ii) whether an efficient algorithm is known; (iii) whether the output is guaranteed to be a satisfying set; (iv)-(v) guarantees on the false positives and false negatives in the reconstruction. The algorithms labelled 'fast' can be implemented in time $O(nT)$, except possibly LP, whose complexity depends on the solver used (see Section 2.6 for discussion).

While this chapter focuses on rigorously provable results, we can also examine the behaviour of the algorithms presented here in simulations. Figure 2.2 compares the five algorithms with a Bernoulli design. We see that SCOMP and linear programming are almost as good as SSS (which would be infeasible for larger problems), while DD is not much worse, and COMP lags behind more significantly.

Figure 2.3 compares the performance of the COMP, DD and SSS algorithms under Bernoulli and near-constant column weight matrix designs. We see that the near-constant column weight design provides a noticeable improvement for all three algorithms.

## 2.2 SSS: Smallest satisfying set

We first discuss the smallest satisfying set (SSS) algorithm. While SSS does not require knowledge of the number of defectives, it appears unlikely to be impractical for a large number of items $n$ due to the computation required. This is due to the highly combinatorial nature of the algorithm, which amounts to solving an integer program (see Section 2.6).

Recall from Definition 2.1 that a set $\mathcal{L}$ is satisfying if every positive test contains at least one item from $\mathcal{L}$, and no negative test contains any item from $\mathcal{L}$. Moreover, since the true defective set $K$ is satisfying, a smallest satisfying set definitely exists – though it may not be unique.

The SSS algorithm is based on the idea that since $K$ is satisfying, and since defectivity is rare, it seems plausible that $K$ should be the smallest satisfying set.

**Algorithm 2.1.** The *smallest satisfying set* (*SSS*) algorithm is defined by setting $\hat{K}_{\mathrm{SSS}}$ to be the smallest satisfying set, with ties broken arbitrarily if such a set is not unique.

**Example 2.1.** In Figure 2.4, we give an example of a group testing matrix and its outcome. (This example will reappear throughout this chapter.) Since we

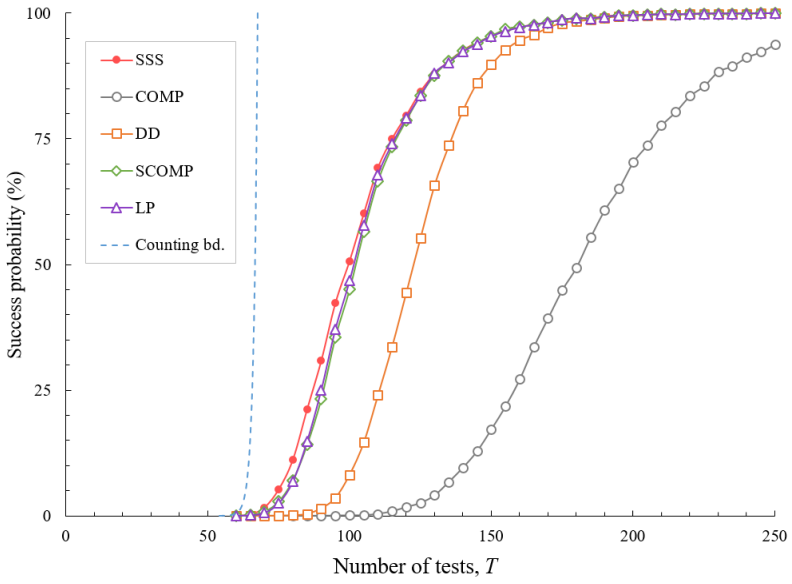



Figure 2.2: Empirical performance through simulation of five algorithms with a Bernoulli design. The parameters are $n = 500$, $k = 10$, and $p = 1/(k+1) = 0.0909$. For comparison purposes, we plot the theoretical upper bound on $\mathbb{P}(\text{suc})$ from Theorem 1.1 as 'counting bd'.

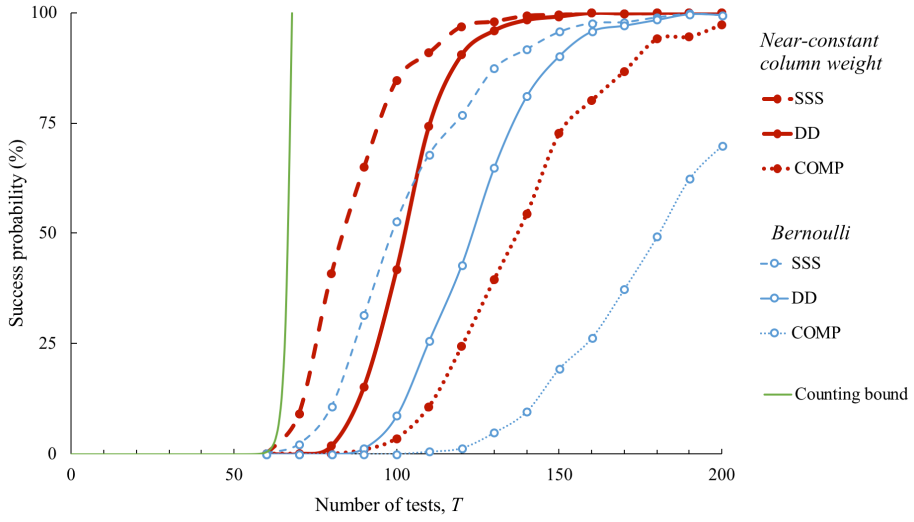



Figure 2.3: Simulation of the COMP, DD and SSS algorithms with Bernoulli and near-constant column weight designs. The problem parameters are $n = 500$, and $k = 10$. The Bernoulli parameter is $p = 1/k = 0.1$; the near-constant column weight parameter $L$ is the nearest integer to $(\ln 2)T/k \simeq 0.0693T$.

| ? | ? | ? | ? | ? | ? | ? | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Figure 2.4: Group testing example. We describe the outcome of the SSS algorithm in Example 2.1, and of the COMP algorithm in Example 2.2.

have only $n = 7$ items, it is, in this small case, practical to check all $2^7 = 128$ subsets. It is not difficult to check that the sets $\{2, 4\}$ and $\{2, 4, 7\}$ are the only satisfying sets. Of these, $\{2, 4\}$ is the smallest satisfying set. Thus, we have $\hat{\mathcal{K}}_{\text{SSS}} = \{2, 4\}$ as the output of the SSS algorithm.

*Remark* 2.1. A naive implementation of SSS requires an exhaustive search over $\binom{n}{k}$ putative defective sets, and in general, we do not regard SSS as being practical in the sense described above. To make this intuition more precise, we describe a connection to the *set cover* problem (see [191]). Given a universe $U$ and a family $\mathcal{S}$ of subsets of $U$, the set cover problem is to find the smallest family of subsets in $\mathcal{S}$ such that its union is the whole of $U$. Suppose that we let $U$ be the set of positive tests, and the subsets in $\mathcal{S}$ list the tests in which each possibly defective item is included. (An item is 'possibly defective' if it appears in no negative tests; this definition will play a key role in the COMP and DD algorithms later.) Then the minimal set cover is exactly the smallest satisfying set. The set cover problem is known to be NP-hard to solve [116], or even to verify that a given putative solution is optimal.

Since SSS is in some sense the 'best possible' (albeit possibly impractical) algorithm, we are interested in upper bounds on its rate. In Section 4.3, we make the term 'best possible' more precise, and show that in fact these upper bounds are achievable in the information-theoretic sense. In fact, if we switch from the combinatorial prior to the i.i.d. prior on the defective set (see the Appendix to Chapter 1), then it can be shown that SSS is equivalent to *maximum a posteriori* (MAP) estimation of $\mathcal{K}$, and in this case its optimality is immediate.

**Theorem 2.2.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, and using the SSS algorithm for decoding. The maximum rate achievable by SSS with a Bernoulli design is bounded above by*

$$\overline{R}_{\text{Bern}}^{\text{SSS}} \leq \max_{\nu > 0} \min \left\{ h(\mathrm{e}^{-\nu}),\ \frac{\nu}{\mathrm{e}^{\nu} \ln 2} \frac{1-\alpha}{\alpha} \right\}. \tag{2.1}$$

*Here and throughout, we write $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ for the binary entropy function, measured in bits.*

*Proof.* The first term follows using an argument from [7], and can be seen as a strengthening of the Fano-based counting bound (1.6) provided by Chan *et al.*

[33]. The idea is to observe that [33] considers tests for which the probability of being negative could take any value, so the entropy is upper bounded by the maximum possible value, $H(Y_t) \leq h(1/2) = 1$. However, under Bernoulli testing with a general value of $p = \nu/k$, the probability that a test is negative is $(1-\nu/k)^k \simeq \mathrm{e}^{-\nu}$, meaning that $H(Y_t) \simeq h(\mathrm{e}^{-\nu})$. The full argument, which is justified by a typical set argument, is given in [7, Lemma 1].

We explain how the second term follows using a simplified version of the argument from [12]. We say that a defective item is *masked* if every time it is tested, it appears in a test with some other defective item. We observe that if some $i \in \mathcal{K}$ is masked then SSS fails, since the set $\mathcal{K} \setminus \{i\}$ forms a smaller satisfying set. In other words, we know that

$$\mathbb{P}(\mathrm{err}) \geq \mathbb{P}\left(\bigcup_{i \in \mathcal{K}} \{i \text{ masked}\}\right). \tag{2.2}$$

Hence, using the Bonferroni inequalities (see for example [77, Chapter IV, eq. (5.6)]), we can bound

$$\begin{aligned}\mathbb{P}(\mathrm{suc}) &\leq 1 - \mathbb{P}\left(\bigcup_{i \in \mathcal{K}} \{i \text{ masked}\}\right) \\ &\leq 1 - \sum_{i \in \mathcal{K}} \mathbb{P}\left(\{i \text{ masked}\}\right) + \sum_{i<j \in \mathcal{K}} \mathbb{P}\left(\{i \text{ and } j \text{ masked}\}\right). \end{aligned} \tag{2.3}$$

Now, any particular defective $i$ is masked if all the tests it is included in also contain one or more other defective items. Using a Bernoulli test design with item probability $p = \nu/k$, the probability that item $i$ appears in a particular test with no other defectives is $p(1-p)^{k-1}$, so the probability it is masked is $(1-p(1-p)^{k-1})^T$. Similarly, the probability that items $i$ and $j$ are both masked is $(1-2p(1-p)^{k-1})^T$, since we need to avoid the two events 'item $i$ and no other defective tested' and 'item $j$ and no other defective tested', which are disjoint.

Overall, then, in (2.3) we can deduce that

$$\mathbb{P}(\mathrm{suc}) \leq 1 - k(1-p(1-p)^{k-1})^T + \frac{k^2}{2}(1-2p(1-p)^{k-1})^T. \tag{2.4}$$

Since $p = \nu/k$, we write $r = p(1-p)^{k-1} \sim \nu/(k\mathrm{e}^{\nu})$. Taking

$$T = \left\lceil \frac{(1-r)\ln k}{r} \right\rceil \tag{2.5}$$

and using the fact that

$$k(1-r)^T - \frac{k^2}{2}(1-2r)^T \geq \left(k\mathrm{e}^{-rT/(1-r)}\right)\left(1 - \frac{k}{2}\mathrm{e}^{-rT/(1-r)}\right) \tag{2.6}$$

(see [12, eq. (41)]), we can deduce that (2.4) is bounded away from 1. Note that proving a converse for the choice (2.5) also proves the same converse for all larger values of $T$, since additional tests can only help the SSS algorithm.

In Definition 1.8, using the binomial coefficient approximation (1.11), this means that if the rate exceeds

$$\frac{\log_2 \binom{n}{k}}{T} \sim \frac{(1-\alpha)}{\ln 2}\frac{k\ln n}{T} = \frac{(1-\alpha)k\ln n}{\ln 2}\frac{r}{(1-r)\ln k} \sim \frac{\nu}{e^{\nu}\ln 2}\frac{1-\alpha}{\alpha}, \tag{2.7}$$

then the success probability is bounded away from 1. □

In Chapter 4, we will survey a result of Scarlett and Cevher [165] showing that the rate (2.1) is achievable for all $\alpha \in (0, 1)$, and deduce that this is the maximum achievable rate for the Bernoulli design.

## 2.3 COMP: Combinatorial orthogonal matching pursuit

The COMP algorithm was the first practical group testing algorithm shown to provably achieve a nonzero rate for all $\alpha < 1$. The proof was given by Chan *et al.* [33, 34].

COMP is based on the simple observation that any item in a negative test is definitely nondefective. COMP makes the assumption that the other items are defective.

**Algorithm 2.2.** The *COMP* algorithm is defined as follows. We call any item in a negative test *definitely nondefective* (*DND*), and call the remaining items *possibly defective* (*PD*). Then the COMP algorithm outputs $\hat{\mathcal{K}}_{\text{COMP}}$ equalling the set of possible defectives.

The basic idea behind the COMP algorithm has appeared many times under many names – the papers [119, 144, 33, 34, 37, 132] are just a few examples. The first appearance of the COMP idea that we are aware of is by Kautz and Singleton [119]. We use the name 'COMP' (for Combinatorial Orthogonal Matching Pursuit) following Chan *et al.* [33].

**Example 2.2.** Recall our worked example from Figure 2.4, previously discussed in Example 2.1. Consider all the negative tests. Test 1 is negative, so items 1, 3, and 6 are definitely nondefective (DND). Test 3 is negative, so items 1 and 5 are definitely nondefective. Putting this together, we deduce that the remaining items 2, 4, 7 are the possible defective (PD) items, so we choose to mark them as defective. In other words, $\hat{\mathcal{K}}_{\text{COMP}} = \{2, 4, 7\}$.

The following lemma shows that COMP can be interpreted as finding the largest satisfying set, in stark contrast with SSS.

**Lemma 2.1.** *The estimate $\hat{\mathcal{K}}_{\text{COMP}}$ generated by the COMP algorithm is a satisfying set (in the sense of Definition 2.1) and contains no false negatives. Every satisfying set is a subset of $\hat{\mathcal{K}}_{\text{COMP}}$, so $\hat{\mathcal{K}}_{\text{COMP}}$ is the unique largest satisfying set.*

*Proof.* Observe that since every DND item appears in a negative test, and so must indeed be nondefective, the COMP algorithm outputs no false defectives, and the true defective set satisfies $\mathcal{K} \subseteq \hat{\mathcal{K}}_{\text{COMP}}$. Furthermore, since every positive test contains an element of $\mathcal{K}$, and hence of $\hat{\mathcal{K}}_{\text{COMP}}$, we deduce that $\hat{\mathcal{K}}_{\text{COMP}}$ is a satisfying set.

Fix a satisfying set $\mathcal{L}$ and consider item $i \notin \hat{\mathcal{K}}_{\text{COMP}}$. By construction $i$ must be a DND, meaning that it appears in a negative test and therefore (see the second part of Definition 2.1) cannot be in $\mathcal{L}$. In other words $\hat{\mathcal{K}}^{\mathsf{c}}_{\text{COMP}} \subseteq \mathcal{L}^{\mathsf{c}}$, or reversing the inclusion, $\mathcal{L} \subseteq \hat{\mathcal{K}}_{\text{COMP}}$. □

(In the above and subsequently, we are writing $\mathcal{K}^{\mathsf{c}} = \{1, \dots, n\} \setminus \mathcal{K}$ for the items not in $\mathcal{K}$.)

The observation that any satisfying set must be a subset of $\hat{\mathcal{K}}_{\mathrm{COMP}}$ can be used to reduce the search space associated with SSS, since the number of subsets of $\hat{\mathcal{K}}_{\mathrm{COMP}}$ is typically much smaller than the number of subsets of $\{1, 2, \dots, n\}$. However, this may not make SSS sufficiently practical in general; Remark 2.5 below shows that in regimes where DD fails, the expected size of $\hat{\mathcal{K}}_{\mathrm{COMP}}$ is at least $kn^{\epsilon} \gg k$ for some $\epsilon > 0$, so the remaining number of possibilities is still $\binom{kn^{\epsilon}}{k} \geq n^{\epsilon k}$.

In the remainder of this section, we will give the rate achievable by the COMP algorithm with a Bernoulli design. These results are due to [33] and [7]. We should expect that COMP is suboptimal, since it does not make use of the positive tests.

**Theorem 2.3.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^{\alpha})$ for some $\alpha \in (0, 1)$, and using the COMP algorithm for decoding. With a Bernoulli design and an optimized parameter $p = 1/k$, the following rate is achievable:*

$$R_{\mathrm{Bern}}^{\mathrm{COMP}} = \frac{1}{\mathrm{e} \ln 2}(1 - \alpha) \approx 0.531(1 - \alpha). \tag{2.8}$$

*Proof.* For any given nondefective item, the probability that it appears in a particular test, and that such a test is negative, is $p(1-p)^k$. This follows from the independence assumption in the Bernoulli design; the test is negative with probability $(1-p)^k$, and the given item appears with probability $p$. Hence, the probability that this given nondefective appears in no negative tests is $(1-p(1-p)^k)^T$.

The COMP algorithm succeeds precisely when every nondefective item appears in a negative test, so the union bound gives

$$\begin{aligned} \mathbb{P}(\mathrm{err}) &= \mathbb{P}\left( \bigcup_{i \in \mathcal{K}^{\mathsf{c}}} \{\text{item } i \text{ does not appear in a negative test}\} \right) \\ &\leq |\mathcal{K}^{\mathsf{c}}| \left(1 - p(1-p)^k\right)^T \\ &\leq n \exp(-Tp(1-p)^k). \end{aligned} \tag{2.9}$$

The expression $p(1-p)^k$ is maximized at $p = 1/(k+1) \sim 1/k$, so we take $p = 1/k$ (or equivalently $\nu = 1$), meaning that $(1 - 1/k)^k \to \mathrm{e}^{-1}$. Hence, taking $T = (1+\delta)\mathrm{e}k \ln n$ means that $Tp(1-p)^k \sim (1+\delta)\ln n$.

Using the binomial coefficient approximation (1.11), we have the asymptotic expression

$$\frac{\log_2 \binom{n}{k}}{T} \sim \frac{(1-\alpha)}{\ln 2} \frac{k \ln n}{T}.$$

Then following Definition 1.8, taking $(1+\delta)\mathrm{e}k \ln n$ gives a rate arbitrarily close to $(1-\alpha)/(\mathrm{e} \ln 2)$, as desired. □

*Remark* 2.2. Using a similar but slightly more involved argument, one can show that the expression $R_{\mathrm{Bern}}^{\mathrm{COMP}}$ from (2.8) gives the maximum achievable rate when COMP is used in conjunction with Bernoulli testing. The argument is outlined as follows for a general parameter $p = \nu/k$.

First, we use binomial concentration to show that the number of negative tests $T_0$ is tightly concentrated around its mean, yielding $T_0 \simeq T\mathrm{e}^{-\nu}$. Conditioned on $T_0$, the probability of a given nondefective failing to be in any negative test is

$$q := (1-p)^{T_0} \simeq \exp(-pT_0) \simeq \exp\left(-\frac{\nu}{k}Te^{-\nu}\right).$$

The error events are (conditionally) independent for different nondefective items, so the total error probability is $1-(1-q)^{n-k}$. Substituting $q \simeq \exp(-T\nu e^{-\nu}/k)$, applying some simple manipulations, and noticing that $\nu\mathrm{e}^{-\nu}$ is maximized at $\nu = 1$, we find that

$$\begin{aligned} T > (1+\eta)\mathrm{e}k\ln n \quad &\implies \quad \mathbb{P}(\mathrm{err}) \to 0 \\ T < (1-\eta)\mathrm{e}k\ln n \quad &\implies \quad \mathbb{P}(\mathrm{err}) \to 1, \end{aligned}$$

for arbitrarily small $\eta > 0$. This matches the choice of $T$ in the proof of Theorem 2.3 above. In fact, this argument not only shows that (2.8) the highest achievable rate, but also that the error probability tends to one when this rate is exceeded.

The preceding argument essentially views COMP as a coupon-collecting algorithm, gradually building up a list of nondefectives using negative tests. We say that a nondefective item is 'collected' if it appears in a negative test. A result dating back to Laplace (see also [70]) states that in order to collect all coupons in a set of size $m$, it suffices to have $m\ln m$ trials. Here, we need to collect $m = n-k \sim n$ coupons, and each of the $\mathrm{e}^{-1}T$ negative tests (assuming $\nu = 1$) contains on average $pn = n/k$ such coupons. Thus, we require $\frac{n}{k}\mathrm{e}^{-1}T \sim n\ln n$, which rearranges to $T \sim \mathrm{e}k\ln n$.

*Remark* 2.3. The analysis above allows us to consider the extent to which COMP and other algorithms require us to know the exact number of defectives $k$. While, for a given test matrix, the decisions taken by COMP do not require this knowledge, clearly to form a Bernoulli test matrix with $p = 1/k$ requires the value of $k$ itself.

However, COMP is reasonably robust to misspecification of $k$, in the following sense. Suppose that for a fixed $c$ we base our matrix on the erroneous assumption that there are $\widehat{k} = k/c$ defectives, and hence use $p = 1/\widehat{k} = c/k$ instead. Repeating the analysis above, we can see that (2.9) behaves like $n\exp(-Tc\mathrm{e}^{-c}/k)$, so we should use $T = (1+\delta)k\ln n/(c\mathrm{e}^{-c})$ tests, corresponding to a rate of $(1-\alpha)c\mathrm{e}^{-c}/\ln 2$.

In other words, even with a multiplicative error in our estimate of $k$, COMP will achieve a nonzero rate, and if the multiplicative factor $c$ is close to 1, COMP will achieve a rate close to that given in Theorem 2.3 above. Although the analysis would be more involved, we anticipate that similar results should hold for other algorithms. We further discuss the question of uncertainty in the number of defectives, including how group testing can provide estimates of $k$ which are accurate up to a multiplicative factor, in Section 5.3.

## 2.4 DD: Definite defectives

The DD (Definite Defectives) algorithm, due to Aldridge, Baldassini and Johnson [12], was the first practical group testing algorithm to provably achieve the optimal rate for Bernoulli designs for a range of values of $\alpha$ (namely, $\alpha \geq \frac{1}{2}$).

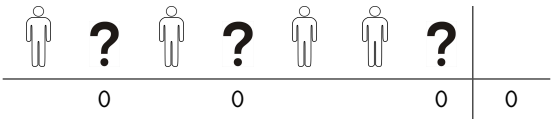
Recall the definitions of 'definite nondefective' and 'possible defective' from Definition 2.2. DD is based on the observation that if a (necessarily positive) test contains exactly one possible defective, then that item is in fact definitely defective.

**Algorithm 2.3.** The *definite defectives* (*DD*) algorithm is defined as follows.

1. We say that any item in a negative test is *definitely nondefective* (*DND*), and that any remaining item is a *possible defective* (*PD*).
2. If any PD item is the only PD item in a positive test, we call that item *definitely defective* (DD).
3. The DD algorithm outputs $\hat{K}_{\mathrm{DD}}$, the set of definitely defective items.

One justification for DD is the observation that removing nondefective items from a test does not affect the outcome of the test, so the problem is the same as if we use the submatrix with columns in PD. In addition, the principle 'assume nondefective unless proved otherwise' (used by DD) should be preferable to the rule 'assume defective unless proved otherwise' (used by COMP) under the natural assumption that defectivity is rare. We illustrate this by continuing Example 2.2.

**Example 2.3.** We present a worked example of DD. From Example 2.2, we know that items 2, 4, and 7 are the possible defectives (PD). Now consider the submatrix with the corresponding columns, illustrated in Figure 2.5. Notice that tests 4 and 5 are positive, and only contain one (possible defective) item, so we can deduce that items 2 and 4 must be defective. The defectivity status of item 7 is still unclear, but the DD algorithm marks it as nondefective. Thus, we have $\hat{K}_{\mathrm{DD}} = \{2, 4\}$.




| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |

Figure 2.5: Example of the DD algorithm. We describe the inferences that we make in Example 2.3, but give the submatrix of PD columns only, having marked the DND items discovered in Example 2.2 by COMP as nondefective (replacing question marks by outlined figures to represent our knowledge).

**Lemma 2.2.** *The estimate $\hat{K}_{\mathrm{DD}}$ generated by the DD algorithm has no false positives.*

*Proof.* Since (as in COMP), all DND items are indeed definitely nondefective, the first stage of DD makes no mistakes. Furthermore, we know that each DD

| ? | ? | ? | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |

Figure 2.6: Example of DD algorithm finding a set which is not satisfying, to illustrate Remark 2.4. In this case $\hat{\mathcal{K}}_{\mathrm{DD}} = \varnothing$.

item is indeed defective, so the second stage of inference in DD is also certainly correct. Hence, DD can only makes an error in the final step, by marking a defective item as nondefective. In other words, DD has no false positives, and $\hat{\mathcal{K}}_{\mathrm{DD}} \subseteq \mathcal{K}$. □

*Remark* 2.4. Unlike COMP, the DD algorithm does not necessarily produce an estimate which is a satisfying set. Figure 2.6 contains a simple example that illustrates this; since no test is negative, all items are marked as PDs, but no test contains a single item, so $\hat{\mathcal{K}}_{\mathrm{DD}} = \varnothing$.

However, when DD does produce a satisfying set, it must necessarily be the smallest satisfying set – that is, $\hat{\mathcal{K}}_{\mathrm{DD}} = \hat{\mathcal{K}}_{\mathrm{SSS}}$. We prove this by contradiction as follows: Assume that $\hat{\mathcal{K}}_{\mathrm{DD}}$ is a satisfying set, but not the smallest one. Then $\hat{\mathcal{K}}_{\mathrm{DD}}$ must have more elements than $\hat{\mathcal{K}}_{\mathrm{SSS}}$, meaning there exists some item $i \in \hat{\mathcal{K}}_{\mathrm{DD}} \cap (\hat{\mathcal{K}}_{\mathrm{SSS}})^{\mathsf{c}}$. By the definition of the DD algorithm, this item $i$ appears in some non-empty collection of positive tests, say indexed by $\{t_1, \dots, t_m\} \subseteq \{1, \dots, T\}$, with no other element of $\hat{\mathcal{K}}_{\mathrm{DD}}$. However, each such test $t_j$ must also contain some item $i_j \in \hat{\mathcal{K}}_{\mathrm{SSS}}$, because $\hat{\mathcal{K}}_{\mathrm{SSS}}$ is satisfying. On the other hand, by definition, no element in $\hat{\mathcal{K}}_{\mathrm{DD}} \cup \hat{\mathcal{K}}_{\mathrm{SSS}}$ appears in any negative tests, so $\{i, i_1, \dots i_m\}$ would all be counted as PD by Stage 1 of the DD algorithm. As a result, item $i$ would never appear as a lone PD item, and so would never be marked as DD, giving a contradiction.

We now discuss how to calculate the rate of DD under a Bernoulli test design. It turns out that DD outperforms the COMP rate given in Theorem 2.3 above, as is immediately deduced from the following result due to [12].

**Theorem 2.4.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, and using the DD algorithm for decoding. With a Bernoulli design and an optimal choice of the parameter $\nu = 1$ (so that $p = \frac{1}{k}$), the following rate is achievable:*

$$R_{\mathrm{Bern}}^{\mathrm{DD}} = \frac{1}{\mathrm{e} \ln 2} \min\left\{1, \frac{1-\alpha}{\alpha}\right\} \approx 0.531 \min\left\{1, \frac{1-\alpha}{\alpha}\right\}. \tag{2.10}$$

*Moreover, for $\alpha \geq 1/2$, this matches the maximum achievable rate for SSS given in Theorem 2.2.*

Note that in (2.10), the first minimand dominates for $\alpha \leq 1/2$ and the second minimand dominates for $\alpha \geq 1/2$.

For reasons of brevity, we do not provide a full proof of this result here, but instead give an outline and refer the reader to the original paper [12] for details.

First, we introduce some notation. We write $G$ for the number of nondefective items that are marked as possible defectives by the first stage of COMP. That is, there are $G$ nondefective items that only appear in positive tests (in [12] such items are referred to as 'intruding'), so that in total there are $k + G$ possible defective items.

We divide the tests into groups, according to the number of defective and possible defective items they contain. For each $i \in \mathcal{K}$, we write

$$\begin{aligned} M_i &= \ \#\text{ tests containing item } i \text{ and no other defective item,} \\ L_i &= \ \#\text{ tests containing item } i \text{ and no other possible defective item.} \end{aligned}$$

Similarly, we write $M_0 = L_0$ for the number of tests containing no defective items, and $M_+ = T - (M_0 + \sum_{i\in\mathcal{K}} M_i)$ and $L_+ = T - (L_0 + \sum_{i\in\mathcal{K}} L_i)$ for the number of remaining tests.

It is clear that the DD algorithm will correctly identify item $i$ as defective if and only if $L_i = 0$, so that we can write

$$\mathbb{P}(\text{suc}) = \mathbb{P}\left(\bigcap_{i\in\mathcal{K}} \{L_i \neq 0\}\right) = 1 - \mathbb{P}\left(\bigcup_{i\in\mathcal{K}} \{L_i = 0\}\right). \tag{2.11}$$

Unfortunately, the joint distribution of $\{L_i : i \in \mathcal{K}\}$ is not simple to characterize. However, we can make some progress by conditioning on the value of $M_0$ (the number of negative tests), to deduce that

$$M_0 \sim \text{Bin}(T, (1-p)^k), \tag{2.12}$$

$$G \mid \{M_0 = m\} \sim \text{Bin}(n-k, (1-p)^m). \tag{2.13}$$

The first result (2.12) follows because a test is negative if and only if it contains no defective items, and for Bernoulli testing this occurs independently across tests with probability $(1-p)^k$. Moreover, (2.13) follows because a nondefective item is a possible defective if and only it does not appear in positive tests, and for Bernoulli testing this occurs independently across items with probability $(1-p)^m$.

Now, we can combine the above findings with the properties of a certain function defined in [12, eq. (13)] as

$$\phi_k(q, T) := \sum_{\ell=0}^{k} (-1)^\ell \binom{k}{\ell} (1 - \ell q)^T. \tag{2.14}$$

The idea is to argue that, conditioned on $M_0$ and $G$, the distribution of $(L_i)$ is multinomial with known parameters, and [12, Lemma 31] shows that the probability

$$\mathbb{P}\left(\bigcup_{i\in\mathcal{K}} \{L_i = 0\} \,\middle|\, M_0 = m, G = g\right)$$

arising from (2.11) can be expressed in terms of the function $\phi_K$ in (2.14). Hence, we can write $\mathbb{P}(\text{suc})$ from (2.11) as an expectation over $G$ and $M_0$ of values of

$$\phi_K\left(\frac{q_1(1-p)^g}{1-q_0}, T - m_0\right),$$

for certain explicit constants. The proof proceeds by combining monotonicity properties of $\phi$ with the fact that $G$ and $M_0$ satisfy certain concentration inequalities.

*Remark* 2.5. Although we have omitted several details, we can give some intuition into the performance of DD by assuming that certain random variables concentrate around their mean. Specifically, assuming that $p = 1/k$ and that $T = Cek \ln n$, we can see that the expected number of negative tests is

$$\mathbb{E}M_0 = T(1-p)^k \sim T\mathrm{e}^{-1} = Ck \ln n.$$

Binomial concentration tells us that $M_0$ is close to its mean with high probability, so that concentration of $G$ means that it is close to

$$\mathbb{E}G \sim n(1-p)^{\mathbb{E}M_0} \sim n \exp(-p\mathbb{E}M_0) \sim n^{1-C}. \tag{2.15}$$

With this result in place, we can establish the following:

1. If $C = \max\{\alpha, 1-\alpha\}+\epsilon \geq 1-\alpha+\epsilon$, then (2.15) gives that $\mathbb{E}G \leq n^{\alpha-\epsilon} \ll k$ (since $k = \Theta(n^\alpha)$). In other words, the number of possible defectives is close to the number of true defectives, so the true defectives should not get 'drowned out'. This choice of $C$ leads to the rate (2.10) using (1.11) as before.

2. As shown in [161], Theorem 2.4 provides the best possible achievable rate for DD with Bernoulli testing, so any further improvements require changing either the algorithm or the test design, not just the analysis. For $\alpha \geq 1/2$, such a claim is trivially deduced from an observation that no algorithm can do better under Bernoulli testing. For $\alpha < 1/2$, taking $C = \max\{\alpha, 1-\alpha\} - \epsilon = 1 - \alpha - \epsilon$ in (2.15) gives $\mathbb{E}G \sim kn^\epsilon \gg k$ which leads to a large number of true defectives getting 'drowned out' by the nondefectives that are marked as possible defectives.

## 2.5 SCOMP: Sequential COMP

SCOMP is an algorithm due to Aldridge, Baldassini, and Johnson [12] that builds a satisfying set by starting from the set of definite defectives (DD) and sequentially adding new items until a satisfying set is reached. The name comes from 'Sequential COMP', as it can be viewed as a sequential version of the COMP algorithm.

**Algorithm 2.4.** The *SCOMP* algorithm is defined as follows.

1. Initialize $\hat{K}$ as the estimate $\hat{K}_{\mathrm{DD}}$ produced by the DD algorithm (Algorithm 2.3), and declare any definitely nondefective items (items appearing in a negative test) to be nondefective. The other possible defectives are not yet declared either way.

2. Any positive test is called *unexplained* if it does not contain any items from $\hat{K}$. Add to $\hat{K}$ the possible defective not in $\hat{K}$ that is in the most unexplained tests, and mark the corresponding tests as no longer unexplained. (Ties may be broken arbitrarily.)

3. Repeat step 2 until no tests remain unexplained. The estimate of the defective set is $\hat{\mathcal{K}}$.

Note that a satisfying set leaves no unexplained tests, and any set containing no definite nondefectives and leaving no unexplained tests is satisfying. Note also that the set of all possible defectives is satisfying, so the SCOMP algorithm does indeed terminate.

The following result [8] is relatively straightforward to prove.

**Theorem 2.5.** *For any given test design, any rate achievable by DD is also achievable by SCOMP. In particular, with a Bernoulli design and optimal choice of the parameter p, SCOMP can achieve the rate given in* (2.4) *above. Moreover, for $\alpha \geq 1/2$, this matches the best achievable rate obtained using SSS (Theorem 2.2).*

*Proof.* The simple idea is that for each particular test design $\mathsf{X}$ and defective set $\mathcal{K}$, whenever DD succeeds, SCOMP also succeeds. More specifically, if $\hat{\mathcal{K}}_{\mathrm{DD}} = \mathcal{K}$, then the initial choice $\hat{\mathcal{K}} = \mathcal{K}$ in step 1 of SCOMP (Algorithm 2.4) is already a satisfying set, so there are no unexplained tests to consider in step 2, and the algorithm terminates. □

It remains an interesting open problem to determine whether SCOMP has a larger achievable rate than DD for Bernoulli testing. We note from simulations such as Figure 2.2 that SCOMP appears to perform strictly better than DD for many specific problems, though it is unclear whether this converts into a strictly larger achievable rate than that of DD for $\alpha < 1/2$. In particular, as described in Section 2.7 below, Coja-Oghlan *et al.* [46] have recently proved that SCOMP provides no such improvement in rate for the near-constant column weight design.

There is an analogy between SCOMP and Chvatal's approximation algorithm for the set cover problem (see Remark 2.1). At each stage, Chvatal's algorithm [43] greedily chooses a subset that covers the largest number of currently uncovered elements in the universe of elements. Similarly, SCOMP makes a greedy choice of possibly defective items that explain as many currently unexplained positive tests as possible. For a universe of $|U| = m$ items, Chvatal's algorithm produces a solution that is at most $H_m$ times larger than the optimal set cover, where $H_m \sim \ln m$ is the $m$-th harmonic number. This can be shown to be the best possible approximation factor for a polynomial-time algorithm for set cover (in the worst case) [191, Theorem 29.31]. This means that for certain test matrices, we can view SCOMP as outputting the 'tightest possible polynomial-time approximation to the smallest satisfying set'. However, this does not preclude the possibility of improved efficient approximations under other well-chosen test matrices.

## 2.6 Linear programming relaxations

Linear programming (LP) algorithms have been proposed as a way to approximate SSS with practical runtime and storage requirements, by solving a relaxed version of the smallest satisfying set problem.

Specifically, recalling that $x_{ti}$ indicates if item $i$ is in test $t$, and $y_t$ is the outcome of test $t$, a smallest satisfying set corresponds to an optimal solution

to the integer program

$$\begin{aligned} \text{minimize}_{\mathbf{z}} \ & \sum_{i=1}^{n} z_i \\ \text{subject to} \ & \sum_{i=1}^{n} x_{ti} z_i \geq 1 \quad \text{when } y_t = \mathtt{1}, \\ & \sum_{i=1}^{n} x_{ti} z_i = 0 \quad \text{when } y_t = \mathtt{0}, \\ & z_i \in \{0, 1\}. \end{aligned}$$

We hope that the optimal $\mathbf{z}$ will be close to the true defectivity vector $\mathbf{u}$ introduced in Definition 1.2, since taking $\mathbf{z} = \mathbf{u}$ will satisfy the constraints. In general, we think of each 0–1 vector $\mathbf{z}$ as the indicator function of some putative defective set $\mathcal{L}$, with $\mathcal{L} = \mathcal{L}(\mathbf{z}) := \{i : z_i = 1\}$. The first two constraints on $\mathbf{z}$ ensure that $\mathcal{L}(\mathbf{z})$ is satisfying in the sense of Definition 2.1, by considering the positive and negative tests respectively. Hence, each $\mathbf{z}$ that achieves the minimal value of the linear program is the indicator function of a satisfying set of minimal size, i.e., $\hat{\mathcal{K}} = \{i : z_i = 1\}$ is a smallest satisfying set.

The LP approach attempts to estimate the defective set via a relaxed version of the 0–1 problem, where each $z_i$ can be any nonnegative real number. That is, the optimization formulation is exactly as above, but with each constraint $z_i \in \{0, 1\}$ replaced by

$$z_i \geq 0.$$

Linear programs of this form can be solved efficiently: the ellipsoid algorithm is guaranteed to find a solution in polynomial time, though it is typically outperformed in practice by the simplex algorithm. (See, for example, [47, p. 897] for a discussion of the running times of linear programming algorithms.)

There are various heuristics for how to turn an optimal solution $\mathbf{z} = (z_i)$ to the relaxed program into an estimate of the defective set. For example, one could consider the following crude method: If there is any $i$ with $z_i \notin \{0, 1\}$, declare a global error; otherwise, estimate $\hat{\mathcal{K}} = \{i : z_i = 1\}$ to be the defective set. Malioutov and Malyutov [139] suggest an estimate $\hat{\mathcal{K}} = \{i : z_i > 0\}$, and show strong performance on simulated problems. Note that this rule will always provide a satisfying set, since each positive test will have some possible defective $i$ with $z_i$ that is declared defective. Alternatively, the estimate $\hat{\mathcal{K}} = \{i : z_i \geq 1/2\}$ appears to be (very) slightly better in simulations, but does not guarantee a satisfying set.

For the purposes of the following theorem, it suffices that in the event that all $z_i$ are 0 or 1, the heuristic chooses $\hat{\mathcal{K}} = \{i : z_i = 1\}$, as any sensible heuristic surely must. This theorem is due to [8], and shows that the above LP approach, like SCOMP, is at least as good as DD.

**Theorem 2.6.** *For any given test design, any rate achievable by DD is also achievable by LP. In particular, with a Bernoulli design and optimal choice of the parameter $p$, LP can achieve the rate given in* (2.4) *above. Moreover, for $\alpha \geq 1/2$, this matches the best achievable rate obtained using SSS (Theorem 2.2).*

*Proof.* Again, for each particular test design $\mathsf{X}$ and defective set $\mathcal{K}$, whenever DD succeeds LP also succeeds. To be precise, any item $i$ which appears in some negative test $t$ must have $z_i = 0$ in order to satisfy the second constraint of the linear program, $\sum_{i=1}^{n} x_{ti} z_i = 0$. Furthermore, if a positive test $t$ contains only one possible defective $i$, the LP solution must have $z_i \geq 1$ to ensure the first constraint $\sum_{i=1}^{n} x_{ti} z_i \geq 1$ holds, and it will choose $z_i = 1$ to minimize $\sum_i z_i$. Finally, if DD succeeds then these definite defectives form a satisfying set, so all constraints are satisfied, and the algorithm will set all other $z_i = 0$, to minimize $\sum_i z_i$. □

As with SCOMP, while simulation evidence suggests that LP outperforms DD for certain problems (see for example Figure 2.2), it remains an interesting open problem to determine whether it achieves a strictly larger rate than DD for $\alpha < 1/2$.

We briefly mention an earlier result that used linear programming to get a nonzero rate for all $\alpha$, albeit with a much lower rate than that of Theorem 2.6. The *LiPo* algorithm of Chan *et al.* [34] is based on relaxing a similar integer program, and further assumes the decoder knows $k$ exactly, so the linear program can be phrased as a feasibility problem. They show that LiPo achieves the rate

$$R_{\mathrm{Bern}}^{\mathrm{LiPo}} = \frac{1}{\frac{8}{3}\,\mathrm{e}^2 \ln 2} \frac{1-\alpha}{1+\alpha} \approx 0.073 \frac{1-\alpha}{1+\alpha}. \tag{2.16}$$

## 2.7 Improved rates with near-constant tests-per-item

Throughout this chapter, we have focused on Bernoulli testing designs, where each item is independently placed in each test with a given probability, and hence $\mathsf{X}$ contains i.i.d. Bernoulli entries. Such a design is conceptually simple, is typically the easiest to analyse mathematically, and is known to be information-theoretically optimal for $k = O(n^{1/3})$ (see Chapter 4).

However, it turns out that we can do better in certain cases. Below, we will see that an alternative random design based on *near-constant tests-per-item* can improve the COMP and DD rates by a factor of $e(\ln 2)^2 \simeq 1.306$, leading to two key implications. First, for $k = \Theta(n^\alpha)$ with $\alpha$ sufficiently close to one, this combination outperforms Bernoulli testing used in conjunction with with *any* decoder. Second, for small $\alpha$, this combination improves on the best known rate for Bernoulli testing under any *practical* decoder. The results of this section are due to Johnson, Aldridge, and Scarlett [113] and Coja-Oghlan *et al.*[46].

The following definition formally introduces the random design that provides the above-mentioned improved rates.

**Definition 2.3.** The *near-constant column weight* (or *near-constant tests-per-item*) design with parameter $\nu > 0$ forms a group testing matrix $\mathsf{X}$ in which $L = \nu T/k$ entries of each column are selected uniformly at random with replacement and set to one,[1] with independence between columns. The remaining entries of $\mathsf{X}$ are set to zero.

[1] We ignore rounding issues here, and note that the results are unaffected by whether we set $L = \lfloor \nu T/k \rfloor$ or $L = \lceil \nu T/k \rceil$.

Since we sample with replacement, some items may be in fewer than $L$ tests, but typically only slightly fewer, hence the terminology 'near-constant'. This is a mathematical convenience that makes the analysis more tractable. The parametrization $L = \nu T/k$ is chosen because such scaling with $\nu = \Theta(1)$ turns out to be optimal, analogously to the scaling $p = \nu/k$ in Definition 2.2. (In Section 5.8, we will briefly survey a setting in which the number of tests per item is constrained to be much smaller than $O(T/k)$.) An intuitive reason as to why the above design may be preferable to the Bernoulli design is that it prevents any item from being included in too few tests.

While Definition 2.3 suffices for our purposes, it is worth mentioning that it is one of a variety of related randomized designs that have appeared in the literature. Indeed, a variety of works have considered *(exactly-)constant tests-per-item* (see for example [136, 148]). There is evidence that such matrices provide similar gains, but to our knowledge, this has not been proved in the same generality and rigour as the case of near-constant tests-per-item. In addition, matrices with a constant row weight have been considered [33], but with no proven gains over Bernoulli testing.

We first present the rate achieved by the simple COMP algorithm (see Section 2.3).

**Theorem 2.7.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, and using the COMP algorithm for decoding. With a near-constant column weight design and an optimized parameter $\nu = \ln 2$, the maximum achievable rate using COMP is*

$$\overline{R}_{\mathrm{NCC}}^{\mathrm{COMP}} = \ln 2(1-\alpha) \approx 0.693(1-\alpha). \tag{2.17}$$

*Proof sketch.* We omit the details of the proof for brevity, and refer the reader to [113]. The key idea is again to formulate a coupon-collector problem (see Remark 2.2).

First, we consider the total number of positive tests $T_1$. A given test $t$ is negative if, for each of the $k$ defective items, none of the $L = \nu T/k$ choices of column entries is $t$. Since these choices take place independently with replacement, this is the same as choosing $kL = \nu T$ entries in total, all independently with replacement. Hence, the probability that test $t$ is negative is $(1-1/T)^{\nu T} \sim \mathrm{e}^{-\nu}$, so the expected number of positive tests $\mathbb{E}T_1 \sim T(1-\mathrm{e}^{-\nu})$.

Now, since changing one choice of column entry changes the number of positive tests by at most 1, the random variable $T_1$ satisfies the bounded difference property in the sense of McDiarmid [147], which allows us to prove a standard concentration bound. Specifically we can deduce by McDiarmid's inequality that $T_1$ is close to its mean, so that $T_1 \approx T(1-\mathrm{e}^{-\nu})$ with high probability.

Finally, conditioned on $T_1$, each nondefective item appears in some negative test with probability $1-(T_1/T)^L$, independently of one another. Hence, assuming the concentration result holds (replacing $T_1$ by its mean), we find that

$$\mathbb{P}(\mathrm{suc}) \sim (1-(T_1/T)^L)^{n-k} \sim (1-(1-\mathrm{e}^{-\nu})^L)^n = (1-(1-\mathrm{e}^{-\nu})^{\nu T/k})^n.$$

It is easy to check that $(1-\mathrm{e}^{-\nu})^\nu$ is maximized at $\nu = \ln 2$, where it takes the value $\mathrm{e}^{-(\ln 2)^2}$. Thus, choosing $T = (1+\delta)(k\ln n)/(\ln 2)^2$ gives us that $(1-\mathrm{e}^{-\nu})^{\nu T/k} = \mathrm{e}^{-(1+\delta)\ln n}$. This allows us to deduce that $\mathbb{P}(\mathrm{suc}) \sim (1-n^{-1+\delta})^n \sim \exp(-n^{-\delta})$, which tends to 1.

In terms of rates, again using Definition 1.7 and (1.11), we can deduce that this equates to a rate of $(1-\alpha)(k\ln n)/(T\ln 2)$ which approaches $\ln 2(1-\alpha)$ as required. □

Comparing with Theorem 2.3, we see that for the COMP algorithm, the near-constant column weight design provides an improvement of roughly 30.6% over Bernoulli testing. In addition, the rate of Theorem 2.7 improves even over that of the DD algorithm with Bernoulli testing, both for sufficiently small $\alpha$ and sufficiently high $\alpha$. See Figure 2.1 for an illustration.

We now turn to the DD algorithm (see Section 2.4), which strictly improves on Theorem 2.7 for all $\alpha \in (0,1)$.

**Theorem 2.8.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^\alpha)$ for some $\alpha \in (0,1)$, and using the DD algorithm for decoding. Under a near-constant column weight design with an optimized parameter $\nu = \ln 2$, the following rate is achievable:*

$$R_{\mathrm{NCC}}^{\mathrm{DD}} = (\ln 2)\,\min\left\{1, \frac{1-\alpha}{\alpha}\right\} \approx 0.693 \min\left\{1, \frac{1-\alpha}{\alpha}\right\}. \tag{2.18}$$

*Moreover, for $\alpha \geq 1/2$, this achieves the maximum achievable rate for SSS using this design, $\overline{R}_{\mathrm{NCC}}^{\mathrm{SSS}}$ (see Theorem 2.9 below).*

The proof of this result bears some similarity to that of Bernoulli testing, but is more technically challenging. The interested reader is referred to [113].

Comparing Theorem 2.8 to Theorem 2.4, we see that the achievable rate with the near-constant column weight design is roughly 30.6% higher than the Bernoulli design – the same gain as that observed for COMP (see Figure 2.1). Overall, Theorem 2.8 currently provides the best known rate for any practical algorithm and any testing design for all $\alpha \in (0,1)$.

As in the case of Bernoulli testing, one immediately deduces (by Theorem 2.5 and Theorem 2.6 respectively) that the SCOMP and LP algorithms (see Sections 2.5 and 2.6) also achieve the rate (2.18). Moreover, in [46], it was shown that (2.18) is the *maximum* achievable rate for the SCOMP (and DD) algorithm under the near-constant column weight design, meaning that asymptotically SCOMP does not outperform DD. The fact that the rate cannot exceed $(\ln 2)\frac{1-\alpha}{\alpha}$ comes from Theorem 2.9 below. As for the $\ln 2$ term, the idea is to show that for rates above $\ln 2$ there exist many nondefectives that explain the maximum possible number $L$ of tests, and hence even the first iteration of SCOMP fails. It remains an open problem as to whether a similar phenomenon holds for other test designs, such as Bernoulli testing.

Next, we present a converse bound [113, Theorem 4] for SSS, revealing that DD is optimal for the near-constant column weight design when $\alpha$ is not too small.

**Theorem 2.9.** *Consider noiseless nonadaptive group testing with exact recovery and the small error criterion, with $k = \Theta(n^\alpha)$ for $\alpha \in (0,1)$, and using the SSS algorithm for decoding. With a near-constant column weight design, the maximum achievable rate is bounded above by*

$$\overline{R}_{\mathrm{NCC}}^{\mathrm{SSS}} \leq \min\left\{1, \ln 2\,\frac{1-\alpha}{\alpha}\right\}. \tag{2.19}$$

This result is analogous to Theorem 2.2. In Chapter 4, we will survey a recent result of Coja-Oghlan *et al.* [46] showing that the rate (2.19) is achievable for all $\alpha \in (0, 1)$, and deduce that (2.19) gives the maximum achievable rate for the near-constant column weight design.

# Chapter 3

# Algorithms for Noisy Group Testing

## 3.1 Noisy channel models

In Chapter 2, we focused on noiseless group testing algorithms and their theoretical guarantees. From both a theoretical and practical perspective, these algorithms (as presented) rely strong on the assumption that there is no noise. In this chapter, we give an overview of a variety of algorithms that are designed to handle noisy scenarios, most of which build on the ideas from the noiseless setting. We initially present heuristic approaches, and then move on to techniques with theoretical guarantees.

For many of the applications described in Section 1.7, it is clearly an unrealistic modelling assumption that the tests would be able to perfectly identify whether any defective item is present in the pool. There are a variety of ways of modelling the noise, which affect the algorithms and their performance in different ways. We proceed by giving several illustrative examples.

Recall that standard noiseless group testing can be formulated componentwise using the Boolean OR operation as $y_t = \bigvee_{i \in \mathcal{K}} x_{ti}$ (see (1.2)). One of the simplest noise models simply considers the scenario where these values $\bigvee_{i \in \mathcal{K}} x_{ti}$ are flipped independently at random with a given probability.

**Example 3.1** (Binary symmetric noise)**.** In the binary symmetric noise model, the $t$-th test outcome is given by

$$Y_t = \begin{cases} \bigvee_{i \in \mathcal{K}} X_{ti} & \text{with probability } 1 - \rho \\ 1 - \bigvee_{i \in \mathcal{K}} X_{ti} & \text{with probability } \rho. \end{cases} \tag{3.1}$$

This is, each test is flipped independently with probability $\rho$.

While the binary noise model is an interesting one, many applications in Section 1.7 suggest that false positive tests and false negative tests may occur with different probabilities. We proceed by presenting some examples, maintaining the standard assumption that distinct test outcomes are conditionally independent given $\mathsf{X}$. Furthermore, we assume that each test has the same probability distribution specifying its outcome, and that this distribution depends on the

test design $\mathsf{X}$ only through the number of defective items in the test and the total number of items in the test.

For reasons of generality, we no longer insist that the test outcomes $y_t$ can only take values in $\{\mathtt{0}, \mathtt{1}\}$, but rather consider the case of $y_t \in \mathcal{Y}$ for some finite alphabet $\mathcal{Y}$. We follow in part the notation of [6, Section 6.3].

**Definition 3.1.** We define the probability transition function $p(\cdot \mid m, \ell)$ such that for a test containing $m$ items, $\ell$ of which are defective, for each outcome $y \in \mathcal{Y}$ we have

$$\mathbb{P}\left(Y_t = y \;\middle|\; \sum_{i=1}^{n} X_{ti} = m, \sum_{i \in \mathcal{K}} X_{ti} = \ell\right) = p(y \mid m, \ell), \tag{3.2}$$

independently of all other tests.

In other words, $p(y \mid m, \ell)$ is the probability of observing outcome $y$ from a test containing $\ell$ defective items and $m$ items in total. Note that $\sum_{y \in \mathcal{Y}} p(y \mid m, \ell) = 1$ for all $m$ and $\ell$.

For example, the standard noiseless group testing model has probability transition function

$$\begin{aligned} p(\mathtt{1} \mid m, \ell) &= 1 \quad \text{if } \ell \geq 1, & p(\mathtt{0} \mid m, \ell) &= 0 \quad \text{if } \ell \geq 1, \\ p(\mathtt{1} \mid m, \ell) &= 0 \quad \text{if } \ell = 0, & p(\mathtt{0} \mid m, \ell) &= 1 \quad \text{if } \ell = 0, \end{aligned} \tag{3.3}$$

independent of $m$. The binary symmetric noise model of Example 3.1 has probability transition function

$$\begin{aligned} p(\mathtt{1} \mid m, \ell) &= 1 - \rho \quad \text{if } \ell \geq 1, & p(\mathtt{0} \mid m, \ell) &= \rho \quad \text{if } \ell \geq 1, \\ p(\mathtt{1} \mid m, \ell) &= \rho \quad \text{if } \ell = 0, & p(\mathtt{0} \mid m, \ell) &= 1 - \rho \quad \text{if } \ell = 0. \end{aligned} \tag{3.4}$$

Definition 3.1 captures a variety of other noise models, one of which is the addition noise model of [17]. Here false negative tests never occur, but false positive tests occur independently with a given probability $\varphi$.

**Example 3.2** (Addition noise)**.** In the addition noise model, the probability transition function is given by

$$\begin{aligned} p(\mathtt{1} \mid m, \ell) &= 1 \quad \text{if } \ell \geq 1, & p(\mathtt{0} \mid m, \ell) &= 0 \quad \text{if } \ell \geq 1, \\ p(\mathtt{1} \mid m, \ell) &= \varphi \quad \text{if } \ell = 0, & p(\mathtt{0} \mid m, \ell) &= 1 - \varphi \quad \text{if } \ell = 0, \end{aligned} \tag{3.5}$$

where $\varphi \in (0, 1)$ is a noise parameter.

We note that the noise processes described in Examples 3.1 and 3.2 can both be thought of as sending the outcome of standard noiseless group testing through a noisy communication channel (see Definition 3.3 below). Another interesting model, which cannot be represented in this way, is the dilution model of [17]. This captures the idea that in some scenarios (for example in DNA testing), the more defectives are present, the more likely we are to observe a positive test.

In this model, the outcome of a test containing $\ell \geq 1$ defectives will be positive if and only if a Binomial$(\ell, 1 - \vartheta)$ random variable is at least one. Equivalently, this can be thought of as a scenario where every defective item included in the test only 'behaves as a defective' with probability $1 - \vartheta$, whereas with probability $\vartheta$ it is 'diluted'.

**Example 3.3** (Dilution noise)**.** In the dilution noise model, the probability transition function is given by

$$p(\mathtt{1} \mid m, \ell) = 1 - \vartheta^{\ell}, \qquad p(\mathtt{0} \mid m, \ell) = \vartheta^{\ell}, \quad \text{for all } \ell \geq 0, \tag{3.6}$$

where $\vartheta \in (0, 1)$ is a noise parameter.

An analogous model to the dilution noise model is the Z channel noise model, in which tests containing defective items are erroneously negative with some fixed probability.

**Example 3.4** (Z channel noise)**.** In the Z channel noise model, the probability transition function is given by

$$\begin{array}{llll} p(\mathtt{1} \mid m, \ell) = 1 - \vartheta & \text{if } \ell \geq 1, & p(\mathtt{0} \mid m, \ell) = \vartheta & \text{if } \ell \geq 1, \\ p(\mathtt{1} \mid m, \ell) = 0 & \text{if } \ell = 0, & p(\mathtt{0} \mid m, \ell) = 1 & \text{if } \ell = 0, \end{array} \tag{3.7}$$

where $\vartheta \in (0, 1)$ is a noise parameter.

By analogy, the addition noise channel (Example 3.2) can also be viewed as 'reverse Z channel' noise.

An example to illustrate the fact that the alphabet $\mathcal{Y}$ need not be $\{\mathtt{0}, \mathtt{1}\}$ is the erasure noise model, where each test may fail to give a conclusive result. We represent such an outcome by a question mark ?. In this case, $\mathcal{Y} = \{\mathtt{0}, \mathtt{1}, ?\}$, and the noise model is defined as follows.

**Example 3.5** (Erasure noise)**.** In the erasure noise model, the probability transition function is given by

$$\begin{array}{llll} p(\mathtt{1} \mid m, \ell) = 1 - \xi & \text{if } \ell \geq 1, & p(? \mid m, \ell) = \xi & \text{if } \ell \geq 1, \\ p(? \mid m, \ell) = \xi & \text{if } \ell = 0, & p(\mathtt{0} \mid m, \ell) = 1 - \xi & \text{if } \ell = 0, \end{array} \tag{3.8}$$

where $\xi \in (0, 1)$ is a noise parameter, and all other values of $p(\,\cdot \mid m, \ell)$ are zero.

Next, we provide another example of interest from [125], falling under the broad category of *threshold group testing* (e.g., see [41, 52]). In this example, a positive result is attained when the proportion of items in the test exceeds some threshold $\overline{\theta}$, a negative result is obtained when the proportion is below another threshold $\underline{\theta}$ (with $\underline{\theta} \leq \overline{\theta}$), and positive and negative outcomes are equally likely when the proportion is in between these thresholds.

**Example 3.6** (Threshold group testing)**.** In the probabilistic threshold group testing noise model, the probability transition function is given by

$$\begin{array}{llll} p(\mathtt{1} \mid m, \ell) = 1 & \text{if } \frac{\ell}{m} \geq \overline{\theta}, & p(\mathtt{0} \mid m, \ell) = 0 & \text{if } \frac{\ell}{m} \geq \overline{\theta}, \\ p(\mathtt{1} \mid m, \ell) = 0 & \text{if } \frac{\ell}{m} \leq \underline{\theta}, & p(\mathtt{0} \mid m, \ell) = 1 & \text{if } \frac{\ell}{m} \leq \underline{\theta}, \\ p(\mathtt{1} \mid m, \ell) = \frac{1}{2} & \text{if } \underline{\theta} < \frac{\ell}{m} < \overline{\theta}, & p(\mathtt{0} \mid m, \ell) = \frac{1}{2} & \text{if } \underline{\theta} < \frac{\ell}{m} < \overline{\theta}\ , \end{array} \tag{3.9}$$

where $\underline{\theta} \leq \overline{\theta}$ are thresholds.

Another variation in [125] instead assumes that the probability of a positive test increases from 0 to 1 in a linear fashion in between the two thresholds,

rather than always equalling $\frac{1}{2}$. It is worth noting that, while our focus is on random noise models, most works on threshold group testing have focused on *adversarial* noise [41, 52].

We remark that the noise models described in Equations (3.3), (3.4), (3.5), (3.6), (3.7), and (3.8) above share the property that $p(\cdot \mid m, \ell)$ does not depend on $m$. Of course, this need not be the case in general, as Example 3.6 shows. However, this property is sufficiently useful that we follow [6, Definition 6.11] in explicitly naming it.

**Definition 3.2.** We say that a noise model satisfies the *only defects matter* property if the probability transition function is of the form

$$p(y \mid m, \ell) = p(y \mid \ell). \tag{3.10}$$

Properties of this type have been exploited in general sparse estimation problems beyond group testing (see for example [5, 144, 165]). In these cases, this property means that only the columns of a measurement matrix that correspond to the nonzero entries of a sparse vector impact the samples, and further that the corresponding output distribution is permutation-invariant with respect to these columns.

While the only defects matter property, Definition 3.2, does not hold in general, it plays a significant role in many proofs of noisy group testing results. For example, this assumption is used throughout Chapter 4 to provide information-theoretic achievability and converse results. Some further evidence for the value of Definition 3.2 is that Furon [83] gives examples where 'only defects matter' does not hold and a nonzero rate cannot be achieved.

A further interesting special case of Definition 3.2 is when the noisy group testing process can be thought of as sending the outcome of standard noiseless group testing through a noisy 'communication' channel.

**Definition 3.3** (Noisy defective channel)**.** If we can express

$$p(y \mid m, \ell) = p(y \mid \mathbf{1}\{\ell \geq 1\}), \tag{3.11}$$

where $p(y \mid \mathbf{1}\{\ell \geq 1\})$ is the transition probability function of a noisy binary communication channel, then we say that the *noisy defective channel property* holds.

In the case that this property holds, the following result is stated in [20].

**Theorem 3.1.** *If the noisy defective channel property (Definition 3.3) holds then the group testing capacity $C$ (in the sense of Definition 1.8) satisfies the following, regardless of whether the test design is adaptive or nonadaptive:*

$$C \leq C_{\text{chan}}, \tag{3.12}$$

*where $C_{\text{chan}}$ is the Shannon capacity of the corresponding noisy communication channel $p(y \mid \mathbf{1}\{\ell \geq 1\})$.*

In fact, a similar result holds more generally even when the channel $p(y|\ell)$ has a nonbinary input indicating the number of defectives in the test; however, it is primarily the form stated in Theorem 3.1 that has been useful when comparing to achievability results.

One may be tempted to conjecture that for $k = o(n)$, equality holds in (3.12) for *adaptive* group testing. This conjecture was recently shown to be true [161] for the Z channel noise model (Example 3.4), but false when $k = \Theta(n^\alpha)$ for $\alpha$ sufficiently close to one under the binary symmetric noise and addition noise models (Examples 3.1 and 3.2).

The argument given in [20] to prove Theorem 3.1 uses the fact that the test outcome vector $\mathbf{y} = (y_1, \dots, y_T)$ acts like the output of the channel whose input codeword is indexed by the defective set. Since the transmission of information is impossible at rates above capacity, it certainly remains impossible in the presence of the extra constraints imposed by the group testing problem.

We remark that the noisy defective channel property of Definition 3.3 is satisfied by the models described in Equations (3.3), (3.4), (3.5), (3.7), and (3.8) (though not the dilution model (3.6) or threshold model (3.9)). For example, we can deduce that the binary symmetric model Definition 3.4 has group testing capacity $C \leq 1 - h(\rho)$. It remains an open problem to determine under what conditions this bound is sharp; in Section 4.5, we will see that it is sharp in the sparse regime $k = O(n^\alpha)$ when $\alpha$ is sufficiently small.

One noisy model where we can determine the adaptive group testing capacity is the erasure model; the following result is from [20, Theorem 1.3.1].

**Theorem 3.2.** *The capacity of adaptive group testing is $C = 1 - \xi$ for the erasure model of Example 3.5 when $k = o(n)$.*

*Proof.* This is achieved by simply using a noiseless adaptive group testing scheme (see Section 1.5), and repeating tests for which $y_t = ?$. Standard concentration-of-measure results tell us that, for any $\epsilon > 0$, with high probability no more than $T(\xi + \epsilon)$ tests will need repeating, and the result follows from Theorem 1.3. □

A similar argument can be used to determine bounds on the rates of non-adaptive algorithms under Bernoulli designs for the erasure noise model of Example 3.5. Again, with high probability, given $T$ tests, we know there should be at least $T(1 - \xi - \epsilon)$ tests that are not erased. Hence, simply ignoring the tests which return a ?, it is as if we have been given a Bernoulli design matrix with at least $T(1 - \xi - \epsilon)$ rows.

Hence, for example, if there are $k = \Theta(n^\alpha)$ defectives, then building on Theorem 2.4, we can achieve a rate of

$$\frac{1-\xi}{\mathrm{e}\ln 2}\min\left\{1, \frac{1-\alpha}{\alpha}\right\} \tag{3.13}$$

using the DD algorithm and a Bernoulli test design. Similarly, building on Theorem 2.2, we know that no algorithm can achieve a rate greater than

$$(1-\xi)\max_{\nu>0}\min\left\{h(\mathrm{e}^{-\nu}),\ \frac{\nu}{\mathrm{e}^\nu \ln 2}\frac{1-\alpha}{\alpha}\right\}. \tag{3.14}$$

for Bernoulli designs.

We also briefly mention that since the addition noise channel (Example 3.2) satisfies the property that a negative outcome is definitive proof of no defectives being present, we can easily extend the analysis of the COMP algorithm to deduce a counterpart to Theorem 2.3 (see [168, Lemma 1] for details). Specifically,

since a proportion $\varphi$ of the negative tests are flipped at random, we can achieve a rate of

$$\frac{1-\varphi}{\mathrm{e}\ln 2}(1-\alpha) \tag{3.15}$$

using the COMP algorithm (see [168, eq. (149)]).

In the remainder of the chapter, we describe a variety of algorithms that can be used to solve noisy group testing problems in the presence of both false positive tests and false negative tests (e.g., for the binary symmetric noise model). Most of these are in fact extensions of the noiseless algorithms presented in Chapter 2, and like that chapter, we focus our attention on nonadaptive Bernoulli test designs, the small-error recovery criterion, and the scaling $k = \Theta(n^\alpha)$ with $\alpha \in (0, 1)$.

## 3.2 Noisy linear programming relaxations

Recall the linear programming relaxation for the noiseless setting in Section 2.6. A similar idea can be used in the noisy setting by introducing *slack variables*, which leads to a formulation allowing 'flipped' test outcomes but paying a penalty in the objective function for doing so. Using this idea, the following formulation was proposed in [139]:

$$\begin{aligned}
\text{minimize}_{\mathbf{z},\boldsymbol{\xi}} \quad & \sum_{i=1}^{n} z_i + \zeta \sum_{t=1}^{T} \xi_j & \\
\text{subject to} \quad & z_i \geq 0 & \\
& \xi_t \geq 0 & \\
& \xi_t \leq 1 & \text{when } y_t = 1 \\
& \sum_{i=1}^{n} x_{ti} z_i = \xi_t & \text{when } y_t = 0 \\
& \sum_{i=1}^{n} x_{ti} z_i + \xi_t \geq 1 & \text{when } y_t = 1.
\end{aligned}$$

As in the noiseless setting, $\mathbf{z}$ represents an estimate of the defectivity indicator vector $\mathbf{u}$ (see Definition 1.2), whereas here we also have a vector of $T$ slack variables $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_T)$. The parameter $\zeta$ controls the trade-off between declaring a small number of items to be defective (sparsity) and the degree to which the test outcomes are in agreement with the decoded $z_i$ (most slack variables being zero). Observe that if we were to further constrain each $z_i$ and $\xi_t$ to be binary-valued (0 or 1), then the above formulation would be minimizing a weighted combination of the number of (estimated) defectives and the number of 'flipped' tests. Such a binary-valued minimization problem, with a suitable choice of $\zeta$, can also be shown to be equivalent to *maximum a posteriori* (MAP) decoding under an i.i.d. defectivity model (see the Appendix to Chapter 1) and symmetric noise (see Example 3.1).

The above formulation treats false positive tests and false negative tests equally. However, it can also be modified to weigh the two differently; in the extreme case, if it is known that a test with no defectives *definitely* results in a negative outcome (e.g., dilution noise of Example 3.3, or Z channel noise of

Example 3.4), then we could replace all of the slack variables corresponding to negative tests by zero. An analogous statement holds true when a test with at least one defective *definitely* results in a positive outcome (e.g., addition noise of Example 3.2).

To the bets of our knowledge, no theoretical results are known for the above noisy LP relaxation. However, this method has been seen to provide state-of-the-art performance in numerical experiments [139]; see Section 3.7 for an illustration.

A related noisy LP relaxation using *negative tests only* was proved to achieve positive rates in [34]. However, there are two notable limitations. First, from a theoretical view, the constants were not optimized in the proofs, and so the rates are far from optimal. Second, from a practical view, ignoring the tests with positive outcomes can significantly worsen the performance.

## 3.3 Belief propagation

A decoding algorithm based on belief propagation was described in [171, Section III]. Although there was no attempt to calculate performance bounds or rates, there was some numerical evidence presented to show that this approach can work very well (see also Section 3.7). The apparent success of belief propagation may not be a surprise, since it has enjoyed considerable success for the decoding of LDPC codes over noisy channels, a problem that shares characteristics with group testing.

Recall from Definition 1.2 that we write $u_i = \mathbf{1}\{i \in \mathcal{K}\}$ to indicate whether or not item $i$ is defective. The idea is to estimate the defective set by working with the marginals of the posterior distribution, and for each $i$, seek to estimate $u_i$ as

$$\widehat{u_i} := \operatorname*{arg\,max}_{u_i \in \{0,1\}} \mathbb{P}(u_i \mid \mathbf{y}), \tag{3.16}$$

where $\mathbf{y}$ is the vector of test outcomes. Clearly, we would prefer to optimize this posterior probability as a function of all the $(u_i)_{i\in\{1,\dots,n\}}$, but this would be computationally infeasible due to the size of the search space.

While exactly computing the probability $\mathbb{P}(u_i \mid \mathbf{y})$ appearing in (3.16) is also difficult, we can approximately compute it using loopy belief propagation. To understand this, we set up a bipartite graph with $n$ nodes on one side corresponding to items, and $T$ nodes on the other side corresponding to tests. Each test node is connected to all of the nodes corresponding to items included in the test. See Figure 3.1 for a simple example.

Assuming that $k$ out of $n$ items are defective, a natural prior is given by

$$\mathbb{P}(U_i = 1) = \frac{k}{n} =: q, \tag{3.17}$$

and for analytical tractability, an independent prior $\mathbb{P}(\mathbf{u}) = \prod_{i=1}^{n} \mathbb{P}(u_i)$ is adopted. Even for a combinatorial prior where $\mathcal{K}$ is uniform over $\binom{n}{k}$ possible defective sets, (3.17) yields a good approximation for large $k$ due to concentration of measure. In either case, as described, this method requires at least approximate knowledge of $k$.

In accordance with general-purpose techniques for loopy belief propagation (e.g., see [134, Ch. 26]), messages are iteratively passed from items to tests and

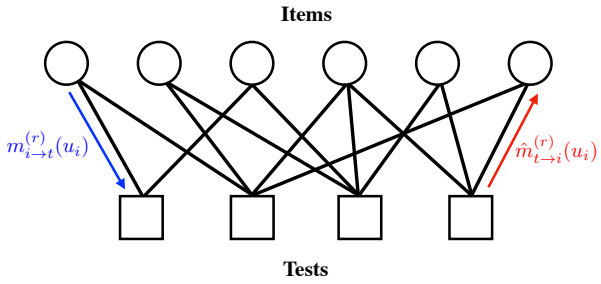



Figure 3.1: Example bipartite graph used in belief propagation decoding. Edges represent the inclusion of items in tests, and messages are passed in both directions.

tests to items. Letting $\mathcal{N}(i)$ and $\mathcal{N}(t)$ denote the neighbours of an item node and test node respectively, the item-to-test and test-to-item message are given as follows [171]:

$$m^{(r)}_{i\to t}(u_i) \propto \big(q\mathbf{1}\{u_i = 1\} + (1-q)\mathbf{1}\{u_i = 0\}\big) \prod_{t' \in \mathcal{N}(i)\setminus\{t\}} \widehat{m}^{(r)}_{t'\to i}(u_i) \tag{3.18}$$

$$\widehat{m}^{(r)}_{t\to i}(u_i) \propto \sum_{\{u_{i'}\}_{i'\in\mathcal{N}(t)\setminus\{i\}}} \mathbb{P}(y_t \mid u_{[t]}) \prod_{i'\in\mathcal{N}(t)\setminus\{i\}} m^{(r)}_{i'\to t}(u_i), \tag{3.19}$$

where $r$ indexes the round of message passing, $\propto$ denotes equality up to a normalizing constant, and $u_{[t]}$ denotes the sub-vector of $\mathbf{u}$ corresponding to the items in test $t$, which are the only ones that impact $y_t$. These messages amount to updating beliefs of the test outcomes $y_t$ in terms of $u_i$, and beliefs of $u_i$ in terms of the test outcomes $y_t$. By iterating these steps, we hope to converge to a sufficiently good approximation of the posterior.

The sum over $\{u_{i'}\}_{i'\in N(t)\setminus\{i\}}$ in (3.19) grows exponentially in the number of items in the test, so these messages are still expensive to compute if the computation is done naively. Fortunately, at least for certain noise models, it is possible to rewrite the messages in a form that permits efficient computation. In [171], this was shown for the following model that combines the addition and dilution models of Examples 3.2 and 3.3:

$$p(\mathtt{1} \mid m, \ell) = 1 - (1-\varphi)\vartheta^\ell, \quad p(\mathtt{0} \mid m, \ell) = (1-\varphi)\vartheta^\ell, \quad \text{for all } \ell \geq 0. \tag{3.20}$$

Observe that setting $\varphi = 0$ recovers the dilution model, whereas setting $\vartheta = 0$ recovers the addition model.

For this model, it is convenient to work with log-ratios of the messages, defined as

$$L^{(r)}_{i\to t} = \ln \frac{m^{(r)}_{i\to t}(\mathtt{1})}{m^{(r)}_{i\to t}(\mathtt{0})} \quad \text{and} \quad \widehat{L}^{(r)}_{t\to i} = \ln \frac{\widehat{m}^{(r)}_{t\to i}(\mathtt{1})}{\widehat{m}^{(r)}_{t\to i}(\mathtt{0})}. \tag{3.21}$$

The natural prior $\mathbb{P}(u_i = \mathtt{1}) = q$ mentioned above means that $L^{(r)}_{i\to t}$ should be initialized as $L^{(0)}_{i\to t} = \ln\left(\frac{q}{1-q}\right)$. Then the item-to-test updates in subsequent

rounds easily follow from (3.18):

$$L_{i\to t}^{(r+1)} = \ln\left(\frac{q}{1-q}\right) + \sum_{t'\in\mathcal{N}(i)\setminus t} \widehat{L}_{t'\to i}^{(r)}. \tag{3.22}$$

The test-to-item messages require a bit more effort to derive, but the analysis is entirely elementary. If the test $t$ is positive ($y_t = 1$), we obtain [171]

$$\widehat{L}_{t\to i}^{(r)} = \ln\left(\vartheta + \frac{1-\vartheta}{1-(1-\varphi)\prod\limits_{j\in\mathcal{N}(t)\setminus\{i\}}\left(\vartheta + \frac{1-\vartheta}{1+\exp(L_{j\to t}^{(r)})}\right)}\right),$$

and if the test $t$ is negative ($y_t = 0$), we simply have $\widehat{L}_{i\to i}^{(r)} = \ln\vartheta$ [171].

We are not aware of any works simplifying the messages (or their log-ratios) for general noise models. Since the binary symmetric noise model of Example 3.1 (with parameter $\rho$) is particularly widely-adopted, we also state such a simplification here without proof. If $y_t = 1$, then

$$\hat{m}_{t\to i}^{(r+1)}(u_i) \propto \begin{cases} \rho \displaystyle\prod_{i'\in N(t)\setminus\{i\}} \left(m_{i'\to t}^{(r)}(0) + m_{i'\to t}^{(r)}(1)\right) & u_i = 1 \\ \rho \displaystyle\prod_{i'\in N(t)\setminus\{i\}} \left(m_{i'\to t}^{(r)}(0) + m_{i'\to t}^{(r)}(1)\right) & \\ \qquad + (1-2\rho)\displaystyle\prod_{i'\in N(t)\setminus\{i\}} m_{i'\to t}^{(r)}(0) & u_i = 0, \end{cases}$$

while if $y_t = 0$, then

$$\hat{m}_{t\to i}^{(r+1)}(u_i) \propto \begin{cases} (1-\rho) \displaystyle\prod_{i'\in N(t)\setminus\{i\}} \left(m_{i'\to t}^{(r)}(0) + m_{i'\to t}^{(r)}(1)\right) & u_i = 1 \\ (1-\rho) \displaystyle\prod_{i'\in N(t)\setminus\{i\}} \left(m_{i'\to t}^{(r)}(0) + m_{i'\to t}^{(r)}(1)\right) & \\ \qquad - (1-2\rho)\displaystyle\prod_{i'\in N(t)\setminus\{i\}} m_{i'\to t}^{(r)}(0) & u_i = 0. \end{cases}$$

Here we found it more convenient to work directly with the messages rather than their log-ratios; the two are equivalent in the sense that either can be computed from the other.

In the case that $k$ is known exactly, instead of declaring $\hat{u}_i$ to be zero or one according to (3.16), one can sort the estimates of $\mathbb{P}(u_i = 1 \mid \mathbf{y})$ in decreasing order and declare the resulting top $k$ items to be the defective set. Moreover, while (3.16) amounts to declaring an item defective if the estimate of $\mathbb{P}(u_i = 1 \mid \mathbf{y})$ exceeds $\frac{1}{2}$, one could threshold at values other than $\frac{1}{2}$. This would be of interest, for example, in scenarios where false positives and false negatives in the reconstruction are not considered equally bad.

### 3.3.1 Related Monte Carlo decoding algorithms

A distinct but related approach to belief propagation is based on generating samples from $\mathbb{P}(\mathcal{K} \mid \mathbf{y})$ via Markov Chain Monte Carlo (MCMC). To our knowledge, the MCMC approach to group testing was initiated by Knill *et al.* [123];

see also [169] and [84] for related follow-up works. Each of these papers uses the notion of Gibbs sampling: A randomly-initialized set $\mathcal{K}_0 \subseteq \{1, \dots, n\}$ is sequentially updated by choosing an item in $\{1, \dots, n\}$ (e.g., uniformly at random) and deciding whether it should be added or removed (or unchanged) from the set. Specifically, this decision is made based on a posterior calculation using Bayes rule, analogously to the belief propagation updates.

The Gibbs sampling procedure is designed to produce a Markov chain with stationary distribution $\mathbb{P}(\mathcal{K} \mid \mathbf{y})$, so that after sufficiently many iterations, the set being maintained is also approximately distributed according to $\mathbb{P}(\mathcal{K} \mid \mathbf{y})$. After taking numerous samples of sets from this distribution, the most commonly-occurring items are taken to be the final estimate $\hat{\mathcal{K}}$. Similarly to BP, a theoretical analysis of MCMC appears to be challenging, but the empirical performance is strong in simulations.

## 3.4 Noisy COMP

In Section 2.3, we discussed the analysis given by [33, 34] of the simple COMP algorithm in the noiseless case. In the same works, the authors also introduced a noisy version of COMP, which we refer to as NCOMP. The authors focused on the binary symmetric noise model (Example 3.1) with parameter $\rho \in \left(0, \frac{1}{2}\right)$, but the algorithm could also potentially be applied to other noise models.

The idea of NCOMP is that for any item $i \in \{1, \dots, n\}$, if the item is defective, then among the tests where $i$ is included, we should expect roughly a fraction $1 - \rho$ of the outcomes to be positive. In contrast, if the item is nondefective, we should expect a smaller fraction of the outcomes to be positive. Thus, the algorithm declares item $i$ to be defective or nondefective according to the following rule:

$$\text{Declare } i \text{ defective} \iff \frac{\sum_{t=1}^{T} \mathbf{1}\{X_{ti} = 1 \cap y_t = 1\}}{\sum_{t=1}^{T} \mathbf{1}\{X_{ti} = 1\}} \geq 1 - \rho(1 + \Delta) \tag{3.23}$$

for some parameter $\Delta > 0$. Note that this rule requires knowledge of the noise level $\rho$.

It was shown in [33, 34] that with a suitable choice of $\Delta$, NCOMP achieves a positive rate for all $\alpha \in (0, 1)$, albeit generally far from the information-theoretic limits of Section 4.5. The rates presented in [33, 34] differ according to the choice of $\nu > 0$, but as discussed in [167, Footnote 3], the best rate that can be ascertained *directly* from these works is

$$R_{\text{Bern}}^{\text{NCOMP}} = \frac{(1 - 2\rho)^2(1 - \alpha)}{4.36(1 + \sqrt{\alpha})^2}, \tag{3.24}$$

and amounts to choosing $\nu = 1$. This rate and the other relevant rates will be compared visually in Section 3.7.

We provide only a high-level outline of the proof of (3.24), and refer the reader to [33, 34] for the details. The analysis separately characterizes the probability of a given defective wrongly being declared as nondefective (i.e., failing the threshold test in (3.23)) and a given nondefective wrongly being declared defective. The i.i.d. nature of the test matrix and noise permits a concentration of measure argument, from which it can be shown that both

of these error events decay exponentially in $T$ as long as $\Delta$ is not too high. Applying the union bound leads to a multiplication of the preceding probabilities by $k$ and $n-k$ respectively, and the analysis is completed by choosing $T$ large enough to make the resulting bound decay to zero, as well as optimizing $\Delta$ and $\nu$.

## 3.5 Separate decoding of items

The NCOMP algorithm described above decodes each item individually: The decision on whether or not item $i$ is defective is based only on the $i$-th column of $\mathsf{X}$, along with $\mathbf{y}$. This general principle of decoding items separately was in fact introduced in an early work of Malyutov and Mateev [145], and shown to come with strong theoretical guarantees in the case that $k = O(1)$. It was originally referred to as *separate testing of inputs*, but we adopt the terminology *separate decoding of items* to avoid confusion with the idea of tests that contain only a single item.

Again, recall that $u_i = \mathbf{1}\{i \in \mathcal{K}\}$ indicates whether or not item $i$ is defective. The decoding rule for item $i$ proposed in [145] is as follows:

$$\text{Declare } i \text{ defective} \iff \sum_{t=1}^{T} \log_2 \frac{P_{Y|X_i,U_i}(y_t|x_{ti},1)}{P_Y(y_t)} \geq \gamma \tag{3.25}$$

where $\gamma > 0$ is a threshold. This can be interpreted as the Neyman-Pearson test for binary hypothesis testing with hypotheses $H_0 \colon u_i = 0$ and $H_1 \colon u_i = 1$; note that $P_{Y|X_i,U_i}(y_t|x_{ti},0)$ is the same as $P_Y(y_t)$ regardless of the value of $x_{ti}$ (i.e., nondefective items do not impact the test outcome).

We briefly mention that the decoder (3.25), along with its analysis (outlined below), can be viewed as a simplified and computationally efficient counterpart to an intractable *joint* decoding rule based on thresholding. The latter is surveyed in Chapter 4 as a means to deriving information-theoretic achievability bounds. See also [125, 105] for further works comparing separate and joint decoding.

The results of [145] indicate the following somewhat surprising fact: When $k = O(1)$ and $n \to \infty$, the rate achieved by separate decoding of items for the noiseless model or binary symmetric noise model is *within an* $\ln 2$ *factor of the optimal (joint) decoder*. For instance, in the noiseless setting, a rate of $\ln 2 \approx 0.7$ bits/test is attained, thus being reasonably close to the optimal rate of one.

For more general noise models, under Bernoulli testing, a sufficient condition on the number of tests for vanishing error probability is [145]

$$T \geq \frac{\log_2 p}{I_1}(1 + o(1)),$$

where the *single-item mutual information* $I_1$ is defined as follows, with implicit conditioning on item 1 being defective, and $X_1$ denoting whether it was included in a given test that produced the outcome $Y$:

$$I_1 = I(X_1; Y). \tag{3.26}$$

In a follow-up work [142], similar results were shown when the rule (3.25) is replaced by a *universal* rule (one that does not depend on the noise distribution) based on the empirical mutual information.

In this monograph, we are primarily interested in the sparse regime $k = \Theta(n^\alpha)$, as opposed to the very sparse regime $k = O(1)$. Separate decoding of items was studied under the former setting in [167], with the main results for specific models including the following.

**Theorem 3.3.** *Consider the separate decoding of items technique under i.i.d. Bernoulli testing with parameter $p = \frac{\ln 2}{k}$ (i.e., $\nu = \ln 2$), with $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$. Then we have the following:*

- *Under the noiseless model, there exists a constant $c(\delta') > 0$ such that the rate*

$$R_{\mathrm{Bern}}^{\mathrm{SD}} = \max_{\delta' > 0} \min \left\{ (\ln 2)(1-\alpha)(1-\delta'), c(\delta') \frac{1-\alpha}{\alpha} \right\} \tag{3.27}$$

  *is achievable. In particular, as $\alpha \to 0$, the rate approaches $\ln 2$ bits/test.*

- *Under the binary symmetric noise model (3.1) with parameter $\rho \in \left(0, \frac{1}{2}\right)$, there exists a constant $c_\rho(\delta') > 0$ such that the rate*

$$R_{\mathrm{Bern}}^{\mathrm{SD}}(\rho) = \max_{\delta' > 0} \min \left\{ (\ln 2)(1 - H_2(\rho))(1-\alpha)(1-\delta'), c_\rho(\delta') \frac{1-\alpha}{\alpha} \right\} \tag{3.28}$$

  *is achievable. Hence, as $\alpha \to 0$, the rate approaches $(\ln 2)(1 - H_2(\rho))$.*

The quantities $c(\delta')$ and $c_\rho(\delta')$ are related to concentration bounds arising in the analysis, as we discuss in the proof outline below. Explicit expressions for these quantities can be found in [167], but they are omitted here since they are somewhat complicated and do not provide significant insight. For both the noiseless and symmetric noise models, in the limit as $\alpha \to 0$, the rate comes within a $\ln 2$ factor of the channel capacity, which cannot be exceeded by any group testing algorithm (see Theorem 3.1). In [125], characterizations of the mutual information $I_1$ in (3.26) were also given for a variety of other noisy group testing models.

**Overview of proof of Theorem 3.3** As stated following (3.25), the decoder for a given item performs a binary hypothesis test to determine whether the item is defective. As a result, analysing the error probability amounts to characterizing the probabilities of false positives and false negatives in the recovery.

We first consider false positives. Letting $i$ represent a nondefective item, and letting $\mathbf{X}_i = [X_{1i}, \dots, X_{Ti}]^T$ be the corresponding column of $\mathsf{X}$, the probability of being declared defective is

$$P_{\mathrm{fp}} = \sum_{\mathbf{x}_i, \mathbf{y}} \mathbb{P}(\mathbf{x}_i) \mathbb{P}(\mathbf{y}) \mathbf{1}\left\{ \sum_{t=1}^{T} \log_2 \frac{P_{Y|X_i,U_i}(y_i \mid x_{ti}, 1)}{P_Y(y_t)} \geq \gamma \right\} \tag{3.29}$$

$$\leq \sum_{\mathbf{x}_i, \mathbf{y}} \mathbb{P}(\mathbf{x}_i) \left( \prod_{t=1}^{T} P_{Y|X_i,U_i}(y_i \mid x_{ti}, 1) \right) 2^{-\gamma} \tag{3.30}$$

$$= 2^{-\gamma}, \tag{3.31}$$

where (3.29) uses the fact that the column $\mathbf{X}_i$ and test outcomes $\mathbf{Y}$ are independent when $i$ is nondefective; (3.30) follows by writing the sum of logarithms as the logarithm of a product and noting that $\mathbb{P}(\mathbf{y}) = \prod_{t=1}^{T} P_Y(y_t)$,

which means that the event in the indicator funtion can be re-arranged to $\mathbb{P}(\mathbf{y}) \leq \big(\prod_{t=1}^{T} P_{Y|X_i,U_i}(y_i \mid x_{ti}, 1)\big)2^{-\gamma}$; and (3.31) follows since we are summing a joint probability distribution over all of its values. Since there are $n-k$ nondefectives, we can use the union bound to conclude that for any $\delta > 0$, the choice

$$\gamma = \log_2 \frac{n-k}{\delta}$$

suffices to ensure that the probability of *any* false positives is at most $\delta$.

With this choice of $\gamma$, the probability of any given defective item $i$ being declared as nondefective is given by

$$P_{\text{fn}} = \mathbb{P}\bigg(\sum_{t=1}^{T} \log_2 \frac{P_{Y|X_i,U_i}(Y_t|X_{ti},1)}{P_Y(Y_t)} \leq \log_2 \frac{n-k}{\delta}\bigg). \tag{3.32}$$

Observe that the mean of the left-hand side inside the probability is exactly $TI_1$. Moreover, the probability itself is simply the lower tail probability of an i.i.d. sum, and hence, we should expect some degree of concentration around the mean. To see this more concretely, we note that as long as

$$T \geq \frac{\log_2 \frac{n-k}{\delta}}{I_1(1-\delta')} \tag{3.33}$$

for some $\delta' \in (0,1)$, we have

$$P_{\text{fn}} \leq \mathbb{P}\bigg(\sum_{t=1}^{T} \log_2 \frac{P_{Y|X_i,U_i}(Y_t \mid X_{ti},1)}{P_Y(Y_t)} \leq TI_1(1-\delta')\bigg), \tag{3.34}$$

which is the probability of an i.i.d. sum being a factor $1-\delta'$ below its mean.

In the very sparse regime $k = O(1)$, establishing the required concentration is straightforward – it suffices to apply Chebyshev's inequality to conclude that $P_{\text{fn}} \to 0$ for arbitrarily small $\delta'$. We can then apply a union bound over the $k$ defective items to deduce that the probability of *any* false negatives vanishes, and we readily deduce (3.5).

The sparse regime $k = \Theta(n^\alpha)$ is more challenging, and the choice of concentration inequality can differ depending on the specific noise model. We omit the details, which are given in [167], and merely state that in Theorem 3.3, the second result makes use of a general bound based on Bernstein's inequality, whereas the first result uses a sharper bound specifically tailored to the noiseless model.

## 3.6 Noisy (near-)definite defectives

We saw in Chapter 2 that the Definite Defectives (DD) algorithm (Algorithm 2.3) achieves the best known rates of any practical algorithm in the noiseless setting. As a result, there is substantial motivation for analogous algorithms in noisy settings. Here we present such an algorithm, developed by Scarlett and Johnson [168], which is suitable for noise models satisfying the noisy defective channel property (Definition 3.3), and is again practical in the sense of Section 2.1.

Under Bernoulli testing with parameter $\nu > 0$, the algorithm accepts two parameters $(\gamma_1, \gamma_2)$ and proceeds as follows:

1. For each $i \in \{1, \dots, n\}$, let $T_{\text{neg}}(i)$ be the number of negative tests in which item $i$ is included. In the first step, we construct the following set of items that are believed to be nondefective:

$$\widehat{\text{ND}} = \left\{ i : T_{\text{neg}}(i) > \frac{\gamma_1 T \nu}{k} \right\} \tag{3.35}$$

   for some threshold $\gamma_1$. The remaining items, $\widehat{\text{PD}} = \{1, \dots, n\} \setminus \widehat{\text{ND}}$, are believed to be 'possible defective' items.

2. For each $j \in \widehat{\text{PD}}$, let $\widetilde{T}_{\text{pos}}(j)$ be the number of positive tests that include item $j$ and no other item from $\widehat{\text{PD}}$. In the second step, we estimate the defective set as follows:

$$\hat{\mathcal{K}} = \left\{ i \in \widehat{\text{PD}} : \widetilde{T}_{\text{pos}}(i) > \frac{\gamma_2 T \nu \mathrm{e}^{-\nu}}{k} \right\} \tag{3.36}$$

   for some threshold $\gamma_2$.

In the noiseless case, setting $\gamma_1 = \gamma_2 = 0$ recovers the standard DD algorithm, Algorithm 2.3. For the addition noise model (Example 3.2), since negative test outcomes are perfectly reliable, one can set $\gamma_1 = 0$. Similarly, for the Z channel noise model Example 3.4, since positive test outcomes are perfectly reliable, one can set $\gamma_2 = 0$. In fact, one of the main goals of [168] was to show that these two noise models can behave quite differently in group testing despite corresponding to channels with the same Shannon capacity.

Using concentration of measure results, it is possible to give exponential tail bounds for error events corresponding to particular values of $\gamma_1$ and $\gamma_2$. By balancing these tail bounds, in certain cases [168] explicitly gives optimal values of these parameters, and deduces the associated achievable rates (whose expressions are omitted here).

The strongest results among those in [168] are for the addition noise model (Example 3.4), in which the achievability curve matches an algorithm-independent converse for Bernoulli testing for a wide range of $\alpha \in (0, 1)$. Various rates are also provided for the Z channel and symmetric noise models; see the following section for example plots for the latter case. For each of these models, the rate converges to the noiseless DD rate (Section 2.4) in the low noise limit. On the other hand, the convergence can be rather slow, with visible gaps remaining even for low noise levels.

## 3.7 Rate comparisons and numerical simulations

In this section, we compare the achievable rates of the algorithms considered throughout this chapter, as well as comparing the algorithms numerically. We focus here on the symmetric noise model (Definition 3.1), since it has received the most attention in the context of proving achievable rates for noisy group testing algorithms.

**Rate comparisons.** In Figure 3.2, we plot the achievable rates of NCOMP, separate decoding of items, and noisy DD with noise levels $\rho = 10^{-4}$ and $\rho = 0.11$. We optimize the Bernoulli testing parameter $\nu > 0$ separately for each design. We also plot information-theoretic achievability and converse bounds

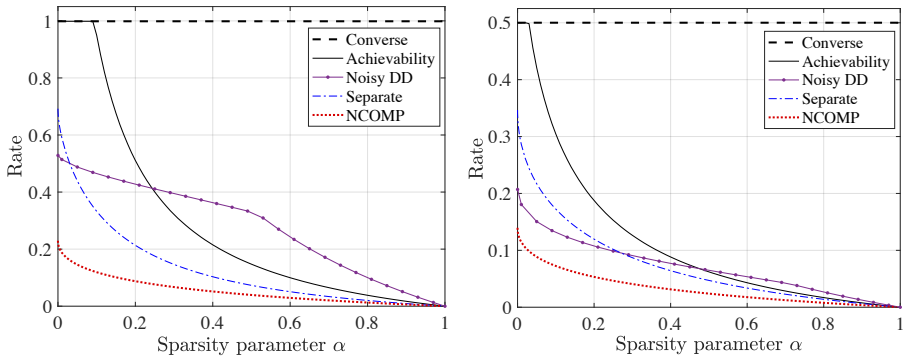



Figure 3.2: Achievable rates for the symmetric noise model with noise levels $\rho = 10^{-4}$ (Left) and $\rho = 0.11$ (Right). The converse and achievability curves correspond to information-theoretic limits given in Chapter 4.

to be presented in Chapter 4, with the achievability part corresponding to a computationally intractable decoding rule.

We observe that at least in this example, the rates for separate decoding of items and noisy DD are uniformly stronger than the rate proved for NCOMP. In the low noise case, noisy DD provides the best rate among the practical algorithms for most values of $\alpha$, but separate decoding of items provides a better rate for small $\alpha$. At the higher noise level, separate decoding of items provides a better rate over a wider range of $\alpha$, but noisy DD still dominates for most values of $\alpha$.

Overall, the rates for the noisy setting remain somewhat less well-understood than the noiseless setting (see Figure 2.1), and closing the remaining gaps remains an interesting direction for further research.

**Numerical simulations.** In Figure 3.3, we plot experimental simulation results under the symmetric noise model (Example 3.1) with parameter $\rho = 0.05$, and with $n = 500$ items and $k = 10$ defectives. We consider i.i.d. Bernoulli testing with parameter $\nu = \ln 2$, along with the following decoding rules:

- Noisy LP as described in Section 3.2, with parameter $\zeta = 0.5$ and each $u_i$ rounded to the nearest integer in $\{0, 1\}$;
- Belief propagation (BP) as described in Section 3.3, with 10 message passing iterations;
- NCOMP as described in Section 3.4, with $\Delta = \frac{0.1(1-2\rho)}{\rho}$ based on the theoretical choice in [33] along with some manual tuning of the constant factor;
- Separate decoding of items as described in Section 3.5, with $\gamma = (1 - \delta)I(X_1; Y)$ in accordance with the theoretical analysis, and $\delta$ chosen based on manual tuning to be $\frac{1}{3}$;
- Noisy DD as described in Section 3.6, with parameters $\gamma_1 = \gamma_2 = 0.175$ based on manual tuning.

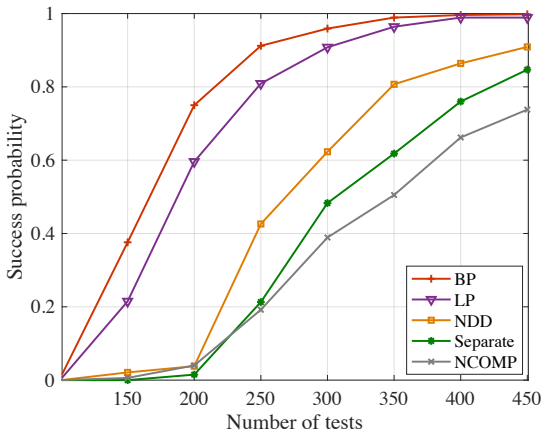



Figure 3.3: Experimental simulations for the symmetric noise model under Bernoulli testing with parameter $\nu = \ln 2$, with $n = 500$ items, $k = 10$ defectives, and noise parameter $\rho = 0.05$.

We observe that BP performs best, followed closely by LP. There is then a larger gap to NDD and separate decoding, and finally NCOMP requires the most tests. While our experiments are far from being an exhaustive treatment, these results indicate somewhat of a gap between the current theory and practice, with the best-performing methods (BP and LP) also being the least well-understood from a theoretical point of view. Closing this gap remains an interesting direction for further research.

To better understand the impact of knowledge of $k$ in noisy group testing, in Figure 3.4, we repeat the experiment with 'oracle' versions of the algorithms for the case that the number of defectives $k$ is known:

- Noisy LP includes the additional constraint that the estimates of $u_i$ sum to $k$;
- Instead of thresholding, BP chooses the $k$ items with the highest estimated probabilities of being defective.
- NCOMP takes the $k$ items for which the proportions of positive tests (relative to those the item is included in) are highest;
- Separate decoding of items chooses the $k$ items with the highest sum of log-probability ratios (see (3.25));
- Noisy DD estimates the 'possible defectives' to be the set of $(1 + \Delta)k$ items in the lowest number of negative tests, where we set $\Delta = \frac{1}{2}$ based on manual tuning. The algorithm then estimates the defective set to be the set of $k$ items with the highest number of positive tests in which it is the unique possible defective.

We observe that knowledge of $k$ brings the performance of NCOMP, separate decoding, and noisy DD closer together, but generally maintains their relative order. On the other hand, the performance of LP improves more than that of BP, making it become the best performing algorithm for most values of $T$.

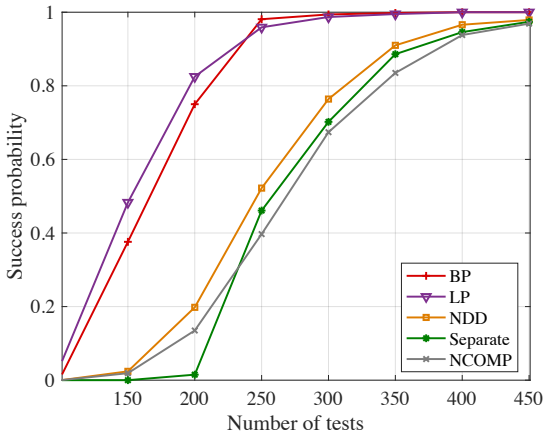



Figure 3.4: Performance of oracle versions of the respective algorithms under the same setup as that of Figure 3.3.

# Chapter 4

# Information-Theoretic Limits

In this chapter, we present information-theoretic achievability and converse bounds characterizing the fundamental limits of group testing regardless of the computational complexity. We have already seen a few converse results in the previous chapters, including the counting bound (Theorem 1.1) in the noiseless setting, and a capacity-based bound for noisy settings (Theorem 3.1).

The main results presented in this chapter are as follows:

- an achievable rate for the noiseless setting under Bernoulli testing, which matches or improves on all the algorithms considered in Chapter 2 (see the discussion in Section 4.1 and the details in Section 4.2);
- a matching converse bound for the noiseless setting establishing the exact maximum achievable rate of nonadaptive testing with a Bernoulli design (Section 4.3);
- an improved achievable rate for the noiseless setting under a near-constant column weight design (see the discussion in Section 4.1 and the details in 4.4);
- analogous achievability and converse bounds for noisy settings under the Bernoulli design, and applications to specific models (Section 4.5).

## 4.1 Overview of the standard noiseless model

Two major results in this chapter give achievable rates for noiseless nonadaptive group testing with two different designs. Theorem 4.1, due to Scarlett and Cevher [165, 163], concerns the Bernoulli design (see Definition 2.2), and Theorem 4.2, due to Coja-Oghlan *et al.* [46], concerns the near-constant column weight design (see Definition 2.3).

**Theorem 4.1.** *Consider noiseless nonadaptive group testing, under the exact recovery criterion in the small error setting, and* $k = \Theta(n^\alpha)$ *defectives with*

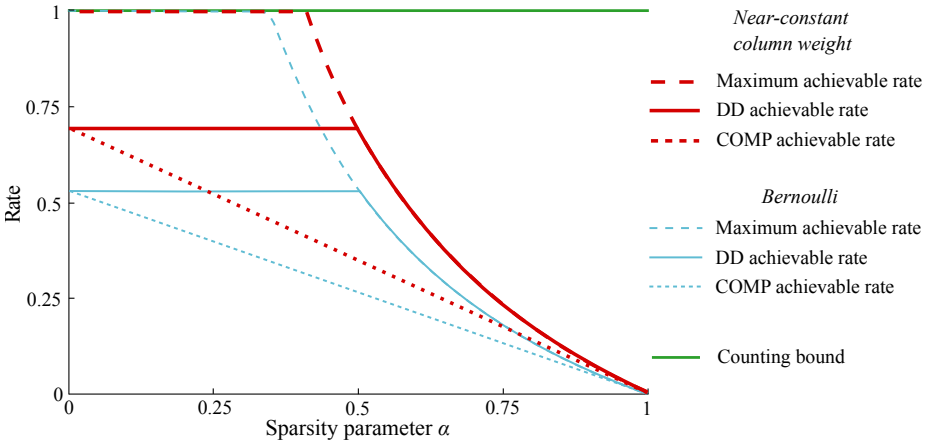



Figure 4.1: Rate for nonadaptive group testing in the sparse regime with a Bernoulli design and with a near-constant column weight designs.

$\alpha \in (0,1)$. *Then the rate*

$$\overline{R}_{\mathrm{Bern}} = \max_{\nu>0} \min \left\{ h(\mathrm{e}^{-\nu}),\ \frac{\nu \mathrm{e}^{-\nu}}{\ln 2} \frac{1-\alpha}{\alpha} \right\} \tag{4.1}$$

*is achievable, and can be achieved by a Bernoulli test design.*

**Theorem 4.2.** *Consider noiseless nonadaptive group testing, under the exact recovery criterion in the small error setting, and $k = \Theta(n^\alpha)$ defectives with $\alpha \in (0,1)$. Then the rate*

$$\overline{R}_{\mathrm{NCC}} = \min \left\{ 1, (\ln 2) \frac{1-\alpha}{\alpha} \right\} \tag{4.2}$$

*is achievable, and can be achieved by a near-constant column weight design with $\nu = \ln 2$.*

The results are shown in Figure 4.1, which is a repeat of Figure 2.1 included here for convenience. We see that for $\alpha \leq 1/3$, both theorems give an equal rate of 1, while for $\alpha > 1/3$, the rate (4.2) for near-constant column weight designs is slightly higher than the rate (4.1) for Bernoulli designs, in particular equalling 1 for $\alpha \leq 0.409$.

Theorem 4.1 is proved in Section 4.2 using information-theoretic methods akin to those used in studies of channel coding. We dedicate a large section of this chapter to the study of this proof, as the information theory approach is a powerful and flexible method that can be applied to other sparse inference problems (see [165]), and in particular to noisy group testing models (see Section 4.5). The proof of Theorem 4.2 uses a more direct probabilistic method to show that there exists only one satisfying set (see Definition 2.1) with high probability – arguably a simpler strategy, but one that may be harder to generalise to other models. We discuss this proof in Section 4.4.

The rate expression in (4.1) is a little complicated. It will become apparent in the forthcoming proof of Theorem 4.1 that the parameter $\nu$ enters through

the choice of the Bernoulli parameter as $p = \nu/k$. It is easy to see that the first minimand of (4.1) is maximized at $\nu = \ln 2$, and is the value of $p$ that corresponds to (asymptotically) half of the tests being positive. By differentiation, we see that the second minimand of (4.1) is maximized at $\nu = 1$, which corresponds to $p = 1/k$, and is the value of $p$ that corresponds to an average of one defective per test. Using these findings, we can check that the following simplification of (4.1) holds:

$$\overline{R}_{\mathrm{Bern}} = \begin{cases} 1 & \text{for } \alpha \le 1/3 \\ \text{as in (4.1)} & \text{for } 1/3 < \alpha < 0.359 \\ 0.531\,\dfrac{1-\alpha}{\alpha} & \text{for } \alpha \ge 0.359, \end{cases}$$

where it should be understood that the decimal values are non-exact (rounded to three decimal places).

The near-constant column weight design with $L = \nu T/k$ tests per item is always optimized with $\nu = \ln 2$, which corresponds to (asymptotically) half of the tests being positive. This makes the expression (4.2) simpler, and we have

$$\overline{R}_{\mathrm{NCC}} = \begin{cases} 1 & \text{for } \alpha \le 0.409, \\ 0.693\,\dfrac{1-\alpha}{\alpha} & \text{for } \alpha > 0.409. \end{cases}$$

Thus, we see that nonadaptive group testing achieves the rate 1 of the counting bound (see Section 1.4) and has the same rate as adaptive testing for $\alpha \le 1/3$ with a Bernoulli design and for $\alpha \le 0.409$ with a near-constant column weight design. On the other hand, for $\alpha$ above these thresholds, the rates are strictly below the counting bound. In fact, Theorems 4.1 and 4.2 provide the best possible rates for their respective designs (see Sections 4.3 and 4.4), meaning nonadaptive testing with these designs is provably worse than adaptive testing in these regimes, since in the latter setting the counting bound is achievable (see Section 1.5).

Before continuing, we briefly review work on achievable rates for noiseless nonadaptive group testing that preceded Theorems 4.1 and 4.2 (although these papers did not necessarily phrase their results this way). We begin with results using Bernoulli test designs.

Freidlina [81] and Malyutov [143] showed that a rate of 1 is achievable in the very sparse regime where $k$ is constant as $n \to \infty$. Malyutov used an information-theoretic approach based on a multiple access channel model with one input for each defective item. Sebő [170] also attained a rate of 1 for constant $k$ using a more direct probabilistic method.

Atia and Saligrama [17] reignited interest in the use of information-theoretic methods for studying group testing. They used a model of channel coding with correlated codewords, where each potential defective set is a message (recall the channel coding interpretation of group testing shown in Figure 1.1). Atia and Saligrama showed that, in the limiting regime where $k \to \infty$ after $n \to \infty$, one can succeed with $T = O(k \log n)$ tests, although they did not specify the implicit constant. Effectively, in our notation, this shows a nonzero rate for $\alpha = 0$, but does not prove the Freidlina–Malyutov–Sebő rate of 1. They also showed that $T = O(k \log n \log^2 k)$ suffices for any $k = o(n)$, though this falls short of proving a nonzero rate for $\alpha \in (0, 1)$. They also gave order-wise results for some noisy

models. A similar approach to Atia and Saligrama was taken by Scarlett and Cevher [165] (outlined below), but with tighter analysis and careful calculation of constants giving the better rates of Theorem 4.1.

Aldridge, Baldassini, and Gunderson [11] generalized Sebő's approach to all $\alpha \in [0, 1)$, showing a nonzero rate for all $\alpha$ that achieves the rate of 1 at $\alpha = 0$, but that is suboptimal compared to (4.1) for $\alpha \in (0, 1)$.

In Chapter 2 of this monograph, we saw some rates that can be achieved with practical algorithms. Chan *et al.* [33, 34] were the first to show a nonzero rate for all $\alpha \in (0, 1)$, albeit one that is suboptimal compared to (4.1), by analysing the COMP algorithm (Theorem 2.3). They also showed nonzero rates for some non-Bernoulli designs. The DD algorithm of Aldridge, Baldassini, and Johnson [12] also achieves nonzero rates for all $\alpha \in (0, 1)$ with Bernoulli testing, in particular matching (4.1) for $\alpha > 1/2$ (Theorem 2.4).

We also saw in Section 2.7 that the performance of these algorithms is improved when used with the near-constant column weight design. In particular, the DD algorithm achieves the same rate as (4.2) for $\alpha > 1/2$, as shown by Johnson, Aldridge and Scarlett [113]. We direct the reader back to Chapter 2 for detailed discussions of these results and other algorithms.

Mézard, Tarzia and Toninelli [148] had suggested that Theorem 4.2 should be true by appealing to heuristics from statistical physics – the innovation of Coja-Oghlan *et al.* [46] was to prove this rigorously.

## 4.2 Proof of achievable rate for Bernoulli testing

### 4.2.1 Discussion of proof techniques

Our proof follows Scarlett and Cevher [165], who proved Theorem 4.1 as a special case of a more general framework for noiseless and noisy group testing. The analysis is based on *thresholding techniques* that are rooted in early information-theoretic works, such as [76, 175], as well as recent developments in information-spectrum methods [99]. In fact, we also saw a simpler version of this approach when studying separate decoding of items in Section 3.5.

To describe these methods in more detail, we momentarily depart from the group testing problem and consider a simple channel coding scenario where $M$ codewords are drawn from some distribution $P_{\mathbf{X}}$, and one of them is transmitted over a channel $P_{\mathbf{Y}|\mathbf{X}}$ to produce an output sequence $\mathbf{y}$. The optimal (yet generally computationally intractable) decoding rule chooses the codeword $\mathbf{x}$ maximizing the likelihood $P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$, and the resulting error probability is upper bounded by the probability that the true codeword is the only one such that $\log_2 \frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})}{P_{\mathbf{Y}}(\mathbf{y})}$ exceeds a suitably-chosen threshold. Intuitively, we should expect $P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$ to be considerably larger than $P_{\mathbf{Y}}(\mathbf{y})$ when $\mathbf{x}$ is the true transmitted codeword, whereas if $\mathbf{x}$ is an incorrect codeword then this is unlikely to be the case.

More precisely, by a simple change of measure technique, the probability of $\log_2 \frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})}{P_{\mathbf{Y}}(\mathbf{y})}$ exceeding any threshold $\gamma$ for a single non-transmitted codeword is at most $2^{-\gamma}$, and hence the union of these events across all $M - 1$ non-transmitted codewords has probability at most $(M - 1)2^{-\gamma}$. Choosing $\gamma$ slightly larger than $\log_2 M$ ensures that this probability is small, and hence the

error probability is roughly the probability that the true codeword $\mathbf{x}$ fails the threshold test, i.e., $\log_2 \frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})}{P_{\mathbf{Y}}(\mathbf{y})} < \gamma \approx \log_2 M$.

Finally, for a memoryless channel taking the form $P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n P_{Y|X}(y_i|x_i)$, and an i.i.d. codeword distribution of the form $P_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^n P_X(x_i)$, the quantity $\log_2 \frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})}{P_{\mathbf{Y}}(\mathbf{y})}$ concentrates about its mean $nI(X;Y)$. As a result, we get vanishing error probability when the number of codewords satisfies $M \lesssim 2^{nI(X;Y)}$, and we can achieve any coding rate up to the mutual information $I(X;Y)$.

For group testing, we follow the same general idea, but with a notable change: the 'codewords' (that is, the $T \times k$ sub-matrices $\mathsf{X}_{\mathcal{K}}$ of the test matrix for $\mathcal{K}$ of cardinality $k$) are not independent. For example, the codewords corresponding to $\mathcal{K}_1 = \{1,2,3\}$ and $\{\mathcal{K}_2\} = \{1,4,7\}$ have a common first column. To handle this issue, we treat different incorrect codewords separately depending on their amount of overlap with the true codeword: If there is no overlap then the analysis is similar to that of channel coding above, while if there is overlap then we consider probabilities of the form $\log_2 \frac{P_1(\mathbf{y}|\cdot)}{P_2(\mathbf{y}|\cdot)}$, where $P_1$ conditions on the true codeword, and $P_2$ only conditions on the overlapping part.

We now proceed with the proof of Theorem 4.1. We first introduce some notation that will allow the initial steps to be re-used for the noisy setting, then formally specify the decoder used, provide the non-asymptotic bound that forms the starting point of the analysis, and finally, outline the subsequent asymptotic analysis that leads to the final result.

### 4.2.2 Information-theoretic notation

While our focus is primarily on the noiseless setting, the initial steps of the analysis are just as easily done simultaneously for general noise models. Specifically, we consider an arbitrary model studying the 'only defects matter' property, given in Definition 3.2. Due to this property and the symmetry in the random construction of $\mathsf{X}$, the analysis will not be impacted by the realization of $\mathcal{K}$, and we will therefore set $\mathcal{K} = \{1,\dots,k\}$ without loss of generality.

We now introduce some notation. We again consider the Bernoulli design (Definition 2.2), in which each item is included in each test independently with probability $p$. For convenience, we write $p = \nu/k$ for some $\nu > 0$, and as usual the $T \times n$ test matrix is denoted by $\mathsf{X}$.

The submatrix $\mathsf{X}_{\mathcal{K}}$ denotes only the columns of the matrix $\mathsf{X}$ indexed by $\mathcal{K}$, and $\mathbf{X}_{\mathcal{K}}$ denotes a single row of $\mathsf{X}_{\mathcal{K}}$. We write $V = V(\mathbf{X}_{\mathcal{K}})$ for the random number of defective items in the test indicated by $\mathbf{X}$.

The observation $Y \in \{0,1\}$ is generated according to some general distribution $P_{Y|\mathbf{X}_{\mathcal{K}}}$ depending on $\mathbf{X}_{\mathcal{K}}$ only through $V$:

$$(Y \mid \mathbf{X}, \mathcal{K}) \sim P_{Y|\mathbf{X}_{\mathcal{K}}} = P_{Y|V}. \tag{4.3}$$

This is precisely the only defects matter property of Definition 3.2. The $T$-fold product of $P_{Y|\mathbf{X}_{\mathcal{K}}}$ gives the distribution of the overall test vector $\mathbf{Y} = (Y_1,\dots,Y_T)$ given $\mathsf{X}_{\mathcal{K}}$, and is denoted by $P_{\mathbf{Y}|\mathsf{X}_{\mathcal{K}}}$.

As discussed above, we consider separate error events according to how much an incorrect defective set $\mathcal{K}'$ overlaps with $\mathcal{K}$. To facilitate this, for a given partition $(S_0, S_1)$ of $\mathcal{K}$, we write

$$P_{Y|\mathbf{X}_{S_0},\mathbf{X}_{S_1}}(y \mid \mathbf{x}_{S_0}, \mathbf{x}_{S_1}) = P_{Y|\mathbf{X}_{\mathcal{K}}}(y \mid \mathbf{x}_{\mathcal{K}}), \tag{4.4}$$

and use this to define the marginal distribution

$$P_{Y|\mathbf{X}_{S_1}}(y|\mathbf{x}_{S_1}) = \sum_{\mathbf{x}_{S_0}} P_{\mathbf{X}_{S_0}}(\mathbf{x}_{S_0}) P_{Y|\mathbf{X}_{S_0},\mathbf{X}_{S_1}}(y \mid \mathbf{x}_{S_0}, \mathbf{x}_{S_1}), \tag{4.5}$$

where $(\mathbf{x}_{S_0}, \mathbf{x}_{S_1}, y)$ is a specific realization of $(\mathbf{X}_{S_0}, \mathbf{X}_{S_1}, Y)$. In the analysis, $S_1$ will represent the intersection $\mathcal{K} \cap \mathcal{K}'$ between the defective set $\mathcal{K}$ and some incorrect set $\mathcal{K}'$, whereas $S_0$ will represent the set difference $\mathcal{K} \setminus \mathcal{K}'$.

Finally, in accordance with the techniques outlined in Section 4.2.1, we define the *information density*

$$\imath(\mathbf{X}_{S_0}; Y \mid \mathbf{X}_{S_1}) = \log_2 \frac{P_{Y|\mathbf{X}_{S_0},\mathbf{X}_{S_1}}(y \mid \mathbf{X}_{S_0}, \mathbf{X}_{S_1})}{P_{Y|\mathbf{X}_{S_1}}(Y \mid \mathbf{X}_{S_1})}, \tag{4.6}$$

and let $\imath^T(\mathsf{X}_{S_0}; \mathbf{Y}|\mathsf{X}_{S_1})$ be the $T$-letter extension obtained by summing (4.6) over the $T$ tests. Since the tests are independent, writing the sum of logarithms as the logarithm of a product yields

$$\imath^T(\mathsf{X}_{S_0}; \mathbf{Y} \mid \mathsf{X}_{S_1}) = \log_2 \frac{P_{\mathbf{Y}|\mathsf{X}_{S_0},\mathsf{X}_{S_1}}(\mathbf{Y} \mid \mathsf{X}_{S_0}, \mathsf{X}_{S_1})}{P_{\mathbf{Y}|\mathsf{X}_{S_1}}(\mathbf{Y} \mid \mathsf{X}_{S_1})}. \tag{4.7}$$

We also note that the expectation of (4.6) is equal to the conditional mutual information $I(\mathbf{X}_{S_0}; Y \mid \mathbf{X}_{S_1})$, and the expectation of (4.7) is equal to $T \cdot I(\mathbf{X}_{S_0}; Y \mid \mathbf{X}_{S_1})$.

### 4.2.3 Choice of decoder

Inspired by classical information-theoretic works such as [76] (again see Section 4.2.1), we consider a decoder that searches for a defective set $\mathcal{K} \subseteq \{1, \ldots, n\}$ of cardinality $k$ such that

$$\imath^T(\mathsf{X}_{S_0}; \mathbf{Y} \mid \mathsf{X}_{S_1}) > \gamma_{|S_0|} \quad \text{for all } (S_0, S_1) \text{ partitioning } \mathcal{K} \text{ with } S_0 \neq \varnothing \tag{4.8}$$

for suitable constants $\gamma_1, \ldots, \gamma_K$ to be chosen later. If no such set exists, or if multiple sets exist, then an error is declared.

The rule (4.8) can be viewed as a weakened version of the maximum-likelihood (ML) rule – that is, the decoder that chooses the set $\mathcal{K}$ maximizing $P_{\mathbf{Y}|\mathsf{X}_{\mathcal{K}}}$. Specifically, if a unique set satisfies (4.8), it must be the ML choice, whereas sometimes the ML decoder might succeed where the above decoder fails – for example, in cases where no $\mathcal{K}$ passes all $2^k - 1$ of its threshold tests.

The above decoder is unlikely to be computationally feasible in practice even for moderate problem sizes. The focus in this section is on information-theoretic achievability regardless of such considerations. Moreover, while this rule requires knowledge of $k$, we argue in Section 4.3 that at least in the noiseless setting, the resulting rate can be achieved even without this knowledge.

### 4.2.4 Non-asymptotic bound

Observe that in order for an error to occur, it must be the case that either the true defective set $\mathcal{K} = \{1, \ldots, k\}$ fails one of the threshold tests in (4.8), or some

incorrect set $\mathcal{K}'$ passes all of the threshold tests. As a result, the union bound gives

$$\begin{aligned}\mathbb{P}(\text{err}) \le \mathbb{P}\Big( \bigcup_{(S_0,S_1)} \big\{ \imath^T(\mathsf{X}_{S_0}; \mathbf{Y} \mid \mathsf{X}_{S_1}) \le \gamma_{|S_0|} \big\} \Big) \\ + \sum_{\mathcal{K}' \neq \mathcal{K}} \mathbb{P}\Big( \imath^T(\mathsf{X}_{\mathcal{K}' \setminus \mathcal{K}}; \mathbf{Y} \mid \mathsf{X}_{\mathcal{K} \cap \mathcal{K}'}) > \gamma_{|\mathcal{K}' \setminus \mathcal{K}|} \Big),\end{aligned} \tag{4.9}$$

where in the first term the union is implicitly subject to the conditions in (4.8), and in the second term, we upper bound the probability of passing all threshold tests by the probability of passing a single one (namely, the one with $S_1 = \mathcal{K} \cap \mathcal{K}'$).

Using the form of $\imath^T$ in (4.7), we can upper bound any given summand of (4.9) as follows with $S_0 = \mathcal{K}' \setminus \mathcal{K}$, $S_1 = \mathcal{K} \cap \mathcal{K}'$, and $\tau = |\mathcal{K}' \setminus \mathcal{K}|$:

$$\mathbb{P}\Big( \imath^T(\mathsf{X}_{\mathcal{K}' \setminus \mathcal{K}}; \mathbf{Y} \mid \mathsf{X}_{\mathcal{K} \cap \mathcal{K}'}) > \gamma_\tau \Big) \tag{4.10}$$

$$= \sum_{\mathsf{x}_{S_0}, \mathsf{x}_{S_1}, \mathbf{y}} \mathbb{P}(\mathsf{X}_{S_0} = \mathsf{x}_{S_0}, \mathsf{X}_{S_1} = \mathsf{x}_{S_1}) P_{\mathbf{Y}|\mathsf{X}_{S_1}}(\mathbf{y} \mid \mathsf{x}_{S_1}) \times \mathbf{1}\Big\{ \log_2 \frac{P_{\mathbf{Y}|\mathsf{X}_{S_0},\mathsf{X}_{S_1}}(\mathbf{y} \mid \mathsf{x}_{S_0}, \mathsf{x}_{S_1})}{P_{\mathbf{Y}|\mathsf{X}_{S_1}}(\mathbf{y} \mid \mathsf{x}_{S_1})} > \gamma_\tau \Big\} \tag{4.11}$$

$$\le \sum_{\mathsf{x}_{S_0}, \mathsf{x}_{S_1}, \mathbf{y}} \mathbb{P}(\mathsf{X}_{S_0} = \mathsf{x}_{S_0}, \mathsf{X}_{S_1} = \mathsf{x}_{S_1}) P_{\mathbf{Y}|\mathsf{X}_{S_0},\mathsf{X}_{S_1}}(\mathbf{y} \mid \mathsf{x}_{S_0}, \mathsf{x}_{S_1}) 2^{-\gamma_\tau} \tag{4.12}$$

$$= 2^{-\gamma_\tau}. \tag{4.13}$$

Here, (4.11) follows since the observations depend on $\mathsf{X}_{\mathcal{K}'}$ only through the columns $S_1 = \mathcal{K} \cap \mathcal{K}'$ overlapping with $\mathcal{K}$, (4.12) follows by upper bounding $P_{\mathbf{Y}|\mathsf{X}_{S_1}}(\mathbf{y} \mid \mathsf{X}_{S_1})$ according to the event in the indicator function and then upper bounding the indicator function by one, and (4.13) follows from the fact that we are summing a joint distribution over all of its values.

Combining (4.9) and (4.13), and also applying the union bound in the first term of the former, we obtain

$$\mathbb{P}(\text{err}) \le \sum_{(S_0,S_1)} \mathbb{P}\Big( \imath^T(\mathsf{X}_{S_0}; \mathbf{Y} \mid \mathsf{X}_{S_1}) \le \gamma_{|S_0|} \Big) + \sum_{\mathcal{K}' \neq \mathcal{K}} 2^{-\gamma_{|\mathcal{K}' \setminus \mathcal{K}|}}.$$

where we have applied the definition $\tau = |\mathcal{K}' \setminus \mathcal{K}|$. By counting the number of $S_0 \subset \mathcal{K}$ of cardinality $\tau \in \{1, \dots, k\}$, as well as the number of $\mathcal{K}' \neq \mathcal{K}$ such that $|\mathcal{K}' \setminus \mathcal{K}| = \tau \in \{1, \dots, k\}$, we can simplify the above bound to

$$\mathbb{P}(\text{err}) \le \sum_{\tau=1}^{k} \binom{k}{\tau} \mathbb{P}\Big( \imath^T(\mathsf{X}_{0,\tau}; \mathbf{Y} \mid \mathsf{X}_{1,\tau}) \le \gamma_\tau \Big) + \sum_{\tau=1}^{k} \binom{k}{\tau} \binom{n-k}{\tau} 2^{-\gamma_\tau}, \tag{4.14}$$

where $\mathsf{X}_{0,\tau} = \mathsf{X}_{S_0}$ and $\mathsf{X}_{1,\tau} = \mathsf{X}_{S_1}$ for an arbitrary partition $(S_0, S_1)$ of $\{1, \dots, k\}$ with $|S_0| = \tau$; by the i.i.d. test design and model assumption (4.3), the probability in (4.14) is the same for any such partition.

Finally, choosing

$$\gamma_\tau = \log_2 \frac{\delta}{k \binom{k}{\tau} \binom{n-k}{\tau}}$$

for some $\delta > 0$, we obtain the non-asymptotic bound

$$\mathbb{P}(\text{err}) \le \sum_{\tau=1}^{k} \binom{k}{\tau} \mathbb{P}\left( \imath^T(\mathsf{X}_{0,\tau}; \mathbf{Y} \mid \mathsf{X}_{1,\tau}) \le \log_2 \frac{\delta}{k\binom{k}{\tau}\binom{n-k}{\tau}} \right) + \delta. \tag{4.15}$$

### 4.2.5 Characterizing the tail probabilities

The next step is to characterize the probability appearing on the right-hand side of (4.15). The idea is to note that this is the tail probability of an i.i.d. sum, and hence we should expect some concentration around the mean. Recall from Section 4.2.2 that the mean of the information density is the conditional mutual information:

$$\mathbb{E}\big[\imath^T(\mathsf{X}_{0,\tau}; \mathbf{Y} \mid \mathsf{X}_{1,\tau})\big] = T \cdot I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau}) =: T \cdot I_\tau, \tag{4.16}$$

where $(\mathbf{X}_{0,\tau}, \mathbf{X}_{1,\tau})$ correspond to single rows in $(\mathsf{X}_{0,\tau}, \mathsf{X}_{1,\tau})$, and $Y$ is the corresponding entry of $\mathbf{Y}$. The following lemma characterizes $I_\tau$ for the noiseless model; we return to the noisy setting in Section 4.5.

**Lemma 4.1.** *Under the noiseless group testing model using Bernoulli testing with probability $p = \nu/k$ for some fixed $\nu > 0$, the conditional mutual information $I_\tau$ behaves as follows as $k \to \infty$:*

1. *If $\tau/k \to 0$, then*
$$I_\tau \sim \mathrm{e}^{-\nu} \nu \frac{\tau}{k} \log_2 \frac{k}{\tau}.$$
2. *If $\tau/k \to \psi \in (0, 1]$, then*
$$I_\tau \sim \mathrm{e}^{-(1-\psi)\nu} h(\mathrm{e}^{-\psi\nu}),$$
*where $h(\psi)$ is the binary entropy function.*

*Proof.* In the noiseless setting, we have $I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau}) = H(Y \mid \mathbf{X}_{1,\tau})$. If $\mathbf{X}_{1,\tau}$ contains any ones, then the conditional entropy of $Y$ is zero, and otherwise, the conditional entropy is the binary entropy function evaluated at the conditional probability of $Y = 1$. Evaluating these probabilities explicitly, we obtain

$$I_\tau = (1-p)^{k-\tau} h\big((1-p)^\tau\big) = \Big(1 - \frac{\nu}{k}\Big)^{k-\tau} h\bigg(\Big(1 - \frac{\nu}{k}\Big)^\tau\bigg).$$

In the case that $\tau/k \to 0$, the lemma now follows from the asymptotic expressions $(1-\nu/k)^{k-\tau} \to \mathrm{e}^{-\nu}$ and $(1-\nu/k)^\tau \sim 1 - \nu\tau/k$, as well as $h(1-\zeta) \sim -\zeta \log_2 \zeta$ as $\zeta \to 0$.

In the case that $\tau/k \to \psi \in (0, 1]$, the lemma follows from the limits $(1 - \nu/k)^{k-\tau} \to \mathrm{e}^{-(1-\psi)\nu}$ and $(1 - \nu/k)^\tau \to \mathrm{e}^{-\psi\nu}$, as well as the continuity of entropy. □

We now fix a set of constants $\delta'_\tau$ for $\tau = 1, \dots, k$, and observe that as long as

$$T \ge \frac{\log_2 \binom{n-k}{\tau} + \log_2 \left(\frac{k}{\delta}\binom{k}{\tau}\right)}{(1-\delta'_\tau) I_\tau}, \tag{4.17}$$

we can upper bound the probability in (4.15) by

$$\mathbb{P}\big(\imath^T(\mathsf{X}_{0,\tau};\mathbf{Y}\mid\mathsf{X}_{1,\tau})<(1-\delta'_\tau)TI_\tau\big). \tag{4.18}$$

As mentioned above, $\imath^T$ is a sum of $T$ i.i.d. random variables having mean $I_\tau$, and as a result, we can bound (4.18) using concentration inequalities.

In fact, in the case that $k$ is constant (that is, not growing with $n$), it suffices to use Chebyshev's inequality to show that each term of the form (4.18) vanishes for arbitrarily small $\delta'_\tau$ [163]. Since each such term vanishes, then so does the weighted sum of all such terms in (4.15) (using the fact that $k$ is constant), and we are left only with the sufficient condition in (4.17) for $\mathbb{P}(\text{err})\le\delta+o(1)$. By taking $\delta\to 0$ sufficiently slowly, we are left with the condition

$$T\ge\max_{\tau=1,\dots,k}\frac{\log_2\binom{n-k}{\tau}}{I_\tau}(1+o(1))$$

for $\mathbb{P}(\text{err})\to 0$.

However, our main interest is not in the fixed-$k$ regime, but in the regime $k=\Theta(n^\alpha)$ for $\alpha\in(0,1)$. In this case, more sophisticated concentration bounds are needed, and these turn out to introduce extra requirements on $T$ beyond (4.17) alone.

**Lemma 4.2.** *Set $\tau^*=k/\sqrt{\log_2 k}$. Under Bernoulli group testing with probability $p=\nu/k$, the quantities $\imath_{\tau,T}:=\imath^T(\mathsf{X}_{0,\tau};\mathbf{Y}\mid\mathsf{X}_{1,\tau})$ satisfy the following concentration bounds provided that the quantities $\delta'_\tau$ are uniformly bounded away from zero and one:*

1. *For $\tau\le\tau^*$, we have*

$$\mathbb{P}\big(\imath_{\tau,T}<TI_\tau(1-\delta'_\tau)\big)\le\exp\bigg(-T\frac{\tau}{k}\mathrm{e}^{-\nu}\nu(1-\delta'_\tau)\log_2(1-\delta'_\tau)(1+o(1))\bigg).$$

2. *For $\tau>\tau^*$, we have*

$$\mathbb{P}\big(\imath_{\tau,T}<TI_\tau(1-\delta'_\tau)\big)\le 2\exp\bigg(-\frac{(\delta'_\tau I_\tau)^2T}{4(8+\delta'_\tau I_\tau)}\bigg).$$

*Proof.* The first bound is proved by lower bounding $i_{\tau,T}$ by a scaled binomial random variable and applying a well-known concentration bound specific to the binomial distribution. The second bound is proved using Bernstein's inequality. The details can be found in [165]. □

The remainder of the proof amounts to rather tedious yet elementary algebraic manipulations, and we therefore provide only an outline. We start by choosing choose $\delta'_\tau=1-\epsilon$ for $\tau\le\tau^*$, and $\delta'_\tau=\epsilon$ for $\tau>\tau^*$, where $\epsilon>0$ is arbitrarily small.

The first requirement on $T$ is that it satisfies (4.17) for all $\tau=1,\dots,k$. Using Lemma 4.1 and the preceding choices of $\delta'_\tau$, one can show that the value of $\tau$ that gives the most stringent requirement on $T$ is $\tau=k$, at least in the asymptotic limit. As a result, we get the condition

$$T\ge\Big(k\log_2\frac{n}{k}\Big)\big(1+O(\epsilon)+o(1)\big). \tag{4.19}$$

This arises from the fact that the numerator in (4.17) with $\tau = k$ is dominated by $\log_2 \binom{n-k}{k}$, which behaves as $\big(k \log_2 \frac{n}{k}\big)(1 + o(1))$ whenever $k = o(n)$.

The second requirement on $T$ is that, upon substituting the bounds of Lemma 4.2 into (4.15) and taking $\delta \to 0$, the resulting summation on the right-hand side vanishes. For this to be true, it suffices that both the summations over $\tau \in \{1, \dots, \tau^*\}$ and $\tau \in \{\tau^* + 1, \dots, k\}$ vanish. The second of these (the 'small overlap' case) turns out to already vanish under the condition in (4.19). On the other hand, after some rearranging and asymptotic simplifications, we find that the first of these summations (the 'large overlap' case) vanishes provided that

$$T \geq \left( \frac{\frac{\alpha}{1-\alpha} k \log_2 \frac{n}{k}}{\nu \mathrm{e}^{-\nu}} \right) \big(1 + O(\epsilon) + o(1)\big). \tag{4.20}$$

Theorem 4.1 follows by combining these bounds and taking $\epsilon \to 0$.

## 4.3 Converse bound for Bernoulli testing

We have seen that nonadaptive Bernoulli matrix designs achieve a rate of 1 bit per test whenever $\alpha \leq \frac{1}{3}$, thus matching the counting bound and proving their asymptotic optimality. On the other hand, for $\alpha \in \big(\frac{1}{3}, 1\big)$, there remains a gap between the two, with the gap growing larger as $\alpha$ approaches one.

A priori, there are several possible reasons for the remaining gaps: the analysis in Section 4.2 could be loose, the use of Bernoulli tests could be suboptimal, or the counting bound itself could be loose. The following result, due to Aldridge [7], rules out the first of these, showing that Theorem 4.1 provides the best rate that one could hope for given that Bernoulli designs are used.

**Theorem 4.3.** *Consider noiseless group testing in the sparse regime $k = \Theta(n^\alpha)$, with the exact recovery criterion, and Bernoulli testing. If the rate exceeds $\overline{R}_{\mathrm{Bern}}$ defined in* (4.1)*, then the error probability averaged over the testing matrix is bounded away from zero, regardless of the decoding algorithm.*

*Proof.* The idea of the proof is as follows. First, we argue that if both the COMP and SSS (see Chapter 2) algorithms fail, then any algorithm fails with a certain probability. Second, we argue that for rates above $\overline{R}_{\mathrm{Bern}}$, both COMP and SSS fail.

Let $\hat{\mathcal{K}}_{\mathrm{COMP}}$ and $\hat{\mathcal{K}}_{\mathrm{SSS}}$ be the sets returned by COMP and SSS, respectively. Recall that these are respectively the largest (see Lemma 2.1) and smallest satisfying sets, where a satisfying set is any putative set of defective items that could have produced the observed output (see Definition 2.1). The key observation is that if $|\hat{\mathcal{K}}_{\mathrm{COMP}}| > k$ and $|\hat{\mathcal{K}}_{\mathrm{SSS}}| < k$, then there exist at least two satisfying sets $\mathcal{L}$ with $|\mathcal{L}| = k$; any such set can be found by adding elements of $\hat{\mathcal{K}}_{\mathrm{COMP}}$ to $\hat{\mathcal{K}}_{\mathrm{SSS}}$ until reaching size $k$. Then, even if $k$ is known, the best one can do given multiple such sets is to choose one arbitrarily, yielding an error probability of at least $1/2$. Thus,

$$\begin{aligned} \mathbb{P}(\mathrm{err}) &\geq \frac{1}{2} \mathbb{P}\big(|\hat{\mathcal{K}}_{\mathrm{COMP}}| > k \cap |\hat{\mathcal{K}}_{\mathrm{SSS}}| < k\big) \\ &\geq \frac{1}{2}\big(1 - \mathbb{P}(|\hat{\mathcal{K}}_{\mathrm{COMP}}| = k) - \mathbb{P}(|\hat{\mathcal{K}}_{\mathrm{SSS}}| = k)\big) \end{aligned} \tag{4.21}$$

by the union bound.

We handle the two above terms separately. First, we observe that $\mathbb{P}(|\hat{\mathcal{K}}_{\mathrm{COMP}}| = k)$ is precisely the probability of COMP succeeding, since the largest satisfying set is necessarily unique (see Lemma 2.1). In Remark 2.2, it was shown that the success probability of COMP tends to zero for rates above the value $R_{\mathrm{Bern}}^{\mathrm{COMP}}$ defined in (2.8), which is strictly less than the rate $\overline{R}_{\mathrm{Bern}}$ that we consider here. Second, $\{|\hat{\mathcal{K}}_{\mathrm{SSS}}| = k\}$ cannot occur when a defective item is masked, and such a masking event was shown to occur with probability bounded away from zero in the proof of Theorem 2.2. Combining these two results, we find that (4.21) is bounded away from 0, which completes the proof. □

## 4.4 Improved rates with near-constant tests-per-item

In Section 2.7, we saw that the near-constant column weight (or near-constant tests-per-item) design introduced in Definition 2.3 gives improved rates for the COMP and DD algorithms, and that the SSS algorithm cannot attain a higher rate than $\min\left\{1, \ln 2 \, \frac{1-\alpha}{\alpha}\right\}$.

Similarly to the Bernoulli design, the converse of $\min\left\{1, \ln 2 \, \frac{1-\alpha}{\alpha}\right\}$ bits per test for SSS under the near-constant column weight design matches the achievability result for the DD algorithm stated in Theorem 2.8 when $\alpha \geq 1/2$. In this section, we describe a recent development that extends the achievability of the preceding rate to *all* $\alpha \in (0, 1)$, albeit at the expense of (potentially considerably) increased computation compared to DD.

Formally, the main result of Coja-Oghlan *et al.* [46] proves that the near-constant column weight design has a maximum achievable rate of

$$\overline{R}_{\mathrm{NCC}} = \min\left\{1, (\ln 2)\frac{1-\alpha}{\alpha}\right\},$$

as we stated in Theorem 4.2 above. We refrain from presenting the full details of the somewhat lengthy proof of Theorem 4.2, but we sketch the main steps.

On the whole, the analysis is less based on tools from information theory, and more based on direct probabilistic arguments. (Coja-Oghlan *et al.* note that similar arguments have been successful in the theory of constraint satisfaction problems.) Nevertheless, some similarities do exist between this approach and the information-theoretic analysis of Section 4.2. Notably, the error events associated with incorrect defective sets are handled separately according to the amount of difference with the true defective set. The error events corresponding to a small overlap (that is, an incorrect set having relatively few items in common with $\mathcal{K}$) have low probability for rates up to 1, and the error events corresponding to a large overlap (a large number of items in common) have low probability for rate up to $(\ln 2)\frac{1-\alpha}{\alpha}$ after the application of a tight concentration bound.

*Proof sketch of Theorem 4.2.* Consider the number of satisfying sets $\hat{\mathcal{K}}$ of the correct size $|\hat{\mathcal{K}}| = k$ that have a set difference with the true defective set of size $|\mathcal{K} \setminus \hat{\mathcal{K}}| = |\mathcal{K} \setminus \hat{\mathcal{K}}| = \tau$. Clearly there is one such set with $\tau = 0$, namely, the true defective set. If it can be shown that with high probability there are no

others, then the true defective set is the only satisfying set, and can – at least given enough computation time – be found reliably. Different bounds are used depending on whether $\tau \geq \tau^*$ or $\tau < \tau^*$, where we choose $\tau^* = k/\log_2 n$.

Similarly to the Bernoulli design, we may assume that $\mathcal{K}$ is fixed, say $\mathcal{K} = \{1, \dots, k\}$, without loss of generality. We first consider the 'small overlap' (or 'large difference') case, where $\tau \geq \tau^*$. Let $\mathcal{S}$ be the event that there exists a satisfying set corresponding to such a $\tau$. Using the union bound, we have

$$\mathbb{P}(\mathcal{S}) \leq \sum_{\tau=\tau^*}^{k} \binom{n-k}{\tau}\binom{k}{\tau}\mathbb{P}\big(\hat{\mathcal{K}}_\tau \text{ is satisfying}\big) \tag{4.22}$$

where $\hat{\mathcal{K}}_\tau$ is any set of size $k$ containing $\tau$ nondefectives and $k-\tau$ defectives. In [46], a concentration result and a coupling argument is used to show that the bound (4.22) for the near-constant column weight design is very close to the analogous bound for the Bernoulli design. This means that we can treat this case as though we were using the Bernoulli($p$) design, where $p = 1 - \mathrm{e}^{-\nu/k} \sim \nu/k$.

Pick $\nu = \ln 2$, so $1 - p = 2^{-1/k}$. (Here we follow an argument from [11].) A test has a different result under $\hat{\mathcal{K}}_\tau$ compared to $\mathcal{K}$ if no item in $\mathcal{K}$ is tested but an item in $\hat{\mathcal{K}}_\tau \setminus \mathcal{K}$ is tested, or vice versa. This has probability

$$2(1-p)^k\big(1-(1-p)^\tau\big) = 2(1-p)^k - 2(1-p)^{k+\tau}.$$

Hence,

$$\mathbb{P}\big(\hat{\mathcal{K}}_\tau \text{ is satisfying}\big) = \big(1 - 2(1-p)^k + 2(1-p)^{k+\tau}\big)^T,$$

and, using $(1-p)^k = \frac{1}{2}$, we have

$$\begin{aligned}
\mathbb{P}(\mathcal{S}) &\leq \sum_{\tau=\tau^*}^{k} \binom{n-k}{\tau}\binom{k}{\tau}\big(1 - 2(1-p)^k + 2(1-p)^{k+\tau}\big)^T \\
&= \sum_{\tau=\tau^*}^{k} \binom{n-k}{\tau}\binom{k}{\tau}\Big(1 - 2\cdot\tfrac{1}{2} + 2\cdot\tfrac{1}{2}\big(2^{-1/k}\big)^\tau\Big)^T \\
&= \sum_{\tau=\tau^*}^{k} \binom{n-k}{\tau}\binom{k}{\tau}2^{-\tau T/k}.
\end{aligned}$$

One can check that the summands here are decreasing, so the largest term is that for $\tau = \tau^*$, and for any $\delta > 0$ we have $\tau^* < \delta k$ for $n$ sufficiently large. Hence, we have

$$\begin{aligned}
\mathbb{P}(\mathcal{S}) &\leq k\binom{n-k}{\delta k}\binom{k}{\delta k}2^{-\delta k T/k} \\
&\leq k\left(\frac{\mathrm{e}n}{\delta k}\right)^{\delta k}\left(\frac{\mathrm{e}}{\delta}\right)^{\delta k}2^{-\delta T} \\
&= k2^{-\delta(T - k\log_2(n/k) - 2(\log_2 \mathrm{e} - \log_2 \delta)k)}
\end{aligned}$$

We see for $T > (1+\eta)k\log_2(n/k)$, which corresponds to any rate up to 1, that $\mathbb{P}(\mathcal{S})$ can be made arbitrarily small.

Next, we consider the 'large overlap' (or 'small difference') case, where $\tau < \tau^*$. The union bound argument above is too weak here, as 'rare solution-rich

instances drive up the expected number of solutions' [46]. Instead, we first show that a certain property $\mathcal{R}$ holds with high probability, and then use the expansion properties of the near-constant column weight design to show that, with high probability, no large-overlap solutions exist when $\mathcal{R}$ holds.

Property $\mathcal{R}$ is the event that every defective item $i \in \mathcal{K}$ is the unique defective item in at least $\delta L$ tests, for some $\delta > 0$. We need to show that $\mathcal{R}$ holds with high probability. To simplify this proof sketch, we present the analysis as though each item $i$ were included in exactly $L = \nu T/k$ tests chosen *without* replacement (rather than with replacement). Each of these $L$ tests contains no other defective items with probability

$$\left(1 - \frac{1}{T}\right)^{L(k-1)} = \left(1 - \frac{1}{T}\right)^{\nu T(1-1/k)} \to \mathrm{e}^{-\nu}.$$

For further simplification here (with the full details given in [46]), we make another non-rigorous approximation and suppose that each such test contains no other defectives with probability exactly $\mathrm{e}^{-\nu}$, and that this event is independent across tests. Write $M_i \overset{\mathrm{d}}{\approx} \mathrm{Bin}(L, \mathrm{e}^{-\nu})$ for the number of tests in which $i$ is the unique defective. Then the probability this is fewer than $\delta L$ tests is (approximately)

$$\mathbb{P}(M_i < \delta L) \approx \mathbb{P}\big(\mathrm{Bin}(L, \mathrm{e}^{-\nu}) < \delta L\big) \leq 2^{-L d(\delta \,\|\, \mathrm{e}^{-\nu})},$$

where $d(p \,\|\, q)$ is the relative entropy between a Bernoulli($p$) and a Bernoulli($q$) random variable, and we have used the standard Chernoff bound for the binomial distribution. It is clearly advantageous to take $\delta$ as small as possible, and doing so yields

$$\begin{aligned} d(\delta \,\|\, \mathrm{e}^{-\nu}) &= h(\delta) - \big(\delta \log_2 \mathrm{e}^{-\nu} + (1-\delta)\log_2(1 - \mathrm{e}^{-\nu})\big) \\ &\to -\log_2(1 - \mathrm{e}^{-\nu}). \end{aligned}$$

We can then use a union bound to write

$$\begin{aligned} \mathbb{P}(\mathcal{R}) &= \mathbb{P}\left(\bigcap_{i \in \mathcal{K}} M_i \geq \delta L\right) \\ &\leq 1 - k\mathbb{P}(M_i < \delta L) \\ &\lesssim 1 - k 2^{L \log_2(1 - \mathrm{e}^{-\nu})} \\ &= 1 - 2^{-(-\nu \log_2(1-\mathrm{e}^{-\nu})T/k - \log_2 k)}, \end{aligned}$$

where we substituted $L = \nu T/k$. The preceding bound can be made to approach 1 provided that

$$T > (1+\eta)\frac{1}{-\nu \log_2(1 - \mathrm{e}^{-\nu})} k \log_2 k$$

for some small $\eta > 0$. The term $-\nu \log_2(1 - \mathrm{e}^{-\nu})$ is maximised at $\nu = \ln 2$, where it takes the value $\ln 2$. Hence, the preceding condition reduces to

$$T > (1+\eta)\frac{1}{\ln 2} k \log_2 k,$$

which corresponds to rates no larger than $(\ln 2)\frac{1-\alpha}{\alpha}$.

It remains to argue that, conditioned on the event $\mathcal{R}$, we have no small-difference satisfying sets with high probability. The key observation of [46] is that switching even a single item from defective to nondefective would change the result of a large number (at least $\delta L$) of formerly positive tests. But turning these tests back positive requires switching many items from nondefective to defective, because the expansion properties of the design imply that it is unlikely that any item will be able to cover many of these tests. These switches in turn change the result of many more tests, requiring more switches, and so on. Hence, to get another satisfying set, one must switch the status of many items, and no 'large overlap' set can exist. While this is only an intuitive argument, it is formalized in [46].

Together, we have establishing vanishing probability for the existence of satisfying sets with either a small overlap or a large overlap (with 'small' and 'large' collectively covering all cases), and we are done. □

The use of the near-constant column weight design was crucial in checking that property $\mathcal{R}$ holds with high probability. Suppose that we instead use a Bernoulli($\nu/k$) design; then, the probability that a defective item is the unique defective in a given tests is

$$p(1-p)^{k-1} = \frac{\nu}{k}\left(1-\frac{\nu}{k}\right)^{k-1} \sim \frac{\nu \mathrm{e}^{-\nu}}{k}.$$

Hence, the probability of being the unique defective item in fewer than $\delta L = \delta\nu T/k$ tests is

$$\mathbb{P}\big(\mathrm{Bin}(T, \nu\mathrm{e}^{-\nu}/k) < \delta\nu T/k\big) \le 2^{-T d(\delta\nu/k \,\|\, \nu\mathrm{e}^{-\nu}/k)}.$$

Again, taking $\delta$ as small as possible, we have

$$d(\delta\nu/k \,\|\, \nu\mathrm{e}^{-\nu}/k) \sim \log_2(1-\nu\mathrm{e}^{-\nu}/k) \sim \frac{1}{\ln 2}\frac{\nu\mathrm{e}^{-\nu}}{k}.$$

Following the same argument leads to the conclusion that we avoid large-overlap errors with rates up to $\frac{\nu\mathrm{e}^{-\nu}}{\ln 2}\frac{1-\alpha}{\alpha}$, as in (4.1). Thus, we see that the achievable rate for Bernoulli designs (Theorem 4.1) can also be proved using the approach of [46].

We can match the achievability result of Theorem 4.2 with a converse for near-constant column weight designs. In particular, the proof of the corresponding result for Bernoulli designs (Theorem 4.3) extends easily to the near-constant column weight design once Theorem 2.9 on the SSS algorithm is in place. This extension is stated formally as follows for completeness.

**Theorem 4.4.** *Consider noiseless group testing in the sparse regime $k = \Theta(n^\alpha)$, with the exact recovery criterion, and the near-constant column weight design (Definition 2.3). If the rate exceeds $\overline{R}_{\mathrm{NCC}}$ defined in* (4.1)*, then the error probability averaged over the testing matrix is bounded away from zero, regardless of the decoding algorithm.*

This result readily establishes that the achievable rate for the DD algorithm, stated in Theorem 2.8, is optimal (with respect to the random test design) when $\alpha \ge 1/2$. Recall that these rates are plotted in Figure 4.1.

## 4.5 Noisy models

We now turn our attention to noisy settings, considering general noise models of the form (4.3) – that is, those that satisfy the only defects matter property of Definition 3.2. As we discussed previously, our initial achievability analysis leading to the non-asymptotic bound (4.15) is valid for any such model, and hence, a reasonable approach is to follow the subsequent steps of the noiseless model. The main difficulty in doing so is establishing suitable concentration inequalities analogous to Lemma 4.2.

We proceed by presenting a general achievability result for the very sparse regime $k = O(1)$, where establishing the desired concentration is straightforward. We also give a matching converse bound that remains valid for the sparse regime $k = \Theta(n^\alpha)$. Achievability in the sparse regime is more difficult, and is postponed to Section 4.5.2.

### 4.5.1 General noise models in the very sparse regime

The following theorem provides a general characterization of the required number of tests in terms of suitable conditional mutual information quantities. This result was given in the works of Malyutov [143] and Atia and Saligrama [17]; see also [64] for a survey paying finer attention to the error exponent (that is, the exponential rate of decay of the error probability) and considering universal decoding rules (where the noise distribution is not known).

**Theorem 4.5.** *Consider any noiseless group testing setup of the form* (4.3)*, with* Bernoulli($p$) *testing and* $k = \Theta(n^\alpha)$ *with* $\alpha \in [0, 1)$*. Then in order to achieve vanishing error probability as* $n \to \infty$*, it is necessary that*

$$T \geq \max_{\tau=1,\dots,k} \frac{\tau \log_2 \frac{n}{\tau}}{I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau})}(1 - o(1)), \tag{4.23}$$

*where the mutual information is with respect to the independent random vectors* $(\mathbf{X}_{0,\tau}, \mathbf{X}_{1,\tau})$ *of sizes* $(\tau, k-\tau)$ *containing independent* Bernoulli($p$) *entries, along with the noise model* $P_{Y|V}$ *in* (4.3)*. Moreover, in the case that* $\alpha = 0$ *(*$k = O(1)$*) a matching achievability bound holds, and the maximum achievable rate is given by*

$$\overline{R}^{\text{noisy}}_{\text{Bern}} = \min_{\tau=1,\dots,k} \frac{1}{\tau} I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau}) \tag{4.24}$$

$$= I(\mathbf{X}_{0,k}; Y). \tag{4.25}$$

Observe that the equality (4.25) states that the minimum in (4.24) is achieved by $\tau = k$, and the capacity reduces to a single unconditional mutual information term. Moreover, if the noisy defective channel property holds (Definition 3.3) and the Bernoulli testing parameter is optimized, this mutual information term reduces to the corresponding channel capacity – for example, $I(U; Y) = 1 - h(\rho)$ for the symmetric noise model.

However, the capacity equalling (4.25) crucially relies on two assumptions: (i) the observation model (4.3) is symmetric, in the sense of depending only on the number of defectives in the test, and not the specific defectives included; and (ii) the number of defectives is bounded, i.e., $k = O(1)$. Counterexamples

to (4.25) in cases that the former condition fails can be found in [143]. As for the latter condition, we observe that the term $\log_2 \frac{n}{\tau}$ can range from $\log_2 \frac{n}{k}$ to $\log_2 n$, and these two terms can have a non-negligible difference when $k$ scales with $n$. For instance, the analysis of [163] reveals that the term corresponding to $\tau = 1$ can dominate in the regime $k = \Theta(n^\alpha)$ when $\alpha \in (0, 1)$ is sufficiently close to one.

The assumption $k = O(1)$ in the achievability part is rather restrictive; we discuss this point further in Section 4.5.2. Another limitation of Theorem 4.5 is that the converse part is specific to Bernoulli testing; however, we present variants for arbitrary test matrices in Section 4.5.3.

### Discussion of achievability proof

Proofs of the achievability part of Theorem 4.5 can be found in [145, 17, 163]; continuing the earlier analysis, we discuss the approach of [163].

As mentioned above, the bound (4.15) remains valid in the noisy setting, and the main step in the subsequent analysis is establishing the concentration of $\imath^T(\mathsf{X}_{0,\tau}; \mathbf{Y} | \mathsf{X}_{1,\tau})$. In general, this is a challenging task, and may introduce extra conditions on $T$, as we saw in the proof of Theorem 4.1. However, it turns out that when $k = O(1)$, the concentration bound given in the second part of Lemma 4.2 (which extends immediately to general noise models [165]) is sufficient. Indeed, assuming bounded $k$ greatly simplifies matters, since it means that the combinatorial term $\binom{k}{\tau}$ in (4.15) is also bounded.

The equality (4.25) follows from elementary information-theoretic arguments, which we outline here. Assuming without loss of generality that $\mathcal{K} = \{1, \dots, k\}$, writing the entries of $\mathbf{X}_\mathcal{K}$ as $(X_1, \dots, X_k)$ accordingly, and letting $\mathbf{X}_j^{j'}$ denote the collection $(X_j, \dots, X_{j'})$ for indices $1 \le j \le j' \le k$, we have

$$\begin{aligned}
\frac{1}{\tau} I(\mathbf{X}_{0,\tau}; Y \mid \mathbf{X}_{1,\tau}) &= \frac{1}{\tau} I(\mathbf{X}_{k-\tau+1}^{k}; Y \mid \mathbf{X}_1^{k-\tau}) && (4.26)\\
&= \frac{1}{\tau} \sum_{j=k-\tau+1}^{k} I(X_j; Y \mid \mathbf{X}_1^{j-1}) && (4.27)\\
&= \frac{1}{\tau} \sum_{j=k-\tau+1}^{k} \big( H(X_j) - H(X_j \mid Y, \mathbf{X}_1^{j-1}) \big), && (4.28)
\end{aligned}$$

where (4.26) follows from the definition of $(\mathbf{X}_{0,\tau}, \mathbf{X}_{1,\tau})$ and the symmetry of the noise model in (4.3), (4.27) follows from the chain rule for mutual information, and (4.28) follows since $X_j$ is independent of $\mathbf{X}_1^{j-1}$. We establish the desired claim by observing that (4.28) is decreasing in $\tau$: The term $H(X_j)$ is the same for all $j$, whereas the term $H(X_j | Y, \mathbf{X}_1^{j-1})$ is smaller for higher values of $j$ because conditioning reduces entropy.

### Discussion of converse proof

In light of the apparent connection between channel coding and group testing (see Figure 1.1), a natural starting point is to apply Fano's inequality, which states that in order to achieve an error probability of $\delta$, it is necessary that

$$I(\mathcal{K}; \mathbf{Y} \mid \mathsf{X}) \ge \log_2 \binom{n}{k} (1 - \delta) - 1. \tag{4.29}$$

Note that $\mathbf{Y}$ depends on $\mathcal{K}$ only through $\mathsf{X}_{\mathcal{K}}$, which corresponds to $\mathsf{X}_{0,k}$ in the above notation. We can therefore replace $I(\mathcal{K};\mathbf{Y} \mid \mathsf{X})$ by $I(\mathsf{X}_{0,k};\mathbf{Y})$, which in turn equals $TI(\mathbf{X}_{0,k};Y)$ since the tests are independent. Substituting into (4.29) and rearranging, we obtain the necessary condition

$$T \geq \frac{\log_2 \binom{n}{k}}{I(\mathbf{X}_{0,k};Y)}\left(1-\delta-\frac{1}{\log_2 \binom{n}{k}}\right). \tag{4.30}$$

This bound matches (4.23) whenever the maximum therein is achieved by $\tau = k$. However, as discussed above, this is not always the case.

The key to overcoming this limitation is to use a 'genie argument' [17], in which a subset of $\mathcal{K}$ is revealed to the decoder, and it only remains to estimate the non-revealed part. This clearly only makes the recovery problem easier, so any converse for this genie-aided setting remains valid in the original setting. Note that since $\mathsf{X}$ is generated in a symmetric i.i.d. manner and the assumed model (4.3) is invariant to relabelling, it makes no difference precisely which indices are revealed; all that matters is the number revealed. (However, the revealed indices must not depend on $\mathsf{X}$ or $\mathbf{y}$.) Letting $\tau$ denote the number of defectives left to estimate, the number revealed is equal to $k-\tau$.

In the genie-aided setting, the number of possible defective sets reduces from $\binom{n}{k}$ to $\binom{n-k+\tau}{\tau}$. Moreover, the relevant mutual information in Fano's inequality is not $I(\mathcal{K};\mathbf{Y}|\mathsf{X})$, but instead $I(\mathcal{K}_{0,\tau};\mathbf{Y}|\mathcal{K}_{1,\tau},\mathsf{X})$, where $\mathcal{K}_{0,\tau}$ (respectively, $\mathcal{K}_{1,\tau}$) denotes the non-revealed (respectively, revealed) defective item indices. Upon upper bounding the mutual information via the data processing inequality, we obtain the following analogue of (4.30):

$$T \geq \frac{\log_2 \binom{n-k+\tau}{\tau}}{I(\mathbf{X}_{0,\tau};Y|\mathbf{X}_{1,\tau})}\left(1-\delta-\frac{1}{\log_2 \binom{n-k+\tau}{\tau}}\right). \tag{4.31}$$

We then recover (4.23) by maximizing over $\tau = 1,\dots,k$ and noting that

$$\log_2 \binom{n-k+\tau}{\tau} = \left(\tau \log_2 \frac{n}{\tau}\right)(1+o(1)). \tag{4.32}$$

We mention that an alternative approach was taken in [163], bearing a stronger resemblance to the above achievability proof and again relying on change-of-measure techniques from the channel coding literature. The proof of [163] has the advantage of recovering the so-called 'strong converse' (see Remark 1.3), but it requires additional effort in ensuring that the suitable sums of information densities concentrate around the corresponding conditional mutual information.

**Examples: Addition and dilution noise**

We briefly discuss the application of Theorem 4.5 to two asymmetric noise models introduced above (symmetric noise will be handled in greater generality in Section 4.5.2):

- Recall from Example 3.2 that the *addition noise* model takes the form $Y_t = \big(\bigvee_{i\in\mathcal{K}} \mathsf{X}_{ti}\big) \vee Z_t$, where $Z_t \sim \text{Bernoulli}(\varphi)$. By bounding the mutual information in Theorem 4.5, it was shown in [17] that the optimal number of tests behaves as $O\big(\frac{k\log_2 \frac{n}{k}}{1-\varphi}\big)$. Hence, we have a simple linear dependence on the addition noise parameter.

- Recall from Example 3.3 that the *dilution noise* model takes the form $Y_t = \bigvee_{i \in \mathcal{K}} \big(\mathsf{X}_{ti} \wedge Z_{ti}\big)$, where $Z_{ti} \sim \text{Bernoulli}(\vartheta)$. By bounding the mutual information in Theorem 4.5, it was shown in [17] that the optimal number of tests behaves as $O\big(\frac{k \log_2 n}{(1-\vartheta)^2}\big)$. Hence, the dependence in the denominator is quadratic.

Additional expressions for the relevant mutual information terms for various noise models, often including precise constant factors, can be found in [125].

### 4.5.2 Achievability in general sparse regimes

Here we outline techniques and results for attaining achievability bounds for the noisy setting in the sparse regime $k = \Theta(n^\alpha)$ with $\alpha \in (0, 1)$, as opposed to the very sparse regime $k = O(1)$ stated in Theorem 4.5. We focus on the symmetric noise model (Example 3.1), but the techniques that we discuss can also be applied to other noise models.

We state the main achievability result of Scarlett and Cevher [163] as follows, and then discuss the proof.

**Theorem 4.6.** *Under the symmetric noise model with parameter $\rho$ in the regime $k = \Theta(n^\alpha)$ ($\alpha \in (0, 1)$), under Bernoulli testing with an optimized parameter, there exists a decoder achieving the rate*

$$R_{\text{symm}}^{\text{Bern}} = \min\big\{1 - h(\rho), c(\rho, \alpha)\big\}, \tag{4.33}$$

*where $h(\rho)$ is the binary entropy function in bits, and $c(\rho, \alpha)$ is a continuous function with $c(\rho, 0) > 1 - h(\rho)$. In particular, for sufficiently small $\alpha$, the capacity is given by*

$$C_{\text{symm}} = 1 - h(\rho). \tag{4.34}$$

An explicit expression for $c(\rho, \alpha)$ is given in [163], but it is omitted here since it is complicated and does not provide additional insight. Most interesting is the fact that Theorem 4.6 provides the *exact* capacity (even for non-Bernoulli and possibly adaptive tests) for all $\alpha \in (0, \alpha_0)$, where $\alpha_0$ is strictly positive but may depend on $\rho$.

The rates of Theorem 4.6 are plotted in Figure 4.2 (repeated from Figure 3.2 for convenience) for two different noise levels. The achievability and converse bounds are identical for sufficiently small $\alpha$, albeit over a much smaller range compared to the noiseless case. This range could potentially be widened by improving a step of the proof (outlined below) based on concentration of measure; currently Bernstein's inequality is used, which is somewhat crude.

**Outline of proof** Starting with the non-asymptotic bound (4.23), the analysis proceeds as follows:

- Analogously to Lemma 4.1, a direct evaluation of the conditional mutual information $I_\tau$ yields the following: If $\frac{\tau}{k} = o(1)$, then

$$I_\tau = \left(\mathrm{e}^{-\nu} \nu \frac{\tau}{k} (1 - 2\rho) \log_2 \frac{1-\rho}{\rho}\right)(1 + o(1)), \tag{4.35}$$

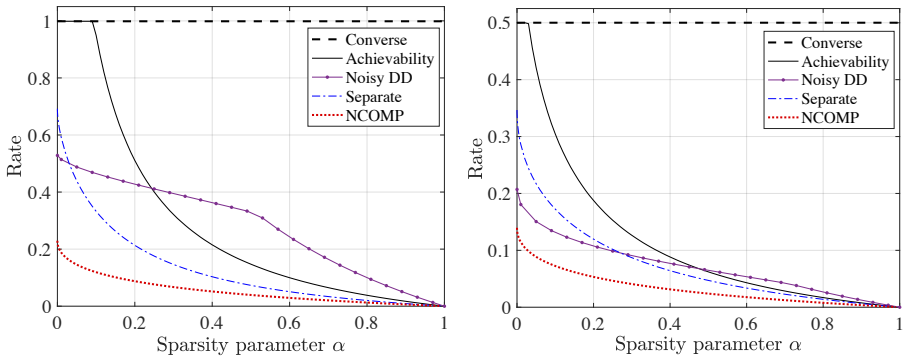



Figure 4.2: Achievable rates for the symmetric noise model with noise levels $\rho = 10^{-4}$ (Left) and $\rho = 0.11$ (Right).

whereas if $\frac{\tau}{k} \to \psi \in (0, 1]$, then

$$I_\tau = \mathrm{e}^{-(1-\psi)\nu}\big(h\big(\mathrm{e}^{-\psi\nu} \star \rho\big) - h(\rho)\big)(1 + o(1)), \tag{4.36}$$

where we use the notation $a \star b = ab + (1-a)(1-b)$.

- In the same way as the noiseless setting, we adopt the condition

$$T \geq \frac{\log_2 \binom{n-k}{\tau} + \log_2 \left(\frac{k}{\delta}\binom{k}{\tau}\right)}{I_\tau(1-\delta'_\tau)}, \tag{4.37}$$

and upper bound the probability in (4.15) by $\mathbb{P}\big(\imath^T(\mathsf{X}_{0,\tau}; \mathbf{Y}|\mathsf{X}_{1,\tau}) < TI_\tau(1-\delta'_\tau)\big)$. Hence, we are again faced with the problem of establishing the concentration of $\imath^T$.

- As before, we write $\tau^* = k/\sqrt{\log_2 k}$. The values $\delta'_\tau$ are again set to some small value $\epsilon$ for $\tau > \tau^*$, whereas for $\tau \leq \tau^*$, they are set to a common value; this value is left as a free parameter $\delta'' \in (0, 1)$ to optimize at the end.[1]

- For $\tau > \tau^*$, we use the concentration bound in the second part of Theorem 4.2, which is valid for any noise model as long as the observations are binary [163]. Some manipulations show that no matter how small the value of $\delta'_\tau = \epsilon$, the concentration is sufficiently sharp to contribute a negligible amount to the error probability as long as $T = \Omega(k \log n)$.

- For $\tau \leq \tau^*$, a different concentration bound is needed. This is established using Bernstein's inequality; recall that the same approach was used for separate decoding of items in Section 3.5. The need for sufficiently sharp concentration, and the condition in (4.37) for $\tau \leq \tau^*$ (with free parameter $\delta'_\tau = \delta''$), lead to the presence of the term $c(\rho, \alpha)$ in (4.33). The optimal choice of $\delta''$ varies depending on both $\rho$ and $\alpha$.

[1] This method for choosing $\delta'_\tau$ differently for small and large $\tau$ was not mentioned explicitly in the relevant section of [163], but it is necessary for establishing Theorem 4.6. See [161, Appendix A] for further discussion.

- The remaining term $1 - h(\rho)$ arises from (4.37) for $\tau \geq \tau^*$, which turns out to be maximized asymptotically by $\tau = k$. Recall that for this range of $\tau$, we have $\delta'_\tau = \epsilon$ for arbitrarily small $\epsilon > 0$.

The interested reader is referred to [163] for details.

### 4.5.3 Converse bounds for general test designs

A weakness of the converse result in Theorem 4.5 is that it only holds for Bernoulli test matrices. We conclude this chapter by briefly discussing converse results that hold for arbitrary (but still nonadaptively designed) test matrices.

Ideally, to match the achievability bound, we would have a converse bound of a similar form to that of Theorem 4.5 with a maximization over $\tau = 1, \dots, k$. However, as discussed in Section 4.5, the term corresponding to $\tau = k$ is arguably the most important, and lower bounding the maximum in (4.23) by this term, we obtain the following necessary condition under Bernoulli testing:

$$T \geq \frac{(1 - \zeta_T)\binom{n}{k}}{I(\mathbf{X}_\mathcal{K}; Y) + \eta_T}. \tag{4.38}$$

We proceed by stating an extension of this converse to arbitrary test matrices, rather than only Bernoulli test matrices. Despite holding for arbitrary test matrices, the result is still stated in terms of the mutual information term

$$I_{\max} = \max_{p \in [0,1]} I(\mathbf{X}'_\mathcal{K}; Y')$$

where $\mathbf{X}'_\mathcal{K}$ is defined as an i.i.d. Bernoulli test vector of length $k$ with probability $p$ of each entry equalling 1, and $Y'$ is generated from $\mathbf{X}'_\mathcal{K}$ according to the noise model $P_{Y|V}$ under consideration (see (4.3)).

**Theorem 4.7.** *For any noise model satisfying the only defects matter property (Definition 3.2), if*

$$T \leq \frac{\log_2 \binom{n}{k}}{I_{\max}}(1 - \eta) \tag{4.39}$$

*for arbitrarily small $\eta > 0$, then the error probability satisfies $\mathbb{P}(\text{err}) \geq 1 - O\left(\frac{1}{n I_{\max}^2}\right)$.*

This result is due to Scarlett and Cevher [162], with the proof adopting a similar approach to the one discussed following (4.32). We note that this is a 'strong converse' result giving conditions under which $\mathbb{P}(\text{err}) \to 1$ (at least when $I_{\max}$ is not too small), as opposed to the weaker statement that $\mathbb{P}(\text{err})$ is bounded away from zero. We made a similar distinction in the noiseless setting, noting that Theorem 1.1 implies a strong converse there.

For channels satisfying the defective channel property (Definition 3.3), it is straightforward to show that $I_{\max}$ is simply the corresponding channel capacity. For example, under the symmetric noise model (Examples 3.1), the bound on the number of tests is simply

$$T \leq \frac{k \log_2 \frac{n}{k}}{1 - h(\rho)}(1 - \eta),$$

for any $\eta > 0$, where as usual we write $h$ for the binary entropy. This matches the capacity expression given in (4.34), with the added benefit of providing the strong converse.

Finally, we briefly remark that (weak) converse bounds for arbitrary test matrices including a maximum over $\tau = 1, \dots, k$ were given in [144, p. 630-631] and [162]. We avoid stating the result of [162] explicitly, but highlight that when $k = O(1)$ it recovers a necessary condition of the following form:

$$T \geq \max_{\tau=1,\dots,k} \min_{p\in[0,1]} \frac{\tau \log_2 \frac{n}{\tau}}{I(\mathbf{X}_{0,\tau}; Y|\mathbf{X}_{1,\tau})}(1 - o(1)), \tag{4.40}$$

where $p \in [0, 1]$ is the Bernoulli probability parameter defining $\mathbf{X}_{0,\tau}$ and $\mathbf{X}_{1,\tau}$. This matches the achievability result of Theorem 4.5 up to the max-min ordering; to our knowledge, it is currently unknown whether this difference in ordering can introduce a gap between the achievability and converse bounds.

# Chapter 5

# Other Topics in Group Testing

In this chapter, we explore extensions of group testing beyond the settings considered in the previous chapters, which were primarily focused on nonadaptive randomized designs, the exact recovery criterion, and sublinear scaling in the number of defectives. Many of the extensions considered below have natural analogues in classical information theory, and we attempt to draw such parallels when they arise naturally.

## 5.1 Partial recovery

In many group testing situations, one might be satisfied with an estimate of the defective set $\hat{\mathcal{K}}$ being very close to the true defective set $\mathcal{K}$, without demanding the exact recovery criterion we have considered throughout this survey. That is, while one would wish for the number of *false negative items* $|\hat{\mathcal{K}}^{\mathsf{c}} \cap \mathcal{K}|$ and the number of *false positive items* $|\hat{\mathcal{K}} \cap \mathcal{K}^{\mathsf{c}}|$ to be small. (Here and subsequently, we write $\mathcal{K}^{\mathsf{c}} = \{1, \dots, n\} \setminus \mathcal{K}$.) it might not always be necessary for both to be zero. For example, when screening for diseases, a small number of false positives might lead to slightly more medical attention for those who did not need it, a cost which might be small compared to performing many more pooled tests.

We briefly mention that moving to approximate recovery is known to significantly help under the zero-error recovery criterion, allowing one to break the $\Omega(k^2)$ barrier discussed in Section 1.6 and instead use only $O(k \log n)$ tests, even in the presence of adversarial test errors [39]. In the following, we focus only on the small-error criterion.

Permitting a limited number of mistakes in the reconstruction is analogous to the notion of *rate–distortion theory* in communication, where one only requires a source to be reconstructed approximately (see [49, Chapter 10] for a review of this topic). A natural performance criterion that treats both error types equally is

$$\mathbb{P}_d(\text{err}) = \mathbb{P}\big(|\mathcal{K} \cap \hat{\mathcal{K}}^{\mathsf{c}}| > d \text{ and } |\hat{\mathcal{K}} \cap \mathcal{K}^{\mathsf{c}}| > d\big),$$

which declares an error if either the number of false negatives or false positives exceeds a common threshold $d$.

The following result, due to Scarlett and Cevher [163], characterizes how this relaxed criterion affects the required number of tests when this threshold is a constant fraction of the number of defectives.

**Theorem 5.1.** *Under nonadaptive Bernoulli(p) group testing with $k = o(n)$ and $d = \gamma k$ for some $\gamma \in (0, 1)$, we have the following:*

- *With $p = \frac{\ln 2}{k}$, there exists an algorithm such that $\mathbb{P}_d(\mathrm{err}) \to 0$ provided that $T > (1+\eta)T^*$ for arbitrarily small $\eta > 0$, where*

$$T^* = k \log_2 \frac{n}{k}.$$

- *For any $p$ and any algorithm, in order to achieve $\mathbb{P}_d(\mathrm{err}) \to 0$ it is necessary that $T > (1-\eta)T^*_\gamma$ for arbitrarily small $\eta > 0$, where*

$$T^*_\gamma = (1-\gamma)k \log_2 \frac{n}{k} = (1-\gamma)T^*.$$

The achievability part can be proved using the argument used to prove Theorem 4.1 (see Section 4.2). We need only consider only the errors with small overlap with the true defective set, a errors with large overlap still suffice for partial reconstruction.

This result implies both positive and negative results on the extent to which approximate recovery reduces the number of tests under the Bernoulli design compared with the bounds discussed in Chapter 4. Letting $\alpha$ be the exponent such that $k = \Theta(n^\alpha)$ as usual, we observe the following:

- For $\alpha \le 1/3$, the gain is very limited, amounting to at most a reduction by the multiplicative factor $1 - \gamma$, which vanishes as $\gamma \to 0$. This is because nonadaptive Bernoulli testing achieves a rate of 1 in this regime.
- For $\alpha > 1/3$, the gain is more significant – the number of tests remains $\big(k \log_2 \frac{n}{k}\big)(1 + o(1))$ under the approximate recovery criterion, whereas for exact recovery the rate tends to zero as $\alpha \to 1$ under Bernoulli testing.

We briefly mention that extensions of Theorem 5.1 are given in [164] to a *list decoding* setting, in which the decoder outputs a list of length $L \ge k$ and it is only required that the list contains $(1-\gamma)k$ defectives. (The concept of list decoding for group testing also appeared much earlier under the zero-error criterion, e.g., see [110]). If $L$ is much larger than $k$, this means that we are potentially allowing a large number of false positives. However, a finding of [164] is that this relaxation often only amounts to a replacement of $k \log_2 \frac{n}{k}$ by $k \log_2 \frac{n}{L}$ in the required number of tests (asymptotically), which is a rather minimal gain.

It is also of interest to understand the achievable rates of the practical algorithms studied in Chapter 2 and 3 under the partial recovery criterion. In the noiseless case, it is in fact straightforward to extend the exact recovery analysis of COMP and DD:

- The COMP algorithm (Algorithm 2.2) always has no false negatives, and the analysis of Section 2.3 shows that when $T \ge (\mathrm{e}k \ln n)(1+\eta)$ so (i.e., the rate is below $(1-\alpha)/(\mathrm{e} \ln 2)$), the average number of false positives

tends to zero, and therefore the probability of having one or more false positives also tends to zero. By a nearly identical analysis, one finds that when $T \geq \left(\mathrm{e}k \ln \frac{n}{k}\right)(1+\eta)$, the average number of false positives behaves as $o(k)$, and therefore the probability of having more than $\gamma k$ false positives tends to zero for any fixed $\gamma \in (0,1)$, by Markov's inequality.

- The DD algorithm (Algorithm 2.3) always has no false positives, and the analysis of Section 2.4 shows that when $T \geq \left(\mathrm{e}k \ln \frac{n}{k}\right)(1+\eta)$, any given defective item is the unique 'possible defective' (PD) in some test, with probability approaching one. For exact recovery, an additional condition $T \geq (\mathrm{e}k \ln k)(1+\eta)$ arises from a union bound over the $k$ defective items. In the case of partial recovery, however, we can instead use the fact that the number of defective items failing to be the unique PD in some test behaves as $o(k)$, and therefore, the probability of having more than $\gamma k$ false negatives tends to zero for any fixed $\gamma \in (0,1)$, by Markov's inequality.

Hence, using Definition 1.7 and (1.11), a rate of $R = \frac{1}{\mathrm{e} \ln 2} \approx 0.531$ is achieved by COMP with no false negatives, and by DD with no false positives. Using similar arguments based on avoiding the union bound and instead applying Markov's inequality, it has been shown that separate decoding of items (see Section 3.5) achieves a rate of $\ln 2 \approx 0.693$ in the noiseless setting when both $\gamma k$ false positives and $\gamma k$ false negatives are allowed [167]. This rate is slightly higher than that of COMP and DD above, but comes with the caveat of requiring both false positives and false negatives. The results are summarized in Table 5.1.

| | Partial recovery rate | No false + | No false − |
|---|---|---|---|
| **Optimal** | 1 | no | no |
| **COMP** | $\frac{1}{\mathrm{e} \ln 2}$ | no | yes |
| **DD** | $\frac{1}{\mathrm{e} \ln 2}$ | yes | no |
| **Separate Dec.** | $\ln 2$ | no | no |

Table 5.1: Summary of achievable rates (in bits/test) for partial recovery under Bernoulli testing. Each achievable rate holds for all $\alpha \in (0,1)$ and an arbitrarily small (but constant) fraction of mistakes in the reconstruction. The final two columns indicate whether the algorithm is guaranteed to have no false positives/negatives.

Analogous results have also been given in noisy settings. For instance, under the symmetric noise model with parameter $\rho \in (0, 1/2)$, the information-theoretic rate given in Theorem 5.1 naturally becomes $1 - h(\rho)$, and the rate based on separate decoding of items becomes $(1 - h(\rho)) \ln 2$. The interested reader is referred to [167, 163].

## 5.2 Adaptive testing with limited stages

We saw in Section 1.5 that adaptive testing permits the exact zero-error identification of $\mathcal{K}$ with an information-theoretically optimal rate $R = 1$. This offers

two key advantages over nonadaptive testing: replacing the small-error criterion by the zero-error criterion, and achieving $R = 1$, which is only known to be possible for $k = O(n^{0.409})$ in the nonadaptive case (and even then, it is not known how to achieve it *efficiently*). On the other hand, adaptive testing schemes may come with considerable overhead compared to nonadaptive testing, since it is no longer possible to perform all of the tests in parallel.

An interesting variation that potentially attains the benefits of both worlds is *two-stage testing*, in which a very limited amount of adaptivity is allowed; namely, one can only perform two stages of testing, in which the tests in the second stage can depend on the outcomes in the first stage. The binary splitting algorithm (Algorithm 1.1) described in Section 1.5 does not fall into this category, and in fact uses $O(\log n)$ stages.

A variety of algorithms and results have been proposed for the two-stage setting [22, 56, 57, 135, 149]. Here we present a result of Mézard *et al.* [149], which improves on the earlier bounds of [22]. Note that here the notion of 'rate' is defined (in the same way as Definition 1.8) with respect to the *average* number of tests for a random defective set; we refer to this as the *variable-T setting*.

**Theorem 5.2.** *Consider the problem of two-stage group testing in the variable-T setting with zero error. When $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, the following rate is achievable:*

$$R_2 = \begin{cases} \frac{1}{\mathrm{e} \ln 2} \approx 0.531 & \alpha \leq \frac{1}{2} \\ \ln 2 \approx 0.693 & \alpha > \frac{1}{2}. \end{cases}$$

This result was stated in [149] under the i.i.d. prior defectivity model (see the Appendix to Chapter 1), but the proof transfers easily to the combinatorial prior. On the other hand, [149] also states a converse of $\ln 2$ for all $\alpha \in (0, 1)$ under the i.i.d. prior (in particular matching the achievability part when $\alpha > \frac{1}{2}$), and while we expect this to remain true under the combinatorial prior, we refrain from stating so formally.

Since the second stage tests all undetermined items individually, the freedom in the test design is entirely in the first stage. For $\alpha > 1/2$, this stage is based on the standard i.i.d. Bernoulli testing procedure considered throughout this monograph, whereas for $\alpha \leq 1/2$, an alternative construction is used in which $\mathsf{X}$ has both constant row weight and constant column weight. We observe that two-stage testing with zero error probability requires *considerably* fewer tests compared to the nonadaptive case, in particular avoiding the $\Omega(k^2)$ barrier (see Section 1.6).

At this point, it is natural to question whether there exists a more general trade-off between the number of stages and the rate. This question was addressed by Damaschke *et al.* [56] under both the small-error and zero-error recovery criteria. Among other things, it was proved that even in the zero-error setting, four stages is enough to attain a rate of one.

**Theorem 5.3.** *For any $k = o(n)$, the capacity of four-stage adaptive group testing in the variable-T setting with zero error is $C_4 = 1$.*

A high-level description of the four-stage algorithm establishing this result is as follows:

- The $n$ items are split into $\frac{k}{\Delta}$ 'cells' of size $\frac{\Delta n}{k}$, where $\Delta \to 0$ sufficiently slowly so that $\Delta \log \frac{n}{k} \to \infty$.

- In the first stage, each test (one per cell) consists of all the items in a given cell, and hence, all empty cells are identified as contaiing only nondefectives.

- In the second stage, for each non-empty cell, a known nonadaptive procedure is used to identify whether the cell has exactly one defective item (and if so, also determine its index) or more than one item.

- In the third and fourth stages, the cells having multiple items are merged, and a two-stage group testing procedure is applied (e.g., the one corresponding to Theorem 5.2 suffices).

It is natural to question whether an analogous result holds with two or three stages. The converse result discussed following Theorem 5.2 suggests that the answer is negative in the case of two stages and the zero-error criterion. On the other hand, if one considers the small-error criterion in place of zero-error, a rate of one can be achieved, as shown in the following result of [161] (see also [56] for an analogous earlier result using three stages).

**Theorem 5.4.** *For any $k = \Theta(n^\alpha)$ with $\alpha \in (0,1)$, the capacity of two-stage adaptive group testing in the fixed-T setting under the small-error criterion is $C_2 = 1$.*

The high-level idea of the proof is straightforward:

- By the partial recovery result of Theorem 5.1, in the first stage, we can find an estimate $\hat{\mathcal{K}}_1$ of cardinality $k$ with at most $\gamma k$ false positives and $\gamma k$ false negatives, where $\gamma > 0$ is an arbitrarily small constant.

- In the second stage, we apply any nonadaptive noisy strategy on the reduced ground set $\{1, \dots, n\} \setminus \hat{\mathcal{K}}_1$ to resolve the false negatives. As long as this strategy achieves a positive rate, the required number of tests will be $O(\gamma k \log n)$, which is negligible since $\gamma$ is arbitrarily small.

- Simultaneously in the second stage, we test the items in $\hat{\mathcal{K}}_1$ individually to resolve the false positives. This only requires $k$ tests.

This approach was adopted in [161] not only for the noiseless setting, but also for noisy settings. The changes compared to the noiseless setting are outlined as follows. In the first step, one can make use of the partial recovery results for the noisy setting outlined at the end of Section 5.1. In the second step, one can use a noisy nonadaptive algorithm that achieves a positive rate, such as the NCOMP algorithm introduced in Section 3.4. In the third step, testing each item once is no longer sufficient, but testing each item $\Theta(\log_2 k)$ times is enough to combat the noise.

This approach, as well as a slightly refined three-stage version, led to significant improvements over the best known information-theoretic achievable rates for noisy nonadaptive group testing, as illustrated in Figure 5.1. Another contribution of [161], also demonstrated in this figure, was to provide an algorithm-independent converse demonstrating that the rate must approach zero as $\alpha \to 1$ (regardless of the number of stages of adaptivity), in stark contrast to the noiseless setting in which a rate of one can be achieved for all $\alpha \in (0,1)$. While the

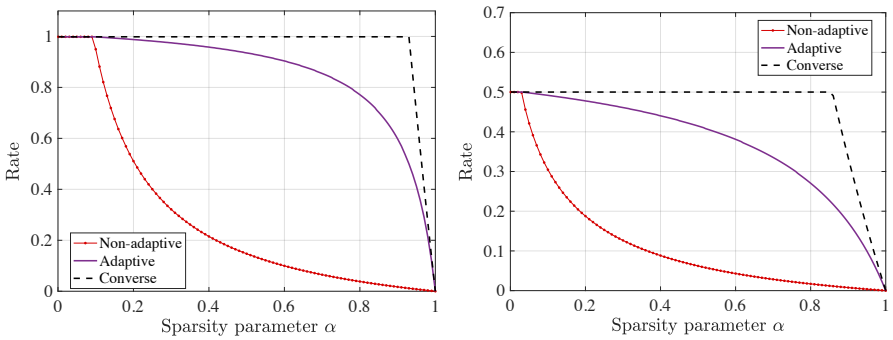



Figure 5.1: Achievable rates for the symmetric noise model (Example 3.1) with noise levels $\rho = 10^{-4}$ (Left) and $\rho = 0.11$ (Right).

achievability curve in Figure 5.1 does not correspond to a computationally efficient algorithm (since the first adaptive stage uses an intractable information-theoretic decoder), a near-identical result for a *four-stage* efficient variant was recently obtained in [160]. (In fact, these four stages can be reduced to three in the noiseless setting.) Closing the remaining gaps between the achievability and converse bounds remains an interesting open problem.

Alongside the results for the noisy setting shown in Figure 5.1, the results for the noiseless setting surveyed in this section are summarized in Table 5.2.

| Reference | Rate | #Stages | Zero-error? |
|---|---|---|---|
| [149] | $\begin{cases} \frac{1}{e \ln 2} & \alpha \leq \frac{1}{2} \\ \ln 2 & \alpha > \frac{1}{2} \end{cases}$ | 2 | yes |
| [56] | 1 | 4 | yes |
| [161] | 1 | 3 | no |
| [161] | 1 | 2 | no |

Table 5.2: Summary of rates for multi-stage adaptive group testing when $k = \Theta(n^\alpha)$, depending on the number of stages and whether the error probability is zero (in which case the number of tests may be variable – the 'variable-$T$' setting).

## 5.3 Universality and counting defectives

A natural consideration in group-testing is that of *universality* – that is, how robust the paradigm of group testing is to lack of *a priori* knowledge of the number of defectives $k$ (or suitable upper and/or lower bounds). This question certainly arises quite naturally when nothing is known in advance of the statistical process generating the defectives.

A related issue, as discussed in Section 1.7, is that for certain applications the problem of interest is not to identify defective individuals $\mathcal{K}$, but merely to

estimate their total number $k = |\mathcal{K}|$. This focus may arise because identifying the individuals themselves is simply not necessary, is not desirable for reasons of privacy, or is not feasible because of the impracticality of distinguishing between test items (e.g., when studying insect populations).

The idea of counting the number of defectives using group testing dates back at least as far as work of Thompson [186], with these original ideas being developed in various works including (in chronological order) [179], [194], [35] and [185]. We first describe an argument based on random Bernoulli testing that forms part of many such defective-counting papers.

*Remark* 5.1. If there are $k$ defectives in some set $A$, and if each item in $A$ lies in a particular test independently with probability $p$, the test will be positive with probability

$$1 - (1-p)^k. \tag{5.1}$$

Inverting this relationship, using Bernoulli designs with probability

$$p(\ell) := 1 - 2^{-1/\ell}, \tag{5.2}$$

we would expect half the tests to be positive on average, if the true number of defectives $k$ was equal to $\ell$. Hence, using a Bernoulli design with parameter $p(\ell)$, if empirically we see many more (respectively, many fewer) than half the tests being positive, then this is evidence that $k \gg \ell$ (respectively, $k \ll \ell$).

We proceed by discussing the adaptive setting, and then turn to the non-adaptive setting. In both cases, we only consider noiseless group testing, since it has received by far the most attention in the relevant literature.

### 5.3.1 Adaptive testing

We first highlight the work of Cheng [38, Theorem 3.1] for adaptive testing, in which the following result was proved.

**Theorem 5.5** (Exact adaptive defective counting)**.** *For any parameter $c > 1$, there exists an adaptive algorithm that can find the exact number of defective items $k$ with error probability at most $1/k^{c-1}$, using a number of tests upper bounded by*

$$4k\left(\lceil c \log_2 k + c \rceil + 2\right). \tag{5.3}$$

Cheng's argument is based on a recursive binary splitting argument reminiscent of the binary search technique of Algorithm 1.1. Given a set $A$ that contains at least one defective, we randomly partition it via independent fair coin flips into two subsets $A_1$ and $A_2$ of roughly equal size. Note that this construction is equivalent to taking $p = 1/2$ in (5.1). Observe that:

1. If $A$ contains exactly one defective, one of $A_1$ and $A_2$ will certainly contain no defectives.

2. If $A$ contains $d > 1$ defectives, the probability that all of these defectives are placed in the same subset $A_i$ is $1/2^{d-1} \leq 1/2$.

In the following, we discuss a method for reliably distinguishing these two cases. Consider testing both subsets $A_1$ and $A_2$. In case 1, one of the two tests will

certainly be negative. In case 2, the probability that there is a negative test is at most $1/2$ (since this only occurs if all $d$ defectives lie in the same subset).

Furthermore, if we repeatedly and independently partition $A$ by using the above coin-tossing procedure $r$ times, the outcomes will be independent between partitions. Hence, our decision rule is simply to ask 'did we see $r$ negative tests?'. In case 1, this certainly happens, and in case 2, the probability that this happens is at most $1/2^r$. In other words, repeatedly randomly partitioning $A$ in this way allows us to efficiently distinguish cases 1 and 2. This allows us to eventually partition the full set of items $\{1, 2, \dots, n\}$ into subsets, each of which contains exactly one defective with high probability, so that $k$ is successfully identified.

A standard information-theoretic argument shows that, with no prior information on the number of defectives, to estimate the exact value of $k$ with zero error probability will require at least $\log_2(n+1)$ tests. In this sense, the question of the optimal order of the number of tests required for exact recovery remains open. Indeed, we note that individual testing of every item will give the exact number of defectives in $n$ tests, which outperforms Cheng's bound (5.3) in the regime where $k = \Theta(n)$.

However, as argued by Falahatgar *et al.* [74], the requirement to know the number of defectives *exactly* may be unnecessarily restrictive. They developed a four-stage adaptive algorithm, for which they proved that that an $O(\log \log k)$ expected number of tests achieves an *approximate recovery* criterion with high probability [74, Theorem 15]. The algorithm works by successively refining estimates, with each stage creating a better estimate with a certain probability of error. The first stage finds an estimated number of defectives that (with high probability) lies in the interval $(k, k^2)$, and later stages tighten this to $(k\epsilon^2, k/\epsilon^2)$ to $(k/4, 4k)$ to $((1-\delta)k, (1+\delta)k)$, using binary search techniques.

By tuning the performance of the algorithm of [74], Bshouty *et al.* [29] improved the performance guarantee by a constant factor, obtaining the following result [29, Theorem 8].

**Theorem 5.6** (Approximate adaptive defective counting)**.** *For any $c > 1$, $\delta \in (0,1)$ and $\epsilon \in (0,1)$, there exists a four-stage adaptive algorithm providing an estimate of $k$ that, with probability at least $1-\epsilon$, lies in the range $\big((1-\delta)k, (1+\delta)k\big)$, using an expected number of at most*

$$(1 - \epsilon + \epsilon^c) \log_2 (\log_2 k) + O\left(\sqrt{\log \log k}\right) + O\left(\frac{c}{\delta^2} \log \frac{1}{\epsilon}\right). \tag{5.4}$$

Taking $c$ arbitrarily large gives a leading term that is arbitrarily close to $(1-\epsilon)\log_2(\log_2 k)$. This result is essentially optimal, since [74, Theorem 16] previously used Fano's inequality to prove an information-theoretic lower bound showing that any such algorithm requires at least $(1-\epsilon)\log_2(\log_2 k) - 1$ tests.

A comparison between the $\log_2(n+1)$ converse result for exact recovery and the significantly tighter $O(\log \log k)$ approximate recovery bound of Theorem 5.6 indicates that the criterion for successful recovery consistently makes a significant difference in this case (in contrast to the more complicated situation described in Theorem 5.1 above). Note that since any subsequent group testing strategy to estimate $\mathcal{K}$ typically requires $\Theta\big(k \log_2 \frac{n}{k}\big)$ tests, it is highly desirable to use $o\big(k \log_2 \frac{n}{k}\big)$ tests in the defective-counting stage, and this is achieved in Theorem 5.6 but not Theorem 5.5 (unless $\log k = o(\log n)$, which is a much stronger requirement than the usual $k = o(n)$).

### 5.3.2 Nonadaptive testing

One challenge associated with creating nonadaptive algorithms to count defectives is that (without further information) any value of $k \in \{0, 1, \ldots, n\}$ is possible. Constructions based on (5.2) for a particular $p(\ell)$ will struggle to distinguish between putative values $k_1$ and $k_2$ that are both far from $\ell$. To overcome this, Damaschke and Muhammad [55, Section 4] proposed a geometric construction, dividing the range of tests into subintervals of exponentially increasing size and using a series of $p(\ell)$ values tailored to each subinterval.

To be more precise, the (nonadaptive) algorithm of [55] constructs parallel group tests indexed by integers $t$. Each item lies in the $t$-th test independently with probability

$$p_t := 1 - \left(1 - \frac{1}{n}\right)^{b^t}, \tag{5.5}$$

for some fixed positive $b > 1$. Again using (5.1), this means that each test is negative with probability $q_t := (1 - 1/n)^{b^t k}$. Hence, if $t^*$ is the largest index of a negative pool, we may imagine that $q_{t^*} \simeq 1/2$ and invert this to construct an estimate of $k$. More precisely, [55] propose an offset of this, taking

$$\widehat{k} = -\frac{1}{b^{t^*-s} \log_2\left(1 - \frac{1}{n}\right)}, \tag{5.6}$$

for an integer $s$. The following result [55, Theorem 4.3] shows that (5.6) satisfies certain success criteria.

**Theorem 5.7** (Approximate nonadaptive defective counting)**.** *For a given value*[1] $b \in (1, 2]$ *and integer* $s$*, the estimate* $\widehat{k}$ *of* (5.6) *can be formed using* $\log_b n$ *tests overall such that:*

1. $\mathbb{P}(\widehat{k} \leq k) = O\big(\frac{1}{2^{b^s} \log b}\big)$,
2. $\mathbb{E}(\widehat{k}/k) \leq b^s F(b)$ *for a certain explicit function* $F$*. This function* $F$ *is monotone increasing in* $b$*, with* $F(b) < 1.466$ *for all* $b \leq 2$*.*

Observe that the first part bounds the probability that the estimate $\widehat{k}$ underestimates $k$, and the second part shows that $\widehat{k}$ overestimates $k$ by at most an explicit constant factor on average.

In [54], the same authors use an argument based on a related hypothesis testing problem to show that this $\Omega(\log n)$ scaling in the number of tests is essentially optimal. Specifically, for nonadaptive group testing any estimate $\widehat{k}$ with a specified probability of underestimating $\mathbb{P}(\widehat{k} \leq k) \leq \epsilon$ and bounded 'competitive ratio' $\mathbb{E}(\widehat{k}/k) \leq c$ requires a multiple of $\log n$ tests, with a constant factor depending on $\epsilon$ and $c$ [54, Theorem 1].

In subsequent work, Bshouty [28] showed how to strengthen Theorem 5.7, building on the techniques developed in [55] and [74] to provide a one-sided estimate within a fixed factor. In particular, [28, Lemma 7] proved the following:

[1] We assume an upper bound $b \leq 2$ to simplify the theorem statement, but this could be replaced by any absolute constant.

**Theorem 5.8.** *Assuming the number of defectives satisfies $k \geq 6$, for any $\delta > 0$, we can form an estimate $\widehat{k}$ using $O\big(\log \frac{1}{\delta} \log n\big)$ tests, with the property that*

$$\mathbb{P}\left(k \leq \widehat{k} \leq 2k\right) \geq 1 - \delta. \tag{5.7}$$

This result can be extended to estimate $k$ to within any fixed constant factor. Bshouty [28, Section 3] also provide a counterpart to [54], namely, a lower bound on the number of tests required by randomized and deterministic algorithms to estimate $k$ to within a constant factor.

Returning to the question of universality of group testing (that is, whether we can recover the defective set $\mathcal{K}$ with no prior knowledge of $k$), we can regard Bshouty's algorithm [28] as the first (pre-processing) stage of a two-stage universal algorithm. In the second stage, we can use the resulting estimate of $k$ to determine $\mathcal{K}$ using (for example) the COMP algorithm under Bernoulli testing with parameter $p = 1/\widehat{k}$.

We know that COMP attains a positive rate when $\widehat{k}$ is within a constant factor of $k$ (see Remark 2.3). We successively apply Bshouty's result (Theorem 5.8) to a sequence of problems with $n \to \infty$ items, taking $\delta = \delta_n$ to zero sufficiently slowly. For example, with $\delta_n = 1/n$ the initial stage requires $O\left((\log n)^2\right)$ to estimate $k$ to within a constant factor with probability approaching one. The number of the tests in this first stage is therefore negligible compared to the $\Theta\big(k \log \frac{n}{k}\big)$ requirement of the second stage when $k \gg \log n$.

The preceding results (both adaptive and nonadaptive) are summarized in Table 5.3.

| References | Recovery guarantee | #Tests | Adaptive? |
|---|---|---|---|
| [38] | Exact | $O(k \log k)$ | yes |
| [74, 29] | $(1-\delta)k \leq \widehat{k} \leq (1+\delta)k$ | $O(\log \log k)$ | yes |
| [56] | $\widehat{k} \geq k$, $\mathbb{E}[\widehat{k}/k] = O(1)$ | $O(\log n)$ | no |
| [28] | $k \leq \widehat{k} \leq 2k$ | $O(\log n)$ | no |

Table 5.3: Summary of recovery guarantees for counting defectives, depending on the recovery criteria and availability of adaptive testing. The results here correspond to the case of a constant non-zero error probability; the precise dependencies on the error probability can be found in the above theorem statements.

### 5.3.3 Discussion

It is worth mentioning a fundamental limitation regarding the quest for universality: If we require the test design to be completely nonadaptive, then achieving a positive rate when $k = \Theta(n^\alpha)$ for some $\alpha$ precludes achieving a positive rate for $\alpha' < \alpha$. This is because the former requirement needs $n = \Omega(n^\alpha \log n)$ by the counting bound, and any such scaling on $n$ gives zero rate for $\alpha' < \alpha$. Hence, having multiple stages of adaptivity (or being fully adaptive) is essential to universally attaining a positive rate.

On the other hand, if the number of defectives is known to be upper bounded by some known value $k^*$, then under Bernoulli testing with $p = 1/k^*$, the analysis of COMP (see the Appendix to Chapter 1) leads to vanishing error probability with $T = O(k^* \log n)$ tests, even if the true value of $k$ is much smaller than $k^*$. More specifically, this can be seen by setting $p = 1/k^*$ in (2.9), upper bounding $k \leq k^*$, and continuing the analysis with $k^*$ in place of $k$. Therefore, we still have guarantees on the performance when only an upper bound on $k$ is known, but we pay a penalty in the number of tests if that bound is loose.

The above notion of universal group testing can be viewed as a counterpart to information-theoretic problems of universal source coding, where one does not have access to the underlying distribution of the source. A prominent example of an adaptive universal source coding algorithm is that of Lempel and Ziv, which achieves asymptotically optimal compression for any stationary ergodic source by parsing strings into trees (e.g., see [49, Chapter 13], [134, Section 6.4]).

Of course, there may be uncertainty regarding the group testing setup in a broader sense than simply a lack of knowledge of the number of defectives $k$. For example, in the noisy settings of Chapter 4.5, we may lack information regarding the parameters of the particular group testing model, or even which model applies. The problem of decoding with no knowledge of the model was referred to as 'blind group testing' in the recent work of Huleihel *et al.* [105], who proved universal variants of the information-theoretic joint and separate decoding rules (see Sections 4.5 and 3.5 respectively). In the sparse regime with $\alpha \to 0$, they show that their decoders achieve the same asymptotic performance as when the channel is known. An earlier work of Malyutov and Sadaka [142] showed such a result in the very sparse regime $k = O(1)$.

Again, we can connect this with classical information-theoretic results, namely, for the problem of universal channel coding. Here a transmitter seeks to send a message over a noisy channel, despite not having precise channel statistics other than a guarantee that its capacity exceeds the transmission rate. The maximum empirical mutual information decoder (e.g., see [50, p. 100]) is the most well-known decoding method for this scenario, and such a decoder was in fact adopted for group testing in [142]. The decoder adopted in [105] is slightly different, but still based on empirical probability measures.

## 5.4 Sublinear-time algorithms

The decoding algorithms such as COMP and DD (but not SSS) studied in Chapters 2 and 3 are efficient, in the sense that they operate in time $O(nT)$. However, in the case that $n$ is extremely large, or practical considerations require a very fast decoding rule, it is of interest to seek algorithms that reduce this runtime further.

To address this question, a recent line of works has considered *sublinear-time algorithms* that run in time that is linear or polynomial in $k \log n$, rather than in $n$. Since even specifying the defective set requires $k \log_2 n$ bits, one cannot improve on $O(k \log n)$ scaling.

Early works on sublinear-time decoding for group testing focused on the zero-error recovery criterion [39, 110, 153], possibly with adversarial noise [39].

Typical results in these works state that with $T = O(k^2 \log n)$ tests (which is nearly order-optimal under the zero-error recovery criterion; see Section 1.6), one can attain a decoding time of the form $O(T^c)$ for some $c > 1$.

Our focus in this manuscript is on the small-error recovery criterion, as opposed to zero-error, and we therefore focus on more recent algorithms targeted at this setting. We immediately see that we cannot rely on the main ideas used in Chapters 2 and 3; for instance:

- Merely traversing the entries of a Bernoulli test matrix requires $\Omega(nT)$ time. Structured test matrices often permit faster computation, but the only structure inherent in the Bernoulli($p$) case is sparsity, and there are still $\Omega(n \log n)$ entries equal to one in the case that $T = \Omega(k \log n)$ and $p = \Theta\big(\frac{1}{k}\big)$.
- Both COMP and DD use the idea of marking items as nondefective if they appear in a negative test. However, individually marking $n - k$ items (or even just a constant fraction thereof) as nondefective already requires linear time.

The first observation suggests either using a highly structured test design, in particular adopting a scheme where the decoder does not need to read the entire test matrix. The second observation suggests that achieving sublinear runtime requires using a decoding algorithm that positively identifies defective items, rather than ruling out nondefective items.

We note that the notion of 'sublinear time' here applies purely to the decoding. Indeed, if we assume that placing an item in a test takes constant time, then *encoding* obviously requires at least linear time in total, if each item is tested at least once. Thus, sublinear time algorithms will be of interest in the case that the encoding work naturally parallelizes over the $n$ elements, and/or the decoding time poses a bottleneck.

In this section, we discuss two schemes for group testing with sublinear-time decoding time and arbitrarily small error probability:

**SAFFRON** [126] is a scheme for nonadaptive noiseless testing based on sparse-graph codes. It requires a number of tests and runtime both of order $O(k \log k \log n)$. We also discuss a partial recovery result, and briefly mention a variant for the noisy setting.

**GROTESQUE** [32] is a scheme with three variants: adaptive, two-stage, and nonadaptive. All three variants work for both the noiseless and noisy settings. The adaptive variant performs $O(k \log n)$ tests and requires $O(k \log n)$ runtime, both of which are the optimal scaling, in particular amounting to a positive rate. In addition, the two-stage variant achieves a positive rate for some scaling regimes of $k$.

At the end of the section, we also discuss some more recent approaches that are known to achieve a positive rate for exact recovery in the nonadaptive setting, unlike SAFFRON and GROTESQUE.

To the best of our knowledge, GROTESQUE appeared in the literature prior to SAFFRON. However, we find it natural to first present a simplified form of SAFFRON that is the easiest to analyze, and then move on to the various forms of GROTESQUE.

### 5.4.1 SAFFRON

SAFFRON is a scheme for nonadaptive group testing with sublinear decoding time due to Lee, Pedarsani and Ramtin [126]. It is based on sparse-graph codes, which were also used earlier in other sparse signal recovery problems (e.g., see [129]). We present here the simple 'singleton-only SAFFRON' version of the scheme.

The basic idea is as follows: The $T$ tests are split into 'bundles' of size $2m$, where $m = \lceil \log_2 n \rceil \sim \log_2 n$. Each item is chosen to either 'appear' or 'not appear' in each bundle. Items appear in bundles according to an outer Bernoulli design, where each item appears in each bundle independently with probability $p$.

If item $i$ it does not appear in a given bundle, then it is absent from all $2m$ tests in that bundle. If item $i$ does appear in a given bundle, it is placed in the tests within that bundle that correspond to the 1s in the vector $\big(\mathbf{b}(i), \overline{\mathbf{b}(i)}\big) \in \{0,1\}^{2m}$, where $\mathbf{b}(i) \in \{0,1\}^m$ is the binary expansion of the number $i$, and $\overline{\mathbf{b}(i)} = \mathbf{1} - \mathbf{b}(i)$ is $\mathbf{b}(i)$ with the 0s and 1s reversed. Note that since the vector $\big(\mathbf{b}(i), \overline{\mathbf{b}(i)}\big)$ always has weight $m$, any item appearing in a bundle is in exactly $m$ tests within that bundle. The idea of encoding binary expansions into tests/measurements was also used earlier in studies of sparse recovery with linear measurements (e.g., see [93] and the references therein).

The key to SAFFRON is in the decoder. When considering the outputs from a given bundle of $2m$ tests, we first look at the the weight of those outputs – that is, the number of positive outcomes within the bundle. If the weight is 0, then no defective items appeared in the bundle; if the weight is exactly $m$, then precisely one defective item appeared in the bundle; if the weight is greater than $m$, then two or more defective items appeared in the bundle. The simplified (singleton-only) SAFFRON decoder considers only those bundles containing precisely one defective. The first $m$ outputs from such a bundle give the binary expansion of the label of the defective item, which is therefore immediately identifiable. Repeating this process for each bundle collects a number of defective items, which form our estimate of the defective set.

The key point here is that the SAFFRON decoder first *detects* a bundle containing exactly one defective item, by calculating the output weight, then affirmatively *identifies* that defective item, using the binary expansion. Thus, it does not rely on ruling out nondefective items, which inevitably takes at least linear time. Note also that while it would take more than linear time to read the Bernoulli outer design, it is not necessary to do so, as defective items identify themselves through their binary expansion.

We then have the following result.

**Theorem 5.9.** *Consider standard nonadaptive group testing with $n$ items, $k$ defectives, and $T$ tests. Singleton-only SAFFRON succeeds at exact recovery with vanishing error probability provided that*

$$T \geq (1+\eta)\, 2\mathrm{e}\, k \ln k \log_2 n$$

*for some $\eta > 0$, and the decoding time is $O(k \log k \, \log n)$.*

*Furthermore, we have the following partial recovery result: SAFFRON finds at least $(1-\gamma)k$ defective items and no false positives with error probability*

*tending to* 0 *provided that*

$$T \geq (1+\eta)\, 2\mathrm{e}\, k \ln\left(\frac{1}{\gamma}\right) \log_2 n$$

*for some* $\eta > 0$*, and the decoding time is* $O(k \log(1/\gamma) \log n)$*.*

The first part of the above theorem is [126, Theorem 4.2]. The second part is not explicitly stated there, but follows from the same argument.

We briefly discuss how the above statements on the number of tests translate into achievable rates when $k = \Theta(n^\alpha)$ with $\alpha \in (0,1)$. In the case of exact recovery, the number of tests is a logarithmic factor higher than the optimal scaling, so the rate is zero. However, the partial recovery result has a positive rate for fixed $\gamma \in (0,1)$, namely,

$$\frac{1}{2\mathrm{e} \ln \frac{1}{\gamma}}(1-\alpha) \simeq \frac{0.184}{\ln \frac{1}{\gamma}}(1-\alpha).$$

We observe also that this rate tends to zero as $\gamma \to 0$.

*Proof sketch.* We sketch a proof based on the coupon collector problem. Write $B$ for the number of bundles, recalling that $T = 2mB \sim 2B \log_2 n$. By picking the Bernoulli parameter as $p = 1/k$, we maximize the average number of bundles containing exactly one defective, and by standard concentration bounds, the actual number is close to the resulting average $\mathrm{e}^{-1}B$ with high probability.

For exact recovery, we need to 'collect' all $k$ defective items. The standard coupon collector problem states that we require $k \ln k$ such bundles to collect the $k$ items (see Remark 2.2). Hence, we need $\mathrm{e}^{-1}B \sim k \ln k$, and thus $T \sim 2\mathrm{e}\, k \ln k \, \log_2 n$.

For the partial recovery criterion, another standard coupon collector result states that that collecting $(1-\gamma)k$ coupons out of $k$ requires $k \ln(1/\gamma)$ such bundles. The result then follows in the same way.

For each bundle, the outputs are read in time $O(\log n)$, the weight computed in time $O(\log n)$, and the single defective – if there is one – identified in time $O(\log n)$. Hence, the running time for the decoder is $O(B \log n) = O(T)$, which is $O(k \log k \log n)$ for exact recovery and $O(k \log(1/\gamma) \log n)$ for partial recovery. □

We re-iterate that the above result concerns the simplified 'singleton-only' SAFFRON scheme; the full SAFFRON scheme of [126] improves the constant factors in the results of Theorem 5.9 as follows: When a bundle contains two defective items, one of which has been identified elsewhere (e.g., via the singleton approach), the second defective can be then also be identified with high probability. The outer Bernoulli design is also replaced by a design with constant bundles-per-item.

In addition, the authors of [126] give a 'robustified SAFFRON' algorithm for noisy group testing. Here, the vectors $\big(\mathbf{b}(i), \overline{\mathbf{b}(i)}\big)$ are extended with the parity-check bits of a positive-rate code in order to provide robustness to noise. The resulting scheme is similar to the nonadaptive variant of GROTESQUE described below.

### 5.4.2 GROTESQUE

In this subsection, we give an overview of another sublinear-time algorithm called GROTESQUE (Group Testing, Quick and Efficient) due to Cai, Jahangoshahi, Bakshi and Jaggi [32]. This approach uses expander codes [182] in its construction, thus highlighting that efficient channel codes (e.g., see [158]) can play a role in practical group testing constructions, and complementing the extensive use of information theory for theoretical studies of group testing.

#### Overview of results

There are three variations of GROTESQUE with different guarantees on the number of tests and decoding time, corresponding to the fully adaptive, two-stage adaptive, and nonadaptive settings. We first summarize the respective performance guarantees, and then give an overview of the algorithms themselves.

**Theorem 5.10.** *There exists an adaptive variant of GROTESQUE using $O(\log n)$ stages of adaptivity that achieves vanishing error probability, performs $O(k \log n)$ tests and requires $O(k \log n)$ decoding time.*

Observe that this result attains the best scaling laws possible when $k = \Theta(n^\alpha)$ with $\alpha < 1$. In particular, the algorithm achieves a positive rate; however, the rate itself may be low according to the existing proof, which does not optimize the constant factors.

**Theorem 5.11.** *There exists a nonadaptive variant of GROTESQUE that achieves vanishing error probability with $O(k \log n \log k)$ tests and $O(k(\log n + \log^2 k))$ decoding time.*

Observe that the number of tests matches that of SAFFRON up to constant factors. In particular, although the rate is zero, the scaling laws are only a log factor away from optimality. While it may seem unusual for the number of tests to exceed the decoding time (e.g., when $k = O(\log n)$), the idea is that the algorithm can 'adaptively' decide which test outcomes to observe, and ultimately leave some tests unobserved. (We implicitly assume that fetching the result of a given test can be done in constant time.)

**Theorem 5.12.** *A two-stage adaptive variant of GROTESQUE achieves vanishing error probability with $O(k(\log n + \log^2 k))$ tests and $O(k(\log n + \log^2 k))$ decoding time.*

This result improves on the number of tests used by the nonadaptive algorithm, and achieves a positive rate whenever $\log^2 k = O(\log n)$. Note, however, that this condition is not satisfied in the regime $k = \Theta(n^\alpha)$ (with $\alpha \in (0, 1)$), which has been the focus of most of this monograph.

We briefly mention that all of the above guarantees hold not only in the noiseless setting, but also for the symmetric noise model with a fixed crossover probability in $\left(0, \frac{1}{2}\right)$.

#### Overview of the algorithm variants

The basic building block of all three variants of the algorithm are two types of 'tests' (in the general sense of the word, rather than the sense of a single group test) that operate on subsets of $\{1, \ldots, n\}$, described as follows.

**Multiplicity test** A *multiplicity test* considers a subset $\mathcal{S}$ of size $n' < n$, and only seeks to establish whether the number of defective items in the subset is 0, 1, or more than 1. To do this, we perform $T_{\text{mul}} = O(\log n)$ tests, where each item in $\mathcal{S}$ is included in each test independently with probability $\frac{1}{2}$. It is easy to show (see also Remark 5.1 above) that the following holds for each such test:

- If $\mathcal{S}$ has no defective items, the output must be 0 (noiseless case), or have a probability strictly less than $\frac{1}{2}$ of being 1 (noisy case).
- If $\mathcal{S}$ has one defective item, the output is equal to 0 or 1 with probability exactly $\frac{1}{2}$ each.
- If $\mathcal{S}$ has more than one defective item, then the output equals 1 with probability at least $\frac{3}{4}$ (noiseless case), or with probability strictly higher than $\frac{1}{2}$ (noisy case).

Therefore, by standard concentration bounds, we can correctly categorize $\mathcal{S}$ into these three categories with probability at least $1 - O\big(\frac{1}{n^c}\big)$ (for any fixed $c > 0$) using $O(\log n)$ tests.

Notice that the reason that this procedure can be done efficiently is that we pay no attention to which items are included in each test; we merely count the number of positive and negative test outcomes.

**Location test** After a multiplicity test is performed on a subset $\mathcal{S}$, we only apply this step to $\mathcal{S}$ if the set is found to contain exactly one defective item. If this is indeed the case, we perform a *location test* to deduce the index $i \in \{1, \dots, n\}$ of that defective item.

To do this, we perform another set of $T_{\text{loc}} = O(\log n)$ tests on $\mathcal{S}$, but this time we use a structured 'test sub-matrix' of size $T_{\text{loc}} \times |\mathcal{S}|$. Specifically, we let the columns of this matrix be the codewords of an expander code [182]. We do not give the details of such a code, but instead highlight its desirable properties:

- The decoding time is linear in the block length (i.e., $O(\log n)$);
- The error probability decays exponentially in the block length (i.e., $O(n^{-c})$ for some $c > 0$);
- The rate is constant (i.e., an item index in $\{1, \dots, n\}$ can be reliably identified with a block length $O(\log n)$);
- The code is robust to independent random bit flips (i.e., symmetric noise) and/or erasures.

These properties suffice to perform a single location test sufficiently reliably using $O(\log n)$ tests and $O(\log n)$ runtime, even in the presence of random noise.

In SAFFRON above, the bundles consisting of binary expansions vectors work a lot like the multiplicity and location step here: A bundle having output weight exactly $m$ certifies that it contains exactly one defective (multiplicity test with $T_{\text{mul}} = 2m \sim 2\log_2 n$), and reading off the binary expansion 'localizes' the defective item (with no extra steps).

With the preceding building blocks in place, we can now describe the three variations of the algorithm.

**Adaptive algorithm** The adaptive algorithm uses $J = O(\log n)$ stages of adaptivity, with all stages except the last using a common procedure. Specifically, the goal of the first $J-1$ stages is to identify all except at most $\log_2 k$ defective items. Letting $k_i$ denote the number of unresolved defective items before the $i$-th stage, we randomly partition the $n - k + k_i$ items (excluding resolved defectives) into $2k_i$ groups, and perform a multiplicity test on each such group.

By standard concentration via McDiarmid's inequality [147], it can be shown that with high probability, a constant fraction of the groups contain a single defective item. Assuming the multiplicity tests are successful (which occurs with high probability), all such groups are identified, and the corresponding defective item can then be found via a location test.

When the number of remaining defectives $k_i$ falls far below $\log_2 n$, the desired concentration behaviour starts to fail to hold. To address this, in the final stage, we form $O((\log k)^2 \log\log k)$ groups, each containing $O\big(\frac{n}{\log k}\big)$ unresolved items chosen uniformly at random. These choices of scaling laws, with suitably-chosen implied constants, ensure that each unresolved defective appears in at least one group by itself with high probability. As a result, we can identify these remaining items via multiplicity and location tests as above.

By the fact that the number of unresolved defectives decays geometrically in the first $J-1$ stages, it can be shown that these stages collectively only require $O(k \log n)$ tests and runtime. The final stage requires $O((\log k)^2 \log\log k \log n)$ tests and runtime, which is strictly smaller than $O(k \log n)$.

**Nonadaptive algorithm** The simplest way to make the algorithm nonadaptive is to note that, since the first stage identifies a random constant fraction of the defective items, repeating that stage independently $O(\log n)$ times is enough to identify all defective items with high probability. This approach gives the $O(k \log k \log n)$ number of tests stated in Section 5.4.2, but requires $O(k \log k \log n)$ runtime instead of the improved $O(k(\log n + \log^2 k))$. To achieve the latter, one adopts a more sophisticated approach that adaptively chooses which tests to observe; we refer the reader to [32] for details.

**Two-stage algorithm** Once the previously-mentioned guarantees of the nonadaptive algorithm are in place, analysing the two-stage algorithm is straightforward. In the first stage, one randomly partitions the $n$ items into $k^3$ bins. Since there are $k$ defective items, a standard 'birthday paradox' argument (see for example [77, p. 33]) reveals that with high probability, each such bin contains either 0 or 1 defective items.

In the first stage, a 'grouped group testing' procedure is applied with $k^3$ 'super-items'. Each super-item corresponds to an entire bin, and testing a super-item amounts to including all the bin's items in the test simultaneously. Using the above nonadaptive algorithm accordingly with $k^3$ in place of $n$, we see that we can reliably identify the $k$ defective bins using $O(k \log^2 k)$ tests and runtime. In the second stage, we simply apply the location test separately to each defective bin, thereby identifying the $k$ defective items using $O(k \log n)$ tests and runtime.

**Discussion**

The GROTESQUE algorithm, as described above, assumes exact knowledge of $k$. However, as highlighted in [32], it can be adapted to the case that $k$ is only known up to a constant factor. As we saw in Section 5.3, such knowledge can be attained with probability at least $1 - \delta$ using one extra stage of adaptivity with only $O\big(\log n \log \frac{1}{\delta}\big)$ tests (Theorem 5.8).

A subtle point is that the computational requirements of translating a codeword in the expander code to an index in $\{1, \dots, n\}$ were ignored above. The justification for this is that the required computation can be done as *preprocessing*, which is done completely offline. Specifically, one can construct a binary tree whose branches correspond to bits in the codeword, and whose leaves are given the appropriate labels in $\{1, \dots, n\}$. This tree can be constructed in time $O(n)$. Given access to this tree and a codeword of interest, the decoder can use the codeword bits to traverse the tree and find the corresponding label at the leaf in time $O(\log n)$. (The above-described version of SAFFRON, on the other hand, does not require preprocessing, and its bit-expansion approach could be incorporated into GROTESQUE in the noiseless case.)

### 5.4.3 Attaining a positive rate

A notable limitation of the theoretical guarantees of SAFFRON and the non-adaptive variant of GROTESQUE is that the number of tests is $O(k \log k \log n)$, meaning that the rate is zero unless $k = O(1)$. Here we briefly highlight two recent works that improved the number of tests at the expense of a higher decoding time.

In [108], a classical construction of Kautz and Singleton [119] was adapted from the zero-error setting to the small-error setting, and was shown to permit exact recovery with $T = O(k \log n \log \frac{\log n}{\log k})$ tests and $O(k^3 \log n \log \frac{\log n}{\log k})$ decoding time. The Kautz-Singleton construction is a type of *concatenated code*, and will be discussed in more detail in Section 5.7. The preceding number of tests amounts to a positive rate whenever $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, but not in sparser regimes such as $k = O((\log n)^c)$ (for fixed $c > 0$).

The problem of attaining exact recovery with $T = O(k \log n)$ and sublinear decoding time without further assumptions on $k$ was recently solved in [26] via an approach termed *bit-mixing coding* (BMC). This technique tests random 'bundles' of items analogously to SAFFRON; the distinction is that instead of seeking to ensure that each defective is the unique one in some bundle corresponding to $O(\log n)$ tests, BMC allows each defective item's index to be encoded in $O(\log n)$ tests with *collisions* between the different defective items. As long as a constant fraction of these tests remains collision-free for each item, the collisions can be controlled using erasure-correcting coding techniques; see [26] for details. The decoding time of BMC is $O(k^2 \log k \log n)$, which improves on that of [108], but remains higher than that of SAFFRON and GROTESQUE by a factor of $k$.

While the $T = O(k \log n)$ guarantee of BMC amounts to a positive rate whenever $k \leq n^{1-\eta}$ for arbitrarily small $\eta > 0$, no effort was made to optimize the constant factors in [26]. As a result, attaining a rate comparable with that of COMP, DD, etc. with sublinear-time decoding still remains an interesting open problem.

The results surveyed in this section are summarized in Table 5.4.

| | #Tests | Decoding time | Adaptive? |
|---|---|---|---|
| **SAFFRON** | $O(k \log k \log n)$ | $O(k \log k \log n)$ | no |
| **GROTESQUE** | $O(k \log k \log n)$ | $O(k \log n + k \log^2 k)$ | no |
| **2-GROTESQUE** | $O(k \log n + k \log^2 k)$ | $O(k \log n + k \log^2 k)$ | two-stage |
| **A-GROTESQUE** | $O(k \log n)$ | $O(k \log n)$ | yes |
| **Kautz-Singleton** | $O\big(k \log n \log \frac{\log n}{\log k}\big)$ | $O\big(k^3 \log n \log \frac{\log n}{\log k}\big)$ | no |
| **BMC** | $O(k \log n)$ | $O(k^2 \log k \log n)$ | no |

Table 5.4: Summary of number of tests and decoding times for sublinear-time group testing algorithms in the small-error setting. '2-GROTESQUE' and 'A-GROTESQUE' refer to the two-stage and fully adaptive versions of GROTESQUE, and the other rows are as described above.

## 5.5 The linear regime

For the majority of this survey, we have focused on the sparse regime, where the number of defective items $k$ scales as $k = o(n)$, specifically $k = \Theta(n^\alpha)$ with $\alpha < 1$. However, for many real-world applications, it may be more realistic to assume that each item has a constant probability of being defective as $n \to \infty$, rather than tending to 0. For example, we might assume each soldier has a probability $\beta$ of having syphilis, but we would not expect this probability to decrease as more soldiers join. This is the regime we consider in this section.

In short, we are interested in the asymptotic behaviour of group testing where $k = \Theta(n)$. It will turn out that, in contrast to the sparse regime $k = \Theta(n^\alpha)$ with $\alpha < 1$, the constant term in front of the $n$ is important; thus, we will consider a limiting regime where $k \sim \beta n$, by which we mean that $k/n \to \beta$, for some constant $\beta \in (0, 1)$.

The theory of group testing in this regime turns out to be decidedly different to the sparse regime studied throughout the monograph. Note that in contrast to (1.11), the term $\log_2 \binom{n}{k}$ from the counting bound (Theorem 1.1) behaves in this regime as [47, p. 1187]

$$\log_2 \binom{n}{k} \sim nh\left(\frac{k}{n}\right) \sim nh(\beta), \tag{5.8}$$

which is linear in $n$. (Here, as before, $h$ is the binary entropy.) Hence, for algorithms having a nonzero rate, we seek a number of tests scaling as $T = O(n) = O(k)$. In this regime, algorithms and designs requiring $T = \Omega(k \log n)$ tests, as we found before, will have rate 0. Moreover, we see here that simply testing each item individually in $T = n$ tests gives a positive rate of

$$\frac{\log_2 \binom{n}{k}}{T} = \frac{\log_2 \binom{n}{k}}{n} \to h(\beta). \tag{5.9}$$

In fact, under the combinatorial prior (see Section 1.1 and the Appendix to Chapter 1), where $k$ is known, we only require $T = n - 1$ tests, since we will

know whether or not the final item is defective by whether we have found $k-1$ or $k$ defectives so far. This still has rate $h(\beta)$, of course. Henceforth, we use the word 'optimal' to mean 'has optimal rate' to avoid considering such 'second-order' behaviour.

Combining (5.9) with the counting bound, we see that the capacity $C = C(\beta)$ (or zero-error capacity $C_0$) of group testing in the linear regime is bounded by $h(\beta) \leq C(\beta) \leq 1$.

In fact, we shall see in this section that for nonadaptive testing, individual testing is optimal, and so we have equality with the lower bound $C(\beta) = C_0(\beta) = h(\beta)$. Furthermore, even for adaptive testing, individual testing is optimal for large $\beta$, although it can be improved on for small $\beta$.

In the rest of this section, we briefly discuss results for the following four types of group testing. Recall that under the combinatorial prior we have exactly $k = k(n)$ defectives, with $k/n \to \beta$, while under the i.i.d. prior each item is independently defective with probability $\beta$.

**Nonadaptive zero-error combinatorial** Individual testing is optimal for all $\beta \in (0,1)$, so the capacity is $C_0(\beta) = h(\beta)$.

**Nonadaptive small-error i.i.d.** Individual testing is optimal for all $\beta \in (0,1)$, so the capacity is $C(\beta) = h(\beta)$. [9]

**Adaptive zero-error combinatorial** Individual testing is optimal for $\beta \geq 1 - \log_3 2 \approx 0.369$ giving $C_0(\beta) = h(\beta)$, and this is conjectured to be true for all $\beta \geq 1/3$. For $\beta < 1/3$, there are algorithms giving rates of at least $0.9 > h(\beta)$, so individual testing is suboptimal. [157, 104, 10]

**Adaptive small-error i.i.d.** Let $\beta^* = (3-\sqrt{5})/2 \approx 0.382$. Individual testing is optimal for $\beta \geq \beta^*$ giving $C(\beta) = h(\beta)$. For $\beta < \beta^*$, there are algorithms giving rates of at least $0.95 > h(\beta)$, so individual testing is suboptimal. [79, 10]

### 5.5.1 Nonadaptive testing

We begin with the nonadaptive cases. That individual testing is optimal for the nonadaptive zero-error combinatorial setting follows immediately from the results discussed in Section 1.6. That the same is true for the small-error setting with an i.i.d. prior was shown by Aldridge [9]. (This result improved on an earlier converse result of Agarwal, Jaggi and Mazumdar [2].)

**Theorem 5.13.** *Consider nonadaptive group testing with an i.i.d. prior where each of the $n$ items is independently defective with a given probability $\beta \in (0,1)$, independent of $n$. Suppose we use $T < n$ tests. Then there exists a constant $\epsilon = \epsilon(\beta) > 0$, independent of $n$, such that the average error probability is at least $\epsilon$.*

The key idea of [9] is the following: Suppose some item $i$ is *totally disguised*, in that every test containing $i$ also contains a defective item distinct from $i$. Then every test containing $i$ is positive, regardless of whether $i$ is defective or nondefective. Thus, we cannot know whether $i$ is defective or not: We either guess that $i$ is defective, and are correct with probability $\beta$; guess that $i$ is nondefective, and are correct with probability $1-\beta$; or take a random choice

between the two. In any case, the error probability is bounded below by the constant $\min\{\beta, 1-\beta\}$, which is nonzero for $\beta \in (0,1)$. Thus, we can attain a converse bound by showing that, again with probability bounded away from 0, there is such a totally disguised item $i$. The probability that item $i$ is totally disguised is bounded by

$$\mathbb{P}(D_i) \geq \prod_{t\,:\,x_{ti}=1} \left(1-(1-\beta)^{w_t-1}\right), \tag{5.10}$$

where $w_t$ is the number of items in test $t$ (that is, the weight of test $t$). The bound (5.10) can easily be shown using the FKG inequality [9, Lemma 4], and an elementary but longer proof is given in [2, Lemma 4].

The proof of Theorem 5.13 given in [9] first shows that when $T < n$ one can assume, without loss of generality, that there are no tests of weight 0 or 1. It then uses (5.10) to show that, for any design with $w_t \geq 2$ for all $t$, the mean probability $\mathbb{P}(D_i)$ of an item being totally disguised, averaged over all items $i$, is bounded away from 0. Hence, *some* item certainly has a probability of being totally disguised that is bounded away from zero, thus proving the theorem.

While Theorem 5.13 shows that the error probability is bounded away from 0, the given bound on the error probability is very small for small $\beta$, say $\beta < 0.1$ [9]. Thus, in applications, for a given *finite* value of $n$, it might be that a desired small error tolerance can be achieved with fewer than $n$ tests. Further work on 'finite size' group testing might help resolve this. For example, Wadayama [192] suggests the use of doubly regular designs may be useful in this regime (though some results of [192] were later reported as incorrect [193, 9]).

### 5.5.2 Adaptive testing

We now turn to adaptive testing. Following Aldridge [10], we can look at generalized binary splitting algorithms, such as that of Hwang [106] described in Section 1.5. Here we consider the following variant.

**Algorithm 5.1.** The following algorithm finds the defectivity status of every item in a set. The algorithm depends on a parameter $m$; we pick $m = 2^s$ to be a power of 2 here for convenience.

1. Choose a subset of $m$ items – say, the first $m$ items. Test this set.

   (a) If the test is negative, we find $m = 2^s$ nondefective items in 1 test.

   (b) If the test is positive, perform binary splitting (Algorithm 1.1); we find 1 defective item and between 0 and $m-1$ nondefective items in $1 + \log_2 m = s + 1$ tests.

   Remove all items whose whose status was discovered in this step.

2. Repeat Step 1 until no items remain.

We proceed by discussing the zero-error and small-error settings separately.

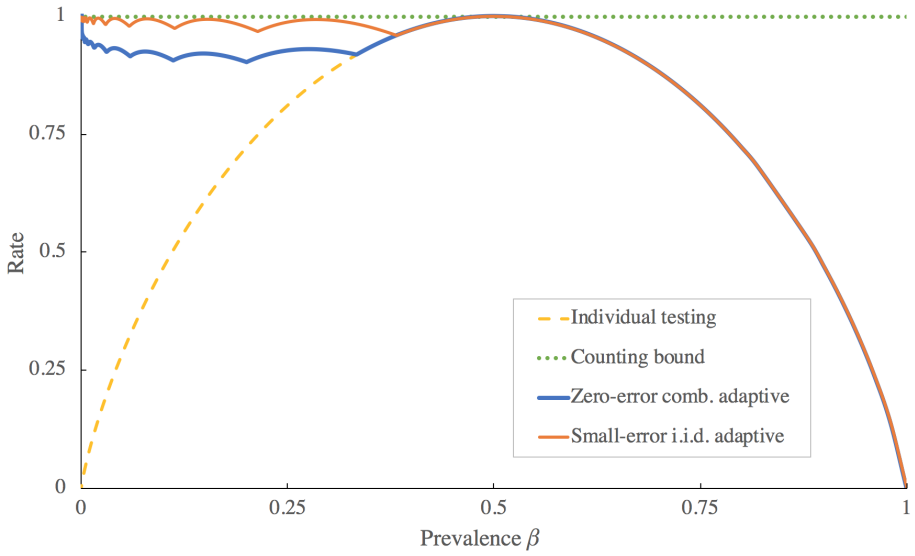



Figure 5.2: Rates of group testing in the linear regime $k \sim \beta n$: an achievable rate of combinatorial zero-error adaptive testing (5.11); an achievable rate of small-error adaptive testing with an i.i.d. prior (5.13); the rate $h(\beta)$ of individual testing, which is the capacity of nonadaptive testing (Theorem 5.13); and the counting bound $C \leq 1$.

**Zero-error combinatorial setting** For the purposes of the zero-error criterion, we take a worst-case analysis and assume that we are always unlucky in step 1(b) and find 0 nondefective items. Thus, in each stage we find either one defective in $s+1$ tests or $m = 2^s$ nondefectives in one test. We see that the number of tests required is at most

$$T \sim (s+1)k + \frac{1}{2^s}(n-k) \sim \left(\frac{1}{2^s} + \left(s + 1 - \frac{1}{2^s}\right)\beta\right) n. \tag{5.11}$$

The number of tests in (5.11) is linear in $\beta$ for fixed $s$, but becomes only piecewise linear after choosing the optimal value of $s$ for each $\beta$, since $s$ must be an integer. The resulting rate achieved by this algorithm is shown in Figure 5.2. After converting the piecewise linear $T$ into a rate, we now observe 'bumps' with endpoints at locations where the optimal value of $s$ in (5.11) changes. We obtain a rate of at least 0.9 for all $\beta \leq 1/2$.

Setting $m = 2^0 = 1$ recovers individual testing, requiring $T = n$ tests. In addition, setting $m = 2^1 = 1$, we see that we outperform individual testing when

$$\frac{1}{2} + \left(1 + 1 - \frac{1}{2}\right)\beta < 1$$

which is precisely when $\beta < 1/3$. That individual testing is suboptimal for $\beta < 1/3$ was first shown by Hu, Hwang and Wang [104], and was also noted in [79]. Hu, Hwang and Wang [104] conjecture that this is the best possible, in the sense that individual testing is optimal for $\beta \geq 1/3$. The best result in this

direction is by Riccio and Colbourn [157], who show that individual testing is optimal for $\beta \geq 1 - \log_3 2 \approx 0.369$.

**Small-error i.i.d. setting** By analysing the average-case behaviour of this binary splitting algorithm, it can be shown that step 1(b) learns the status of

$$\begin{aligned} E &= \sum_{j=1}^{m} j\beta(1-\beta)^{j-1} + m(1-\beta)^m \\ &= \frac{1}{\beta}\big(1 + m(1-\beta)^{m+1} - (m+1)(1-\beta)^m\big) + m(1-\beta)^m \end{aligned} \tag{5.12}$$

items on average, using an average of

$$F = 1 + \big(1 - (1-\beta)^m\big)s$$

tests. It is now plausible – and can be shown formally [10] – that the average number of tests required by this binary splitting algorithm is

$$T_{\mathrm{av}} \sim \frac{F}{E} n. \tag{5.13}$$

The corresponding achievable rate is shown in Figure 5.2. Again, the 'bumps' come from changing integer values of the optimal choice of $s$.

A similar algorithm is studied in [10], following the work of Zaman and Pippenger [202], where we allow $m$ to be any integer, not just a power of 2, and use an optimal Huffman code to perform the binary splitting. Under such a Huffman code, a defective item will be found in either $\lfloor \log_2 m \rfloor$ or $\lceil \log_2 m \rceil$ tests. After optimising over all integers $m$, Zaman and Pippenger [202] show that this algorithm is optimal among a subset of adaptive algorithms called *nested* algorithms.

In [10], the above findings are combined with a concentration argument to show that one requires no more that $(1+\epsilon)T_{\mathrm{av}}$ with probability arbitrarily close to one for $n$ sufficiently large. Note from Figure 5.2 that we achieve rates of at least 0.95 for all $\beta \leq 1/2$.

Again, setting $m = 2^0 = 1$ recovers individual testing. Setting $m = 2^1 = 2$ recovers an algorithm of Fischer, Klasner and Wegenera [79], which, as they note, outperforms individual testing when $\beta < \beta^* = (3 - \sqrt{5})/2 \approx 0.382$. Note that $H((1-\beta)^2) > H(1-\beta)$ precisely when $\beta < \beta^*$, so intuitively $\beta < \beta^*$ is the regime where a test of two items is 'more informative' than a test of one item. Fischer, Klasner and Wegenera combine this observation with a converse result showing that this is best possible, since individual testing is optimal under the i.i.d. prior for $\beta \geq \beta^*$ [79].

## 5.6 Group testing with prior statistics

### 5.6.1 Prior defectivity model

In Section 1.1 and the Appendix to Chapter 1, we discussed the distinction between the combinatorial prior (every defective set of size $k$ is equally likely) and i.i.d. prior (every item is defective with the same probability $q$). Both of

these models have the common feature of exchangeability of items: We can swap the labels on any two items without affecting overall defectivity probabilities.

Although such a feature is attractive in some ways, it can also be unnecessarily restrictive. In particular, if certain items are *a priori* more likely to be defective than others, then we should ideally try to exploit this additional prior information. In this section, we describe some test designs and results along these lines. We focus on the setting described in the following definition, which to the best of our knowledge was first studied in the 1970s in papers such as [88, 107, 152].

**Definition 5.1** (Prior defectivity model)**.** Each item $i$ has a (known) prior probability $q_i \in [0,1]$ of being defective, and individual items are defective independently of one another.

For example, this could model a situation where different individuals have a particular level of immunity to infection according to their genetics and date of vaccination. In the following, we assume that all of the values of $q_i$ are known exactly, though we expect the resulting techniques to work well even when they are only known approximately.

Under the model in Definition 5.1, the key metrics of success will be expressed in terms of the overall entropy $H_n = \sum_{i=1}^{n} h(q_i)$ (where, as before, $h$ denotes the standard binary entropy) and the average number of defectives $\overline{k}_n = \sum_i q_i$. A similar information-theoretic argument to that used to prove the counting bound (*cf.*, Theorem 1.1) shows that at least $H_n$ tests are required to ensure that the success probability tends to one. As before, we will regard this counting bound as the benchmark for the performance of any algorithm.

Recall that for any defectivity model, we write $\mathbf{U} = (U_1, \dots, U_n)$ for a random vector which encodes the defectivity status of all the items, with $U_i$ being the indicator function of the event that item $i$ is defective, as in Definition 1.2. We make the following definition, which generalizes Definition 1.7 (since in the combinatorial case $H(\mathbf{U}) = \log_2 \binom{n}{k}$).

**Definition 5.2.** Given a random process generating the defectivity vector $\mathbf{U}$, and $T$ tests, we define the *rate* to be

$$\text{rate} := \frac{H(\mathbf{U})}{T}. \tag{5.14}$$

Of course, one may wish to create richer models of defectivity under which $\mathbf{U}$ is generated. For example, one could imagine individuals represented by the vertices of some graph, and the probability of defectivity of an item being affected by the defectivity status of each individual's graph neighbours, perhaps according to a Markov process. However, such a model is beyond the scope of our discussion here.

We briefly describe the contributions of two specific papers here. In each case, the key idea is that, by grouping items with a similar value of $q_i$ together, we can reduce the problem to one which behaves 'locally approximately' like the standard i.i.d. prior. The design of these algorithms can be understood in analogy with the design of lossless data compression algorithms. In this sense, one can design a splitting strategy based on binary trees that are balanced (with roughly equal probability on each branch to maximize the information gained from each test). These trees can be understood in analogy with Huffman codes,

which are known to achieve optimal lossless data compression (see for example [96]).

### 5.6.2 Adaptive testing

Kealy, Johnson and Piechocki [120] give a Hwang-type binary splitting algorithm (see Section 1.5) in the adaptive case, building on an earlier work of Li *et al.* [128] who developed the *Laminar algorithm* for the prior defectivity model (Definition 5.1). They discard very low probability items (which are unlikely to be defective anyway, so can be assumed nondefective without wasting tests). The remaining items are grouped together to form a collection of search sets $S_j$ that contain items with $\max_{l,m\in S_j} q_l/q_m \leq \Gamma$ (for some $\Gamma$), and with total probability $\sum_{i\in S_j} q_i \geq 1/2$ wherever possible. Then, one can perform binary splitting over each of these search sets $S_j$ one by one.

Using this approach, [120, Theorem 3.9] gives a technical condition on $q_i$ under which a rate of 1 (in the sense of Definition 5.2) is achievable in a regime where $\overline{k}_n/H_n \to 0$. In other words, roughly $H_n$ tests suffice to make the success probability tend to one. This algorithm can be viewed as the non-identical version of Hwang's generalized binary splitting algorithm [106], and this result is the non-identical version of Theorem 1.3.

### 5.6.3 Nonadaptive testing

In the nonadaptive setting, relatively less is known. As a baseline for the performance, we point out that if the number of defectives behaves as $\overline{k}_n(1+o(1))$ with probability approaching one (as we should usually expect for growing $\overline{k}_n$ due to concentration), then the sufficient number of tests proved for various algorithms in Chapter 2 (e.g., COMP and DD) remain valid. This is because the analysis of these algorithms was based on fixing a defective set of cardinality $k$ and bounding the probability with respect to the randomness in the test design. Hence, other than the slight modification of $k$ to $k(1+o(1))$ (which was discussed previously in Remark 2.3), the non-uniform prior does not impact the analysis. It should be noted, however, that using the same number of tests as the uniform setting does not amount to achieving the same rate; the rate can be much smaller for a given number of tests when $H(\mathbf{U}) \ll k\log_2 \frac{n}{k}$.

Nonadaptive test designs that introduce block structure into the test matrix were explored in [128]. The performance guarantee given for this approach does not quite amount to a positive rate, as the scaling achieved is $T = O(H(\mathbf{U})\log n)$ rather than $T = O(H(\mathbf{U}))$. We refer the interested reader to [128] for further details, and instead focus our attention on providing evidence towards designs that achieve a positive rate, or even a rate of one.

To do so, we consider a simplified setting in which the items are arranged into disjoint groups $G_1, \dots, G_m$ whose union equals $\{1, \dots, n\}$. Suppose that group $j$ contains $n_j$ items, each of which is defective with probability $\frac{\overline{k}_j}{n_j}$, with $\overline{k}_j$ denoting the average number of defectives in the group (in contrast with the above, the dependence on $n$ is left implicit). Motivated by the idea of using block designs [128], we can consider a simple approach in which we apply a standard group testing algorithm on each group of items separately.

Specifically, for group $j$, we fix a number of tests $T_j$ and form a $T_j \times n_j$

i.i.d. Bernoulli test matrix in which each entry is positive with probability $\frac{\nu}{\overline{k}_j}$, for some $\nu > 0$. We can study each such group using the techniques of the previous chapters, apply a union bound over the $m$ groups to deduce an overall upper bound on the error probability, and note that the total number of tests is $T = \sum_{j=1}^{m} T_j$.

For instance, putting aside computational considerations, suppose that we use the SSS decoding algorithm that achieves a rate of one (in the standard setting) when $k = O(n^{0.409})$ (*cf.*, Section 4.4). To simplify the analysis, we make the following assumptions:

- Both $n_j \to \infty$ and $\overline{k}_j \to \infty$ as $n \to \infty$, with $\overline{k}_j = O(n_j^{0.409})$. These assumptions readily yield that group $j$ contains $\overline{k}_j(1 + o(1))$ defectives with probability approaching one.
- The number of groups is bounded ($m = O(1)$), so if the error probability for each group vanishes, then so does the overall error probability.

In light of these observations and the fact that SSS achieves a rate of one when $k = O(n^{0.409})$ under the near-constant column weight design, one can achieve vanishing error probability with a number of tests satisfying

$$T = \sum_{j=1}^{m} \Big( \overline{k}_j \log_2 \frac{n_j}{\overline{k}_j} \Big) (1 + o(1)). \tag{5.15}$$

We claim that this in fact corresponds to a rate of one in the non-uniform prior defectivity model. To see this, note that

$$H(\mathbf{U}) = \sum_{j=1}^{m} n_j h\Big( \frac{\overline{k}_j}{n_j} \Big) \tag{5.16}$$

$$= \sum_{j=1}^{m} \Big( \overline{k}_j \log_2 \frac{n_j}{\overline{k}_j} \Big) (1 + o(1)), \tag{5.17}$$

where we have used $h(\alpha) = (-\alpha \log_2 \alpha)(1 + o(1))$ as $\alpha \to 0$.

It remains an interesting direction for future research to generalize the above approach to general values of $(q_1, \ldots, q_n)$ and understand what rates can be achieved, both information-theoretically and with practical decoding techniques.

## 5.7 Explicit constructions

Throughout the monograph, we have focused on randomized test designs, in particular Bernoulli designs and near-constant column weight designs. The sublinear-time algorithms described in Section 5.4 also use randomization in the design stage. The understanding of explicit deterministic constructions, by contrast, is only in its early stages when it comes to the small-error recovery criterion. In this section, we give an overview of some progress in this direction.

For the zero-error recovery criterion, several explicit constructions have been proposed. A prominent early example was provided by Kautz and Singleton [119], and achieves zero-error reconstruction in the noiseless setting with $T = O\big(k^2 \frac{\log^2 n}{\log^2 k}\big)$ tests. A related but more recent construction due to Porat and

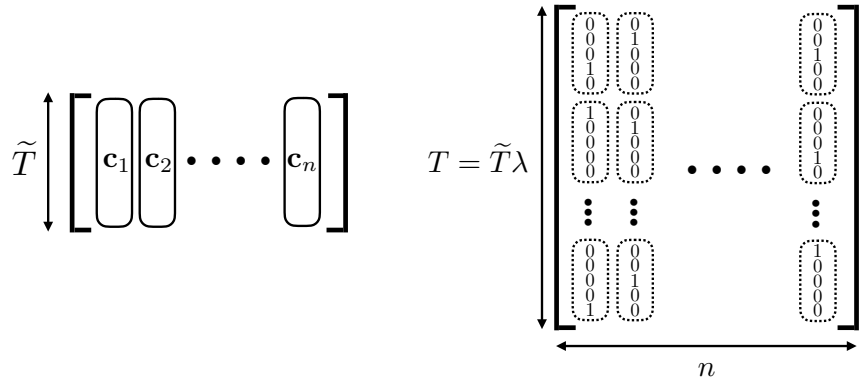



Figure 5.3: (Left) The codewords $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ of a length-$\widetilde{T}$ nonbinary code of size $n$ are arranged to form a $\widetilde{T} \times n$ nonbinary matrix, with each symbol taking one of $\lambda$ values. (Right) Each nonbinary symbol of the matrix on the left is replaced by a length-$\lambda$ binary string with a 1 in the entry indexing the value (out of $\lambda$ possibilities) of the corresponding nonbinary symbol.

Rothschild [155] achieves $T = O\big(k^2 \log n\big)$. Of course, these results are not sufficient for achieving a positive rate when $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$. The interested reader is referred to [155] for more detailed overview of deterministic constructions in the zero-error setting, and to [40] for an overview of alternative approaches based on derandomization.

Interestingly, recent developments on achieving a positive rate under the small-error criterion have made use of very similar constructions. Much like most works on the zero-error setting, these constructions are based on the idea of concatenated codes, depicted in Figure 5.3. The construction starts with a nonbinary channel code $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ containing $n$ codewords of length $\widetilde{T}$, with symbols taking one of $\lambda$ values. As shown in the left of the figure, these codewords are arranged in columns to form a $\widetilde{T} \times n$ matrix. To construct the final group testing matrix, each code-symbol is replaced by a length-$\lambda$ binary vector with a one in the entry indexing the corresponding nonbinary code-symbol and zeros elsewhere. This produces a constant column weight design, where each of the $n$ items is in exactly $\widetilde{T}$ of the $T = \widetilde{T}\lambda$ tests. If the original codewords are sufficiently well-separated, then different defective sets should lead to different test outcomes.

For zero-error group testing, the performance of this construction depends crucially on the minimum distance of the code $\mathcal{C}$. In contrast, Mazumdar [146] related the (nonzero) error probability of the construction to both the minimum distance and the average distance. This led to vanishing error probability with a number of tests of the form $T = O\big(k \frac{\log^2 n}{\log k}\big)$ using either of the following two nonbinary codes: (i) the above-mentioned construction of Porat and Rothschild [155], which achieves the Gilbert-Varshamov bound; (ii) a construction based on algebraic-geometric codes due to Tsfasman *et al.* [188]. The behaviour $T = O\big(k \frac{\log^2 n}{\log k}\big)$ leads to a positive rate whenever $k = \Theta(n^\alpha)$ for some $\alpha \in (0, 1)$, although the rate vanishes as $\alpha \to 0$.

The original construction of Kautz and Singleton [119] used a Reed–Solomon code with $\widetilde{T} = \lambda - 1$ and $n = \lambda^k$. In a recent work, Inan *et al.* [108] studied a similar construction with Reed-Solomon codes, but proposed different parameters, choosing $\widetilde{T} = \Theta(\log n)$ while still using $n = \lambda^k$. They performed a direct analysis without relying on distance properties, and showed that one can achieve vanishing error probability with $T = O(k \log n)$ tests as long as $k = \Omega(\log^2 n)$. As a result, a positive rate is achieved even in the limit $\alpha \to 0$.

Both [146] and [108] obtained the preceding results using the COMP algorithm (under a different name). In addition, in [108] the same construction was combined with the NCOMP algorithm (see Section 3.4) to obtain the same $T = O(k \log n)$ scaling under the symmetric noise model.

While these results provide very important contributions in understanding explicit constructions under the small-error criterion, their focus is on the scaling laws rather than the constant factors or the rates achieved. It remains an important open challenge to develop achievable rates for explicit constructions that can compete with those of randomized constructions, or better yet, approach the algorithm-independent converse bounds.

The preceding results are summarized in Table 5.5.

| Reference | #Tests | Zero-error? |
|---|---|---|
| Kautz-Singleton [119] | $O\big(k^2 \frac{\log^2 n}{\log^2 k}\big)$ | yes |
| Porat-Rothschild [155] | $O(k^2 \log n)$ | yes |
| Mazumdar [146] | $O\big(k \frac{\log^2 n}{\log k}\big)$ | no |
| Inan *et al.* [108] | $O(k \log n)$ | no |

Table 5.5: Summary of the number of tests required for explicit group testing designs, in the zero-error and small-error settings. The result in [108] additionally assumes that $k = \Omega(\log^2 n)$.

## 5.8 Constrained group testing

Thus far, we have assumed that any given group test can contain an arbitrary set of items. However, in several practical applications, the tests are in fact subject to certain constraints that may make standard designs (e.g., i.i.d. Bernoulli) infeasible. In this section, we give an overview of several such constraints and how they are motivated by particular applications of interest.

### 5.8.1 Path constraints on a graph

The work of Cheraghchi *et al.* [42] considers a model in which items may correspond to either nodes or edges in a graph (but not both), and the only tests allowed are those that correspond to a path on that graph. One of the main motivating applications for this scenario is the problem of network tomography (mentioned previously in Section 1.7), in which the nodes correspond to machines and the edges correspond to connections between machines. The goal is to identify faults in the network, and this is done by checking whether a packet

sent from one machine successfully arrived at a target machine. There are at least two interesting variants of this setup:

- If the goal is to find faulty connections between machines, then one can consider the edges as corresponding to items, and each test as corresponding to sending a packet between two machines along a specific path and checking whether it arrived successfully. Hence, each test is constrained to be the set of edges along some path in the graph.
- If the goal is to find faulty machines, then one can consider the nodes as corresponding to items, and a test again corresponds to sending a packet between two machines along a specific path and checking whether it arrived successfully. Hence, each test is constrained to be the set of nodes along some path in the graph.

The second of these generalizes an earlier model corresponding to the special case of a line graph [44].

For both of the above models, [42] propose random testing designs constructed by performing a *random walk* on the graph, thus ensuring that the path constraint is not violated. It was shown the $O(k^2 \log n)$ achievability result for zero-error group testing can be generalized to $O(k^2\tau(n)^2 \log n)$, where $\tau(n) \geq 1$ is a property of the graph known as the mixing time, defined formally in this context in [42, Definition 6]. For many graphs, $\tau(n)$ is small (e.g., constant or logarithmic in $n$). For example, [42, Section V] discusses the fact that for expander graphs with a constant spectral gap, and for Erdős-Rényi random graphs, $\tau(n) = O(\log n)$ (with high probability). This means that for graphs of this kind, the additional constraints do not considerably increase the required number of tests in the zero-error setting.

To our knowledge, these constraints have not been studied in the small-error setting, which is the main focus of this monograph. This poses a potentially interesting direction for future research.

### 5.8.2 Edge constraints on a graph

A series of works rooted in early studies in theoretical computer science (e.g., see [3, 4]) have considered a different type of constraint imposed by a graph $G = (V, E)$. Here, the edges correspond to items, and $k$ of the $|E|$ items are defective. However, each test is a group of *nodes* rather than edges, and the test outcome is positive if and only if there is at least one defective edge connecting two different nodes in the group. An interesting special case of this problem is obtained when $E$ is the complete graph, and the goal is to identify a sub-graph of $k$ edges corresponding to those that are defective.

As an example application, this might correspond to a scenario where we wish to identify interactions between chemicals, and each test amounts to combining a number of chemicals and observing whether any reaction occurs.

The work of [111] considers the adaptive setting, and shows that even in the general formulation, it suffices to have $k \log_2 \frac{|E|}{k} + O(k)$ tests, which matches (a slight variation of) the counting bound asymptotically whenever $k = o(|E|)$. More recently, significant effort has been put into developing algorithms with limited stages of adaptivity. We refer the reader to [1] and the references therein for further details, and highlight a particularly interesting result: In the special

case that $E$ is the complete graph, it is known that when $k$ grows sufficiently slowly compared to $|V|$,[2] any nonadaptive algorithm requires $\Omega(k^2 \log |V|)$ tests even when a small probability of error is tolerated. This is in stark contrast with standard group testing, where $k^2$ terms only arise under the zero-error criterion (see Section 1.6). On the other hand, it was recently shown that $O(k \log |V|)$ nonadaptive tests indeed suffice in the small-error setting for certain *random* (rather than worst-case) graphs with an average of $k$ edges [130].

### 5.8.3 Sparse designs

Another interesting form of constrained group testing is that in which the number of tests-per-item or items-per-test is limited. Using the terminology of Gandikota *et al.* [85], we consider the following:

- The constraint of $\gamma$-*divisibility* requires that each item participates in at most $\gamma$ tests. If one considers the classical application of testing blood for diseases, this corresponds to the case that each patient's blood sample can only be split into a limited number of smaller subsamples.
- The constraint of $\rho$-*sized tests* requires that any given test contains at most $\rho$ items. This may correspond to equipment limitations that prevent arbitrarily many items from being included in a pool.

As noted in [85], the interesting cases are the regimes $\gamma = o(\log n)$ and $\rho = o\big(\frac{n}{k}\big)$; this is because if one allows $\gamma = O(\log n)$ or $\rho = O\big(\frac{n}{k}\big)$ and the implied constants are not too small, then even standard designs (such as constant column weight designs) can be used that are already near-optimal in the unconstrained sense.

Under the $\gamma$-divisibility constraint with $\gamma = o(\log n)$, we have the following results:

- A simple adaptive algorithm attains zero error probability using $\gamma k(n/k)^{1/\gamma}$ tests (this was also noted in an early work of Li [127]). The algorithm is a '$\gamma$-stage' algorithm in the sense of Section 5.2, and is defined recursively. The case $\gamma = 1$ uses individual testing. Then the algorithm for $\gamma$-divisibility is as follows: Split the $n$ items into $A = k(n/k)^{1/\gamma}$ sets of size $n/A$, and test each set. As usual, all items in negative tests are nondefective. For the remaining items – of which there are at most $kn/A$ – continue with the $(\gamma - 1)$-divisibility algorithm. The first step takes $k(n/k)^{1/\gamma}$ tests, and the worst-case number of tests in each subsequent step is easily checked to also be $k(n/k)^{1/\gamma}$, giving the desired result. The case $\gamma = 2$ gives Dorfman's original adaptive algorithm [61] (see equation (1.13) in Chapter 1).
- For nonadaptive testing and the small-error criterion, Gandikota *et al.* [85] show that any algorithm requires roughly $\gamma k(n/k)^{1/\gamma}$ tests. In addition, they show that one can attain error probability at most $\epsilon$ using at most $e\gamma k(n/\epsilon)^{1/\gamma}$ tests, which behaves similarly to the converse bound but nevertheless leaves a gap in the scaling laws. The corresponding algorithm and test design are discussed below.

[2] This is a more stringent requirement than it may seem, since there are $\binom{|V|}{2} \approx \frac{1}{2}|V|^2$ items in total.

Under the $\rho$-sized tests constraint with $\rho = \Theta\big((n/k)^\beta\big)$ for some $\beta \in [0,1)$ and in the sparse regime $k = \Theta(n^\alpha)$, $\alpha < 1$, we have the following results:

- Any testing requires at least $n/\rho$ tests, as this many tests is required to test each item once. Applying the generalized binary splitting algorithm (see Sections 1.5 and 5.5) starting with sets of size $\rho$ requires $\frac{n}{\rho} + k\log_2\rho + O(k) \sim \frac{n}{\rho}$ tests, where $n/\rho$ is for testing each set once and $k\log_2\rho + O(k)$ is for $k$ rounds of binary splitting.
- For nonadaptive small-error testing, Gandikota *et al.* [85] show that any algorithm requires roughly $\frac{1}{1-\beta}\frac{n}{\rho}$ tests. In addition, they show that vanishing error probability can be attained using roughly $\frac{1}{(1-\beta)(1-\alpha)}\frac{n}{\rho}$ tests.

We observe that under both types of constraint, the number of tests required can be far higher than the usual $O(k\log n)$ scaling.

The lower bounds of Gandikota *et al.* [85] are based on Fano's inequality, but with more careful entropy bounds than the standard approach of upper bounding the entropy of a test outcome by 1. For instance, if each test contains at most $\rho = o(n/k)$ items, then the probability of the test being positive tends to zero, so the entropy per test is much smaller than one bit. The upper bounds of [85] are based on test designs with exactly $\gamma$ tests per item or exactly $\rho$ items per test, along with the use of the basic COMP algorithm. Analysing more sophisticated algorithms such as DD may lead to further improvements.

We briefly mention that a complementary constrained group testing problem has also been considered, in which the number of *positive tests* (rather than items-per-test or tests-per-item) is constrained [53].

## 5.9 Other group testing models

Throughout the monograph, we have focused primarily on the noiseless model and certain simple random noise models such as symmetric noise, addition noise, and dilution noise. There are extensive additional models that have been considered previously in the literature, but in most cases, understanding them via the information-theoretic viewpoint remains open. We provide an incomplete list of examples as follows:

- The *linear model* (also known as the *quantitative group testing model* or the *adder channel model*) outputs the number of defectives in the test, thus providing much more information than the presence or absence of at least one defective. In fact, this reduces the required number of tests from $O\big(k\log\frac{n}{k}\big)$ to $O\big(k\frac{\log\frac{n}{k}}{\log k}\big)$ [170, 166, 90, 27]; in the sparse regime $k = \Theta(n^\alpha)$ for $\alpha < 1$, this is an improvement from $O(k\log n)$ to $O(k)$. This setting can be viewed as a very specific case of compressive sensing (see, for example, [80]) in which both the measurement matrix and unknown signal are binary-valued.
- The *semi-quantitative model* [68] lies in between the two extremes of the standard model and the linear model. The model has a number of thresholds, and we get to observe the largest of those thresholds that the number of defective items in the test $|\{i \in \mathcal{K} : x_i = 1\}|$ is greater than or equal to.

- Various forms of *threshold group testing* output a 0 if there are too few defectives, output a 1 if there are sufficiently many defectives, and exhibit either random [125] or adversarial [52, 41] behaviour in between the two corresponding thresholds. We presented a simple randomized version of this model in Example 3.6.

- Other models have been considered with more than two types of items, with a prominent example being *group testing with inhibitors* introduced by Farach *et al.* [75] and discussed in [63, Chapter 8]. In the simplest case, each item is either defective, an inhibitor, or neither of the two, and the test is positive if and only if it contains at least one defective but no inhibitors. In other words, inhibitors may 'hide' the fact that the test contains one or more defective items. Some results concerning this model are provided, for example, in [58, 87].

- Different items may have different 'defectivity levels' represented by a non-negative real number, and one may either observe the sum of defectivity levels of the tested items (linear case), or only the highest defectivity level (nonlinear case) [62, Ch. 6]. If all the defectivity levels are zero or one, then we recover linear group testing (linear case) or standard group testing (nonlinear case).

While some of these models can be studied under the information-theoretic framework considered in Chapter 4, characterizing the number of tests still requires the non-trivial step of bounding mutual information terms, for example, as in Theorem 4.5. To our knowledge, this has only been done explicitly for the linear model in the more general context of pooled data problems with multiple types of items (that is, possibly more than two) [166], and in certain random models for threshold group testing [125].

On the other hand, several upper and lower bounds on the number of tests required in the above models have indeed been developed, often using rather different approaches compared to the standard setting. Hence, it is of significant interest to further study what the information theory perspective can provide for these models, potentially building on the concepts and techniques surveyed in this monograph.

# Chapter 6

# Conclusions and Open Problems

We have surveyed recent theoretical and algorithmic developments in group testing, with an emphasis on achievable rates under nonadaptive testing in the small-error regime.

In the noiseless setting (Chapter 2), we presented the achievable rates of COMP and DD, and showed the SCOMP and linear programming (LP) algorithms to perform better experimentally while achieving rates at least as high as DD (which in turn exceed those of COMP). In the noisy setting (Chapter 3), we presented noisy variants of COMP, DD, and LP, as well as two additional algorithms: separate decoding of items, which is convenient to analyze theoretically; and belief propagation, which performs very well experimentally but currently lacks a theoretical analysis.

The information-theoretic results presented in Chapter 4 establish certain regimes where the practical algorithms are optimal or near-optimal – notably, this includes the DD algorithm with a Bernoulli or near-contest column weight design in the sparse regime $k = \Theta(n^\alpha)$ with $\alpha \in \left(\frac{1}{2}, 1\right)$. There are also regimes where there remains a significant gap between the algorithmic rates and the information-theoretic limits, e.g., Bernoulli designs with $\alpha < \frac{1}{3}$. In the noiseless case, both the information-theoretic limits and the algorithmic rates were seen to improve by moving from the Bernoulli design to the near-constant column weight design.

In Chapter 5, we surveyed a wide range of important variants of the standard group testing problem, including partial recovery, multi-stage adaptive algorithms, counting defectives, sublinear-time decoding, linear sparsity, non-uniform prior statistics, explicit constructions, constrained test designs, and other group testing models.

We conclude the monograph with a list of some prominent open problems relating to the material that we surveyed.

**Open Problem 1.** *What is the capacity of nonadaptive group testing in the sparse regime, where $k = \Theta(n^\alpha)$ with $\alpha \in (0, 1)$?*

We know from Theorem 4.2 that the capacity is 1 for $\alpha \leq \frac{\ln 2}{1+\ln 2} \approx 0.409$, but all values of $\alpha > 0.409$ remain open. The best known achievability result is

$(\ln 2)\frac{1-\alpha}{\alpha}$, for the near-constant column weight design (Theorem 4.2), while the counting bound gives an upper bound of 1. See Figure 1.4 for an illustration.

By Theorem 4.4, improving the achievable rate will require a different design, and not merely a better decoding algorithm or proof strategy. Alternatively, it may be that Theorem 4.2 already gives the best possible achievability result, in which case a matching converse result would be needed. While converse results already exist for particular designs, the challenge here would be to prove a converse which is universal across all designs and decoding algorithms.

**Open Problem 2.** *What more can be said about finite-size group testing problems using information-theoretic methods?*

Results regarding the rate of group testing indicate how the number $T = T(n)$ of tests required behaves as $n \to \infty$. However, in practice, we might be interested in a fairly modest number of items, perhaps of the order 100 to $10,000$. In such cases, results concerning the asymptotic rate may be of limited value, particularly if $\log_2 \binom{n}{k}/T$ only converges slowly to the maximum achievable rate. What then can we say about the number of tests required with such 'finite $n$'? Information-theoretic approaches to 'finite blocklength' results in channel coding were pioneered by Polyanskiy, Poor and Verdú [154]. Can similar methods provide additional insight in the context of group testing?

One existing work in this direction is [112], which built on the ideas of [154] to develop converse results that generalize Theorem 1.1. However, in stark contrast with channel coding, we are not aware of any works attempting a finite-size *achievability* analysis of group testing.

**Open Problem 3.** *Find a practical decoding algorithm (and a non-adaptive test design) that achieves a rate higher than* $\ln 2 \approx 0.693$*, or prove that no such algorithm exists.*

The DD algorithm was discussed in Sections 2.4 and 2.7 for the Bernoulli design and near-constant column weight design, respectively. The rates were seen to be optimal with respect to these designs for a high sparsity parameter $\alpha \in [\frac{1}{2}, 1)$, while saturating to $\frac{1}{\mathrm{e}\ln 2} \approx 0.531$ or $\ln 2 \approx 0.693$ for lower values of $\alpha < \frac{1}{2}$ (see Figure 2.1). Interestingly, separate decoding of items items (Section 3.5) with a Bernoulli design also achieves a rate approaching $\ln 2 \approx 0.693$ as $\alpha \to 0$. However, no practical algorithm is known to achieve a rate exceeding $\ln 2$ for any value of $\alpha$.

While the more sophisticated SCOMP algorithm outperforms DD in practice, its rate with a near-constant column weight design is the same as DD (as discussed in Section 2.7). Approaches based on linear programming (Section 2.6) or belief propagation (Section 3.3) could be candidates for algorithms that not only work better in practice, but achieve a strictly higher rate.

Alternatively, one could seek to establish a negative result based on the theory of computation establishing that the apparent 'barrier' at the rate $\ln 2$ cannot be improved with any polynomial-time algorithm. In the related inference problem of linear regression (e.g., see [156]), problems may be divided into three categories: According to the amount and quality of data, problems are 'easy' (practically feasible in our terminology; comparable in complexity with DD or COMP), 'hard' (solvable by brute force methods, but not efficiently; comparable with SSS) and 'impossible'. In this sense, the key challenge in noiseless nonadaptive group testing is to determine whether rates in the range $(\ln 2, 1]$ are 'easy' or 'hard'.

**Open Problem 4.** *Establish improved achievable rates and converse results for noisy group testing.*

In Chapers 3 and 4, we studied various achievable rates and converse bounds for noisy group testing, but left many questions unanswered. For instance, while the achievable rate of Theorem 4.6 for symmetric noise is tight in the sparse regime for very small $\alpha$, for larger $\alpha$ the use of Bernstein's inequality appears to be overly crude. In addition, it is generally unclear whether the rates known for Noisy DD (Section 3.6) are the best possible, even with respect to the Bernoulli design. Of course, it may be that closing the gaps in the bounds requires improvements in both the achievability and converse parts.

Beyond tightening the analysis of the Bernoulli design, it is also natural to further consider the near-constant column weight design (see Sections 2.7 and 4.4), which is known to provide improved rates in the noiseless setting, but is yet to be explored in the noisy setting. We expect that the analysis of NCOMP (Section 3.4) under this design would be relatively straightforward. However, the research directions having a greater potential for tight results appear to be much more challenging. Can one adapt the existing analysis of DD with near-constant tests-per-item (Section 2.7) to the noisy setting? Can one adapt the tight achievability analysis for near-constant column weight designs (Section 4.4) to the noisy setting?

**Open Problem 5.** *What are the fundamental limits of noiseless group testing in the linear regime with partial recovery and/or constant error probability?*

In Section 5.5, we discussed the linear regime $k \sim \beta n$, and showed that in the nonadaptive setting with exact recovery and the small-error criterion (i.e., asymptotically vanishing error probability), it is optimal to test every item individually. While this is a seemingly very negative result, it is not the end of the story – it remains plausible that considerably fewer tests may be required if either (i) one does not insist on exact recovery, but instead allows some false negatives and/or false positives in the reconstruction, or (ii) one does not insist on the error probability approaching zero, but instead tolerates it converging to some fixed constant in $(0, 1)$

The fact that partial recovery is possible with vanishing error probability when $T \sim k \log_2 \frac{n}{k}$ in the sublinear regime for all $\alpha \in (0, 1)$ (see Section 5.1) suggests that it should remain possible at least as $\beta \to 0$ in the linear regime. In the regime $k \leq \Theta(n^{0.409})$ with exact recovery, the *strong converse* (e.g., see Remark 1.3) indicates that any target error probability $\epsilon \in (0, 1)$ yields the same asymptotic number of tests, corresponding to the counting bound. However, the picture in the linear regime is potentially very different, and this direction is yet to be explored.

**Open Problem 6.** *Find a non-adaptive group testing algorithm that succeeds with $O(k \log n)$ tests and has $O(k \log n)$ decoding time.*

As discussed in Section 5.4, the non-adaptive GROTESQUE and SAFFRON algorithms achieve $O(k \log k \log n)$ for both the number of tests and decoding time, falling short of the desired $O(k \log n)$ scaling by a logarithmic factor. On the other hand, bit-mixing coding achieves the optimal scaling $O(k \log n)$ for the number of tests, but has decoding time $O(k^2 \log k \log n)$, which is suboptimal by at least a factor of $k$. Ideally, one would bring both the number of tests and the decoding time down to $O(k \log n)$, with both being optimal. As a more

modest goal, a combination of $O(k \log n)$ tests and $O(k(\log n)^c)$ decoding time for some $c > 1$ would in itself be a significant improvement.

It is worth noting that when it comes to *adaptive* algorithms, the corresponding question was already resolved using a variant of the GROTESQUE algorithm; see Section 5.4.2.

**Open Problem 7.** *Establish precise achievable rates for group testing algorithms (a) with sublinear decoding time, and/or (b) with an explicit construction of the test matrix.*

The results attaining $T = O(k \log n)$ for sublinear-time algorithms (see Section 5.4) and explicit constructions (see Section 5.7) were focused on scaling laws and not constant factors, and it remains an open problem to prove explicit achievable rates competitive with those of COMP, DD, and so on. A natural step towards this goal is to modify the analyses of the existing works with an explicit view towards attaining the best possible constants. Alternatively, it could be the case that to truly compete with the best known rates, new algorithmic ideas are needed.

**Open Problem 8.** *Prove the Hu–Hwang–Wang conjecture, that individual testing is optimal for adaptive combinatorial zero-error group testing when $k \geq n/3$.*

Adaptive testing in the linear regime where $k \sim \beta n$ was discussed in Section 5.5.2. In particular, for zero-error testing (with the combinatorial prior), we saw that one could improve on testing each item individually when $\beta < \frac{1}{3}$. Hu, Hwang and Wang [104] conjecture that this result is tight, in the sense that one requires $T \geq n - 1$ when $\beta \geq \frac{1}{3}$. (Recall that, since $k$ is known, we need not test the final item, hence we need only $n - 1$ tests for 'individual testing'.) The best known result is that individual testing is optimal for $\beta \geq 1 - \log_3 2 \approx 0.369$ [157].

**Open Problem 9.** *What does the information theory perspective have to offer other non-standard group testing models, and more general structured signal recovery problems?*

This question is intentionally open-ended, and we mention at the outset that information theory has indeed already played a major role in extensive problems concerning structured signal recovery and high-dimensional statistics. Nevertheless, non-standard group testing models such as those covered in Section 5.9, including quantitative group testing, threshold group testing, and more, remain less well-understood than the standard 'OR' model and its noisy counterparts. There is potentially significant potential for the further development of fundamental performance limits and algorithms from the information theory perspective, and we expect that the techniques surveyed in this monograph could play a significant role in doing so.

# Bibliography


[1] H. Abasi and N. H. Bshouty. On learning graphs with edge-detecting queries, 2018, arXiv: 1803.10639.

[2] A. Agarwal, S. Jaggi, and A. Mazumdar. Novel impossibility results for group-testing, 2018, arXiv: 1801.02701.

[3] M. Aigner. Search problems on graphs. *Discrete Applied Mathematics*, 14(3):215–230, 1986. DOI: 10.1016/0166-218X(86)90026-0.

[4] M. Aigner and E. Triesch. Searching for an edge in a graph. *Journal of Graph Theory*, 12(1):45–57, 1988. DOI: 10.1002/jgt.3190120106.

[5] C. Aksoylar, G. K. Atia, and V. Saligrama. Sparse signal processing with linear and nonlinear observations: A unified shannon-theoretic approach. *IEEE Transactions on Information Theory*, 63(2):749–776, 2017. DOI: 10.1109/TIT.2016.2605122.

[6] M. Aldridge. *Interference mitigation in large random wireless networks*. PhD thesis, University of Bristol, 2011, arXiv: 1109.1255.

[7] M. Aldridge. The capacity of bernoulli nonadaptive group testing. *IEEE Transactions on Information Theory*, 63(11):7142–4148, 2017. DOI: 10.1109/TIT.2017.2748564.

[8] M. Aldridge. On the optimality of some group testing algorithms. In *IEEE International Symposium on Information Theory (ISIT)*, pages 3085–3089, 2017. DOI: 10.1109/ISIT.2017.8007097.

[9] M. Aldridge. Individual testing is optimal for nonadaptive group testing in the linear regime. *IEEE Transactions on Information Theory*, 65(4):2058–2061, 2019. DOI: 10.1109/TIT.2018.2873136.

[10] M. Aldridge. Rates for adaptive group testing in the linear regime, 2019, arXiv: 1901.09687.

[11] M. Aldridge, L. Baldassini, and K. Gunderson. Almost separable matrices. *Journal of Combinatorial Optimization*, 33(1):215–236, 2017. DOI: 10.1007/s10878-015-9951-1.

[12] M. Aldridge, L. Baldassini, and O. T. Johnson. Group testing algorithms: Bounds and simulations. *IEEE Transactions on Information Theory*, 60(6):3671–3687, 2014. DOI: 10.1109/TIT.2014.2314472.

[13] A. Allemann. An efficient algorithm for combinatorial group testing. In H. Aydinian, F. Cicalese, and C. Deppe, editors, *Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede*, pages 569–596. Springer, 2013. DOI: 10.1007/978-3-642-36899-8_29.

[14] A. Ambainis, A. Belovs, O. Regev, and R. De Wolf. Efficient quantum algorithms for (gapped) group testing and junta testing. In *Proceedings of the 27th Annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 903–922, 2016. DOI: 10.1137/1.9781611974331.ch65.

[15] R. B. Ash. *Information Theory*. Dover Publications Inc., New York, 1990.

[16] G. K. Atia, S. Aeron, E. Ermis, and V. Saligrama. On throughput maximization and interference avoidance in cognitive radios. In *5th IEEE Consumer Communications and Networking Conference (CCNC)*, pages 963–967, 2008. DOI: 10.1109/ccnc08.2007.222.

[17] G. K. Atia and V. Saligrama. Boolean compressed sensing and noisy group testing. *IEEE Transactions on Information Theory*, 58(3):1880–1901, 2012. DOI: 10.1109/TIT.2011.2178156. See also [18].

[18] G. K. Atia, V. Saligrama, and C. Aksoylar. Correction to 'Boolean compressed sensing and noisy group testing'. *IEEE Transactions on Information Theory*, 61(3):1507–1507, 2015. DOI: 10.1109/TIT.2015.2392116.

[19] A. Atıcı and R. A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007. DOI: 10.1007/s11128-007-0061-6.

[20] L. Baldassini, O. T. Johnson, and M. Aldridge. The capacity of adaptive group testing. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2676–2680, 2013. DOI: 10.1109/ISIT.2013.6620712.

[21] D. J. Balding, W. J. Bruno, D. C. Torney, and E. Knill. A comparative survey of non-adaptive pooling designs. In T. Speed and M. S. Waterman, editors, *Genetic Mapping and DNA Sequencing*, pages 133–154. Springer, 1996. DOI: 10.1007/978-1-4612-0751-1_8.

[22] T. Berger and V. I. Levenshtein. Asymptotic efficiency of two-stage disjunctive testing. *IEEE Transactions on Information Theory*, 48(7):1741–1749, 2002. DOI: 10.1109/TIT.2002.1013122.

[23] T. Berger, N. Mehravari, D. Towsley, and J. Wolf. Random multiple-access communication and group testing. *IEEE Transactions on Communications*, 32(7):769–779, 1984. DOI: 10.1109/TCOM.1984.1096146.

[24] E. Blais. Testing juntas: A brief survey. In O. Goldreich, editor, *Property Testing: Current research and surveys*, pages 32–40. Springer, 2010. DOI: 10.1007/978-3-642-16367-8_4.

[25] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970. DOI: 10.1145/362686.362692.

[26] S. Bondorf, B. Chen, J. Scarlett, H. Yu, and Y. Zhao. Sublinear-time non-adaptive group testing with $O(k \log n)$ tests via bit-mixing coding, 2019, arXiv: 1904.10102.

[27] N. H. Bshouty. Optimal algorithms for the coin weighing problem with a spring scale. In *Conference on Learning Theory*, 2009. URL: https://www.cs.mcgill.ca/~colt2009/papers/004.pdf.

[28] N. H. Bshouty. Lower bound for non-adaptive estimate the number of defective items. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2018. URL: https://eccc.weizmann.ac.il/report/2018/053/. TR18-053.

[29] N. H. Bshouty, V. E. Bshouty-Hurani, G. Haddad, T. Hashem, F. Khoury, and O. Sharafy. Adaptive group testing algorithms to estimate the number of defectives. In *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 93–110, 2018. URL: http://proceedings.mlr.press/v83/bshouty18a.html.

[30] N. H. Bshouty and A. Costa. Exact learning of juntas from membership queries. *Theoretical Computer Science*, 742:82–97, 2018. DOI: 10.1016/j.tcs.2017.12.032. Algorithmic Learning Theory.

[31] P. Busschbach. Constructive methods to solve problems of $s$-surjectivity, conflict resolution, coding in defective memories, 1984. Rapport Interne ENST 84 D005.

[32] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi. Efficient algorithms for noisy group testing. *IEEE Transactions on Information Theory*, 63(4):2113–2136, 2017. DOI: 10.1109/TIT.2017.2659619.

[33] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In *49th Annual Allerton Conference on Communication, Control, and Computing*, pages 1832–1839, 2011. DOI: 10.1109/ALLERTON.2011.6120391.

[34] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri. Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Transactions on Information Theory*, 60(5):3019–3035, 2014. DOI: 10.1109/TIT.2014.2310477.

[35] C. L. Chen and W. H. Swallow. Using group testing to estimate a proportion, and to test the binomial model. *Biometrics*, 46(4):1035–1046, 1990. DOI: 10.2307/2532446.

[36] H.-B. Chen and F. K. Hwang. Exploring the missing link among $d$-separable, $\bar{d}$-separable and $d$-disjunct matrices. *Discrete Applied Mathematics*, 155(5):662–664, 2007. DOI: 10.1016/j.dam.2006.10.009.

[37] H.-B. Chen and F. K. Hwang. A survey on nonadaptive group testing algorithms through the angle of decoding. *Journal of Combinatorial Optimization*, 15(1):49–59, 2008. DOI: 10.1007/s10878-007-9083-3.

[38] Y. Cheng. An efficient randomized group testing procedure to determine the number of defectives. *Operations Research Letters*, 39(5):352–354, 2011. DOI: 10.1016/j.orl.2011.07.001.

[39] M. Cheraghchi. Noise-resilient group testing: Limitations and constructions. In *International Symposium on Fundamentals of Computation Theory*, pages 62–73, 2009. DOI: 10.1007/978-3-642-03409-1_7.

[40] M. Cheraghchi. Derandomization and group testing. In *48th Allerton Conference on Communication, Control, and Computing*, pages 991–997, 2010. DOI: 10.1109/ALLERTON.2010.5707017.

[41] M. Cheraghchi. Improved constructions for non-adaptive threshold group testing. *Algorithmica*, 67(3):384–417, 2013. DOI: 10.1007/s00453-013-9754-7.

[42] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama. Graph-constrained group testing. *IEEE Transactions on Information Theory*, 58(1):248–262, 2012. DOI: 10.1109/TIT.2011.2169535.

[43] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. DOI: 10.1287/moor.4.3.233.

[44] F. Cicalese, P. Damaschke, and U. Vaccaro. Optimal group testing algorithms with interval queries and their application to splice site detection. *International Journal of Bioinformatics Research and Applications*, 1(4):363–388, 2005. DOI: 10.1504/IJBRA.2005.008441.

[45] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild. Pattern matching with don't cares and few errors. *Journal of Computer and System Sciences*, 76(2):115–124, 2010. DOI: 10.1016/j.jcss.2009.06.002.

[46] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick. Information-theoretic and algorithmic thresholds for group testing, 2019, arXiv: 1902.02202.

[47] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition edition, 2009.

[48] G. Cormode and S. Muthukrishnan. What's hot and what's not: Tracking most frequent items dynamically. *ACM Transactions on Database Systems (TODS)*, 30(1):249–278, 2005. DOI: 10.1145/1061318.1061325.

[49] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition edition, 2006. DOI: 10.1002/047174882X.

[50] I. Csiszár and J. Körner. *Information Theory: Coding theorems for discrete memoryless systems*. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511921889. 2nd edition.

[51] R. N. Curnow and A. P. Morris. Pooling DNA in the identification of parents. *Heredity*, 80(1):101–109, 1998. DOI: 10.1038/sj.hdy.6882420.

[52] P. Damaschke. Threshold group testing. In *General Theory of Information Transfer and Combinatorics*, pages 707–718. Springer, 2006. DOI: 10.1007/11889342_45.

[53] P. Damaschke. Adaptive group testing with a constrained number of positive responses improved. *Discrete Applied Mathematics*, 205:208–212, 2016. DOI: 10.1016/j.dam.2016.01.010.

[54] P. Damaschke and A. S. Muhammad. Bounds for nonadaptive group tests to estimate the amount of defectives. In *Combinatorial Optimization and Applications (COCOA)*, pages 117–130, 2010. DOI: 10.1007/978-3-642-17461-2_10.

[55] P. Damaschke and A. S. Muhammad. Competitive group testing and learning hidden vertex covers with minimum adaptivity. *Discrete Mathematics, Algorithms and Applications*, 2(03):291–311, 2010. DOI: 10.1142/S179383091000067X.

[56] P. Damaschke and A. S. Muhammad. Randomized group testing both query-optimal and minimal adaptive. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 214–225, 2012. DOI: 10.1007/978-3-642-27660-6_18.

[57] A. De Bonis, L. Gasieniec, and U. Vaccaro. Optimal two-stage algorithms for group testing problems. *SIAM Journal on Computing*, 34(5):1253–1270, 2005. DOI: 10.1137/S0097539703428002.

[58] A. De Bonis and U. Vaccaro. Improved algorithms for group testing with inhibitors. *Information Processing Letters*, 67(2):57–64, 1998. DOI: 10.1016/S0020-0190(98)00088-X.

[59] A. De Bonis and U. Vaccaro. Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels. *Theoretical Computer Science*, 306(1-3):223–243, 2003. DOI: 10.1016/S0304-3975(03)00281-0.

[60] A. De Bonis and U. Vaccaro. $\epsilon$-almost selectors and their applications to multiple-access communication. *IEEE Transactions on Information Theory*, 63(11):7304–7319, 2017. DOI: 10.1109/TIT.2017.2750178.

[61] R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943. DOI: 10.1214/aoms/1177731363.

[62] D.-Z. Du and F. Hwang. *Combinatorial Group Testing and Its Applications*. World Scientific, 2nd edition edition, 1999. DOI: 10.1142/4252.

[63] D.-Z. Du and F. K. Hwang. *Pooling Designs and Nonadaptive Group Testing: Important tools for DNA sequencing*. World Scientific, 2006. DOI: 10.1142/6122.

[64] A. G. D'yachkov. Lectures on designing screening experiments, 2004, arXiv: 1401.7505. Lecture Note Series 10, POSTECH.

[65] A. G. D'yachkov and V. V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13, 1982. URL: https://www.researchgate.net/publication/268498029_Bounds_on_the_length_of_disjunctive_codes. Translation: *Problems of Information Transmission*. 18(3): 166–171.

[66] A. G. D'yachkov and V. V. Rykov. A survey of superimposed code theory. *Problems of Control and Information Theory*, 12(4):1–13, 1983. URL: https://www.researchgate.net/publication/235008674_Survey_of_Superimposed_Code_Theory.

[67] A. Emad and O. Milenkovic. Poisson group testing: A probabilistic model for nonadaptive streaming Boolean compressed sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, pages 3335–3339, 2014. DOI: 10.1109/ICASSP.2014.6854218.

[68] A. Emad and O. Milenkovic. Semiquantitative group testing. *IEEE Transactions on Information Theory*, 60(8):4614–4636, 2014. DOI: 10.1109/TIT.2014.2327630.

[69] A. Emad, K. R. Varshney, and D. M. Malioutov. A semiquantitative group testing approach for learning interpretable clinical prediction rules. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2015. URL: https://krvarshney.github.io/pubs/EmadVM_spars2015.pdf.

[70] P. Erdős and A. Rényi. On a classical problem of probability theory. *A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 6:215–220, 1961. URL: http://www.renyi.hu/~p_erdos/1961-09.pdf.

[71] P. Erdős and A. Rényi. On two problems of information theory. *A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 8:229–243, 1963. URL: https://www.renyi.hu/~p_erdos/1963-12.pdf.

[72] Y. Erlich, A. Gilbert, H. Ngo, A. Rudra, N. Thierry-Mieg, M. Wootters, D. Zielinski, and O. Zuk. Biological screens from linear codes: Theory and tools. *bioRxiv*, 2015. DOI: 10.1101/035352.

[73] Y. Erlich, A. Gordon, M. Brand, G. Hannon, and P. Mitra. Compressed genotyping. *IEEE Transactions on Information Theory*, 56(2):706–723, 2010. DOI: 10.1109/TIT.2009.2037043.

[74] M. Falahatgar, A. Jafarpour, A. Orlitsky, V. Pichapati, and A. T. Suresh. Estimating the number of defectives with group testing. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1376–1380, 2016. DOI: 10.1109/ISIT.2016.7541524.

[75] M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan. Group testing problems with sequences in experimental molecular biology. In *Proceedings of Compression and Complexity of SEQUENCES 1997*, pages 357–367, 1997. DOI: 10.1109/SEQUEN.1997.666930.

[76] A. Feinstein. A new basic theorem of information theory. *Transactions of the IRE Professional Group on Information Theory*, 4(4):2–22, 1954. DOI: 10.1109/TIT.1954.1057459.

[77] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley & Sons, 3rd edition edition, 1968.

[78] H. M. Finucan. The blood testing problem. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, pages 43–50, 1964. DOI: 10.2307/2985222.

[79] P. Fischer, N. Klasner, and I. Wegenera. On the cut-off point for combinatorial group testing. *Discrete Applied Mathematics*, 91(1):83–92, 1999. DOI: 10.1016/S0166-218X(98)00119-X.

[80] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, 2013. DOI: 10.1007/978-0-8176-4948-7.

[81] V. L. Freidlina. On a design problem for screening experiments. *Teoriya Veroyatnostei i ee Primeneniya*, 20(1):100–114, 1975. DOI: 10.1137/1120008. Translation: *Theory of Probability & Its Applications*. 20(1): 102–115.

[82] Z. Füredi. On $r$-cover-free families. *Journal of Combinatorial Theory, Series A*, 73(1):172–173, 1996. DOI: 10.1006/jcta.1996.0012.

[83] T. Furon. The illusion of group testing. Technical Report RR-9164, Inria Rennes Bretagne Atlantique, 2018. URL: https://hal.inria.fr/hal-01744252.

[84] T. Furon, A. Guyader, and F. Cérou. Decoding fingerprints using the markov chain monte carlo method. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 187–192, 2012. DOI: 10.1109/WIFS.2012.6412647.

[85] V. Gandikota, E. Grigorescu, S. Jaggi, and S. Zhou. Nearly optimal sparse group testing. In *54th Annual Allerton Conference on Communication, Control, and Computing*, pages 401–408, 2016. DOI: 10.1109/ALLERTON.2016.7852259.

[86] A. Ganesan, S. Jaggi, and V. Saligrama. Learning immune-defectives graph through group tests. In *IEEE International Symposium on Information Theory (ISIT)*, pages 66–70, 2015. DOI: 10.1109/ISIT.2015.7282418.

[87] A. Ganesan, S. Jaggi, and V. Saligrama. Non-adaptive group testing with inhibitors. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. DOI: 10.1109/ITW.2015.7133108.

[88] M. R. Garey and F. K. Hwang. Isolating a single defective using group testing. *Journal of the American Statistical Association*, 69(345):151–153, 1974. DOI: 10.2307/2285514.

[89] J. L. Gastwirth and P. A. Hammick. Estimation of the prevalence of a rare disease, preserving the anonymity of the subjects by group testing: Application to estimating the prevalence of AIDS antibodies in blood donors. *Journal of Statistical Planning and Inference*, 22(1):15–27, 1989. DOI: 10.1016/0378-3758(89)90061-X.

[90] O. Gebhard, M. Hahn-Klimroth, D. Kaaser, and P. Loick. Quantitative group testing in the sublinear regime, 2019, arXiv: 1905.01458.

[91] A. C. Gilbert, B. Hemenway, A. Rudra, M. J. Strauss, and M. Wootters. Recovering simple signals. In *2012 Information Theory and Applications Workshop (ITA)*, pages 382–391, 2012. DOI: 10.1109/ITA.2012.6181772.

[92] A. C. Gilbert, M. A. Iwen, and M. J. Strauss. Group testing and sparse signal recovery. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1059–1063, 2008. DOI: 10.1109/ACSSC.2008.5074574.

[93] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. Algorithmic linear dimension reduction in the $\ell_1$ norm for sparse vectors. In *Allerton Conference on Communication, Control, and Computing*, 2006. URL: https://web.eecs.umich.edu/~martinjs/papers/GSTV06-allerton.pdf.

[94] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. One sketch for all: Fast algorithms for compressed sensing. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing (STOC '07)*, pages 237–246, 2007. DOI: 10.1145/1250790.1250824.

[95] C. Gille, K. Grade, and C. Coutelle. A pooling strategy for heterozygote screening of the ΔF508 cystic fibrosis mutation. *Human Genetics*, 86(3):289–291, 1991. DOI: 10.1007/BF00202411.

[96] C. Goldie and R. Pinch. *Communication Theory*. Cambridge University Press, 1991. DOI: 10.1017/CBO9781139172448.

[97] M. T. Goodrich, M. J. Atallah, and R. Tamassia. Indexing information for data forensics. In *Applied Cryptography and Network Security*, pages 206–221, 2005. DOI: 10.1007/11496137_15.

[98] M. T. Goodrich and D. S. Hirschberg. Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis. *Journal of Combinatorial Optimization*, 15(1):95–121, 2008. DOI: 10.1007/s10878-007-9087-z.

[99] T. S. Han. *Information-Spectrum Methods in Information Theory*. Springer–Verlag, 2003. DOI: 10.1007/978-3-662-12066-8.

[100] N. J. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. W. Chan. Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In *26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 697–705. IEEE, 2007. DOI: 10.1109/INFCOM.2007.87.

[101] J. F. Hayes. An adaptive technique for local distribution. *IEEE Transactions on Communications*, 26(8):1178–1186, 1978. DOI: 10.1109/TCOM.1978.1094204.

[102] E. S. Hong and R. E. Ladner. Group testing for image compression. *IEEE Transactions on Image Processing*, 11(8):901–911, 2002. DOI: 10.1109/TIP.2002.801124.

[103] Y.-W. Hong and A. Scaglione. Group testing for sensor networks: The value of asking the right questions. In *38th Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1297–1301, 2004. DOI: 10.1109/ACSSC.2004.1399362.

[104] M. C. Hu, F. K. Hwang, and J. K. Wang. A boundary problem for group testing. *SIAM Journal on Algebraic Discrete Methods*, 2(2):81–87, 1981. DOI: 10.1137/0602011.

[105] W. Huleihel, O. Elishco, and M. Médard. Blind group testing. *IEEE Transactions on Information Theory*, 65(8):5050–5063, 2019. DOI: 10.1109/TIT.2019.2906607.

[106] F. K. Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608, 1972. DOI: 10.1080/01621459.1972.10481257.

[107] F. K. Hwang. A generalized binomial group testing problem. *Journal of the American Statistical Association*, 70(352):923–926, 1975. DOI: 10.1080/01621459.1975.10480324.

[108] H. A. Inan, P. Kairouz, M. Wootters, and A. Ozgur. On the optimality of the Kautz–Singleton construction in probabilistic group testing, 2018, arXiv: 1808.01457.

[109] P. Indyk. Deterministic superimposed coding with applications to pattern matching. In *38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 127–136, 1997. DOI: 10.1109/SFCS.1997.646101.

[110] P. Indyk, H. Q. Ngo, and A. Rudra. Efficiently decodable non-adaptive group testing. In *21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1126–1142, 2010. DOI: 10.1137/1.9781611973075.91.

[111] P. Johann. A group testing problem for graphs with several defective edges. *Discrete Applied Mathematics*, 117(1-3):99–108, 2002. DOI: 10.1016/S0166-218X(01)00181-0.

[112] O. T. Johnson. Strong converses for group testing from finite blocklength results. *IEEE Transactions on Information Theory*, 63(9):5923–5933, 2017. DOI: 10.1109/TIT.2017.2697358.

[113] O. T. Johnson, M. Aldridge, and J. Scarlett. Performance of group testing algorithms with near-constant tests-per-item. *IEEE Transactions on Information Theory*, 65(2):707–723, 2019. DOI: 10.1109/TIT.2018.2861772.

[114] A. B. Kahng and S. Reda. New and improved bist diagnosis methods from combinatorial group testing theory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(3):533–543, 2006. DOI: 10.1109/TCAD.2005.854635.

[115] R. M. Kainkaryam and P. J. Woolf. Pooling in high-throughput drug screening. *Current Opinion on Drug Discovery and Development*, 12(3):339, 2009. URL: https://www.ncbi.nlm.nih.gov/pubmed/19396735.

[116] R. M. Karp. *Reducibility among combinatorial problems*, pages 85–103. Springer US, 1972. DOI: 10.1007/978-1-4684-2001-2_9.

[117] C. R. Katholi, L. Toé, A. Merriweather, and T. R. Unnasch. Determining the prevalence of *onchocerca volvulus* infection in vector populations by polymerase chain reaction screening of pools of black flies. *Journal of Infectious Diseases*, 172(5):1414–1417, 1995. DOI: 10.1093/infdis/172.5.1414.

[118] G. O. H. Katona. Combinatorial search problems. In J. N. Srivastava, editor, *A Survey of Combinatorial Theory*, pages 285–308. North-Holland, 1973. DOI: 10.1016/B978-0-7204-2262-7.50028-4.

[119] W. Kautz and R. Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964. DOI: 10.1109/TIT.1964.1053689.

[120] T. Kealy, O. Johnson, and R. Piechocki. The capacity of non-identical adaptive group testing. In *52nd Annual Allerton Conference on Communication, Control, and Computing*, pages 101–108, 2014. DOI: 10.1109/ALLERTON.2014.7028442.

[121] S. A. Khan, P. Chowdhury, P. Choudhury, and P. Dutta. Detection of west nile virus in six mosquito species in synchrony with seroconversion among sentinel chickens in india. *Parasites and Vectors*, 10(1):13, 2017. DOI: 10.1186/s13071-016-1948-9.

[122] S. Khattab, S. Gobriel, R. Melhem, and D. Mosse. Live baiting for service-level DoS attackers. In *27th IEEE International Conference on Computer Communications (INFOCOM)*, pages 171–175, 2008. DOI: 10.1109/INFOCOM.2008.43.

[123] E. Knill, A. Schliep, and D. Torney. Interpretation of pooling experiments using the Markov chain Monte Carlo method. *Journal of Computational Biology*, 3:395–406, 1996. DOI: 10.1089/cmb.1996.3.395.

[124] J. Komlos and A. Greenberg. An asymptotically fast nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Transactions on Information Theory*, 31(2):302–306, 1985. DOI: 10.1109/TIT.1985.1057020.

[125] T. Laarhoven. Asymptotics of fingerprinting and group testing: Tight bounds from channel capacities. *IEEE Transactions on Information Forensics and Security*, 10(9):1967–1980, 2015. DOI: 10.1109/TIFS.2015.2440190.

[126] K. Lee, R. Pedarsani, and K. Ramchandran. Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes, 2015, arXiv: 1508.04485.

[127] C. H. Li. A sequential method for screening experimental variables. *Journal of the American Statistical Association*, 57(298):455–477, 1962. DOI: 10.1080/01621459.1962.10480672.

[128] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi. Group testing with prior statistics. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2346–2350, 2014. DOI: 10.1109/ISIT.2014.6875253.

[129] X. Li, S. Pawar, and K. Ramchandran. Sub-linear time compressed sensing using sparse-graph codes. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1645–1649, 2015. DOI: 10.1109/ISIT.2015.7282735.

[130] Z. Li, M. Fresacher, and J. Scarlett. Learning Erdős–Rényi random graphs via edge detecting queries, 2019, arXiv: 1905.03410.

[131] C. Lo, M. Liu, J. P. Lynch, and A. C. Gilbert. Efficient sensor fault detection using combinatorial group testing. In *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS),*, pages 199–206, 2013. DOI: 10.1109/DCOSS.2013.57.

[132] J. Luo and D. Guo. Neighbor discovery in wireless ad hoc networks based on group testing. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 791–797, 2008. DOI: 10.1109/ALLERTON.2008.4797638.

[133] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe. Node failure localization via network tomography. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC)*, pages 195–208, 2014. DOI: 10.1145/2663716.2663723.

[134] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. DOI: 10.2277/0521642981.

[135] A. J. Macula. Probabilistic nonadaptive and two-stage group testing with relatively small pools and DNA library screening. *Journal of Combinatorial Optimization*, 2(4):385–397, 1998. DOI: 10.1023/A:1009732820981.

[136] A. J. Macula. Probabilistic nonadaptive group testing in the presence of errors and dna library screening. *Annals of Combinatorics*, 3(1):61–69, 1999. DOI: 10.1007/BF01609876.

[137] A. J. Macula and L. J. Popyack. A group testing method for finding patterns in data. *Discrete Applied Mathematics*, 144(1):149–157, 2004. DOI: 10.1016/j.dam.2003.07.009.

[138] T. Madej. An application of group testing to the file comparison problem. In *9th International Conference on Distributed Computing Systems*, pages 237–243, 1989. DOI: 10.1109/ICDCS.1989.37952.

[139] D. M. Malioutov and M. Malyutov. Boolean compressed sensing: LP relaxation for group testing. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3305–3308, 2012. DOI: 10.1109/ICASSP.2012.6288622.

[140] D. M. Malioutov and K. R. Varshney. Exact rule learning via Boolean compressed sensing. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28:3 of *Proceedings of Machine Learning Research*, pages 765–773, 2013. URL: http://proceedings.mlr.press/v28/malioutov13.html.

[141] D. M. Malioutov, K. R. Varshney, A. Emad, and S. Dash. Learning interpretable classification rules with boolean compressed sensing. In T. Cerquitelli, D. Quercia, and F. Pasquale, editors, *Transparent Data Mining for Big and Small Data*, pages 95–121. Springer, 2017. DOI: 10.1007/978-3-319-54024-5_5.

[142] M. Malyutov and H. Sadaka. Maximization of ESI. Jaynes principle in testing significant inputs of linear model. *Random Operators and Stochastic Equations*, 6(4):311–330, 1998. DOI: 10.1515/rose.1998.6.4.311.

[143] M. B. Malyutov. The separating property of random matrices. *Matematicheskie Zametki*, 23(1):155–167, 1978. DOI: 10.1007/BF01104893. Translation: *Mathematical Notes of the Academy of Sciences of the USSR*. 23(1): 84–91.

[144] M. B. Malyutov. Search for sparse active inputs: A review. In H. Aydinian, F. Cicalese, and C. Deppe, editors, *Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede*, pages 609–647. Springer, 2013. DOI: 10.1007/978-3-642-36899-8_31.

[145] M. B. Malyutov and P. S. Mateev. Planning of screening experiments for a nonsymmetric response function. *Matematicheskie Zametki*, 27:109–127, 1980. DOI: 10.1007/BF01149816. Translation: *Mathematical Notes of the Academy of Sciences of the USSR*. 27(1): 57–68.

[146] A. Mazumdar. Nonadaptive group testing with random set of defectives. *IEEE Transactions on Information Theory*, 62(12):7522–7531, 2016. DOI: 10.1109/TIT.2016.2613870.

[147] C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics 1989: Invited Papers at the Twelfth British Combinatorial Conference*, pages 148–188. Cambridge University Press, 1989. DOI: 10.1017/CBO9781107359949.008.

[148] M. Mézard, M. Tarzia, and C. Toninelli. Group testing with random pools: Phase transitions and optimal strategy. *Journal of Statistical Physics*, 131(5):783–801, 2008. DOI: 10.1007/s10955-008-9528-9.

[149] M. Mézard and C. Toninelli. Group testing with random pools: Optimal two-stage algorithms. *IEEE Transactions on Information Theory*, 57(3):1736–1745, 2011. DOI: 10.1109/TIT.2010.2103752.

[150] E. Mossel, R. O'Donnell, and R. A. Servedio. Learning functions of $k$ relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. DOI: 10.1016/j.jcss.2004.04.002. Special Issue on STOC 2003.

[151] R. Mourad, Z. Dawy, and F. Morcos. Designing pooling systems for noisy high-throughput protein-protein interaction experiments using Boolean compressed sensing. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(6):1478–1490, 2013. DOI: 10.1109/TCBB.2013.129.

[152] E. Nebenzahl and M. Sobel. Finite and infinite models for generalized group-testing with unequal probabilities of success for each item. In *Discriminant Analysis and Applications*, pages 239–289. Elsevier, 1973. DOI: 10.1016/B978-0-12-154050-0.50020-4.

[153] H. Q. Ngo, E. Porat, and A. Rudra. Efficiently decodable error-correcting list disjunct matrices and applications. In *Automata, Languages and Programming (ICALP)*, pages 557–568, 2011. DOI: 10.1007/978-3-642-22006-7_47.

[154] Y. Polyanskiy, H. V. Poor, and S. Verdú. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, 2010. DOI: 10.1109/TIT.2010.2043769.

[155] E. Porat and A. Rothschild. Explicit nonadaptive combinatorial group testing schemes. *IEEE Transactions on Information Theory*, 57(12):7982–7989, 2011. DOI: 10.1109/TIT.2011.2163296.

[156] G. Reeves and H. D. Pfister. Understanding phase transitions via mutual information and MMSE, 2019, arXiv: 1907.02095.

[157] L. Riccio and C. J. Colbourn. Sharper bounds in adaptive group testing. *Taiwanese Journal of Mathematics*, 4(4):669–673, 2000. DOI: 10.11650/twjm/1500407300.

[158] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511791338.

[159] M. Ruszinkó. On the upper bound of the size of the $r$-cover-free families. *Journal of Combinatorial Theory, Series A*, 66(2):302–310, 1994. DOI: 10.1016/0097-3165(94)90067-1.

[160] J. Scarlett. An efficient algorithm for capacity-approaching noisy adaptive group testing. In *IEEE International Symposium on Information Theory (ISIT)*, 2019.

[161] J. Scarlett. Noisy adaptive group testing: Bounds and algorithms. *IEEE Transactions on Information Theory*, 65(6):3646–3661, 2019. DOI: 10.1109/TIT.2018.2883604.

[162] J. Scarlett and V. Cevher. Converse bounds for noisy group testing with arbitrary measurement matrices. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2868–2872, 2016. DOI: 10.1109/ISIT.2016.7541823.

[163] J. Scarlett and V. Cevher. Phase transitions in group testing. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 40–53, 2016. DOI: 10.1137/1.9781611974331.ch4.

[164] J. Scarlett and V. Cevher. How little does non-exact recovery help in group testing? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6090–6094, 2017. DOI: 10.1109/ICASSP.2017.7953326.

[165] J. Scarlett and V. Cevher. Limits on support recovery with probabilistic models: An information-theoretic framework. *IEEE Transactions on Information Theory*, 63(1):593–620, 2017. DOI: 10.1109/TIT.2016.2606605.

[166] J. Scarlett and V. Cevher. Phase transitions in the pooled data problem. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 377–385, 2017. URL: https://papers.nips.cc/paper/6641-phase-transitions-in-the-pooled-data-problem.

[167] J. Scarlett and V. Cevher. Near-optimal noisy group testing via separate decoding of items. *IEEE Journal of Selelected Topics in Signal Processing*, 2(4):625–638, 2018. DOI: 10.1109/JSTSP.2018.2844818.

[168] J. Scarlett and O. T. Johnson. Noisy non-adaptive group testing: A (near-)definite defectives approach, 2018, arXiv: 1808.09143.

[169] A. Schliep, D. C. Torney, and S. Rahmann. Group testing with DNA chips: Generating designs and decoding experiments. In *IEEE Bioinformatics Conference*, pages 84–91, 2003. DOI: 10.1109/CSB.2003.1227307.

[170] A. Sebő. On two random search problems. *Journal of Statistical Planning and Inference*, 11(1):23–31, 1985. DOI: 10.1016/0378-3758(85)90022-9.

[171] D. Sejdinovic and O. T. Johnson. Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 998–1003, 2010. DOI: 10.1109/ALLERTON.2010.5707018.

[172] P. Sham, J. S. Bader, I. Craig, M. O'Donovan, and M. Owen. Dna pooling: A tool for large-scale association studies. *Nature Reviews Genetics*, 3(11):862–871, 2002. DOI: 10.1038/nrg930.

[173] C. Shangguan and G. Ge. New bounds on the number of tests for disjunct matrices. *IEEE Transactions on Information Theory*, 62(12):7518–7521, 2016. DOI: 10.1109/TIT.2016.2614726.

[174] C. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956. DOI: 10.1109/TIT.1956.1056798.

[175] C. E. Shannon. Certain results in coding theory for noisy channels. *Information and Control*, 1(1):6–25, 1957. DOI: 10.1016/S0019-9958(57)90039-6.

[176] A. Sharma and C. R. Murthy. Group testing-based spectrum hole search for cognitive radios. *IEEE Transactions on Vehicular Technology*, 63(8):3794–3805, 2014. DOI: 10.1109/TVT.2014.2305978.

[177] N. Shental, A. Amir, and O. Zuk. Identification of rare alleles and their carriers using compressed se(que)nsing. *Nucleic Acids Research*, 38(19):e179, 2010. DOI: 10.1093/nar/gkq675.

[178] M. Shi, T. Furon, and H. Jégou. A group testing framework for similarity search in high-dimensional spaces. In *Proceedings of the ACM International Conference on Multimedia*, pages 407–416, 2014. DOI: 10.1145/2647868.2654895.

[179] M. Sobel and R. Elashoff. Group testing with a new goal, estimation. *Biometrika*, pages 181–193, 1975. DOI: 10.2307/2334502.

[180] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell Labs Technical Journal*, 38(5):1179–1252, 1959. DOI: 10.1002/j.1538-7305.1959.tb03914.x.

[181] M. Sobel and P. A. Groll. Binomial group-testing with an unknown proportion of defectives. *Technometrics*, 8(4):631–656, 1966. DOI: 10.1080/00401706.1966.10490408.

[182] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996. DOI: 10.1109/18.556668.

[183] M. R. Stan, P. D. Franzon, S. C. Goldstein, J. C. Lach, and M. M. Ziegler. Molecular electronics: From devices and interconnect to circuits and architecture. *Proceedings of the IEEE*, 91(11):1940–1957, 2003. DOI: 10.1109/JPROC.2003.818327.

[184] A. Sterrett. On the detection of defective members of large populations. *The Annals of Mathematical Statistics*, 28(4):1033–1036, 1957. DOI: 10.1214/aoms/1177706807.

[185] W. H. Swallow. Group testing for estimating infection rates and probabilities of disease transmission. *Phytopathology*, 75(8):882–889, 1985. DOI: 10.1094/Phyto-75-882.

[186] K. H. Thompson. Estimation of the proportion of vectors in a natural population of insects. *Biometrics*, 18(4):568–578, 1962. DOI: 10.2307/2527902.

[187] M. Tilghman, D. Tsai, T. P. Buene, M. Tomas, S. Amade, D. Gehlbach, S. Chang, C. Ignacio, G. Caballero, S. Espitia, S. May, E. V. Noormahomed, and D. Smith. Pooled nucleic acid testing to detect antiretroviral treatment failure in hiv-infected patients in mozambique. *Journal of Acquired Immune Deficiency Syndromes*, 70(3):256, 2015. DOI: 10.1097/QAI.0000000000000724.

[188] M. A. Tsfasman, S. G. Vlădut, and T. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982. DOI: 10.1002/mana.19821090103.

[189] X. M. Tu, E. Litvak, and M. Pagano. On the informativeness and accuracy of pooled testing in estimating prevalence of a rare disease: Application to HIV screening. *Biometrika*, pages 287–297, 1995. DOI: 10.2307/2337408.

[190] M. K. Varanasi. Group detection for synchronous Gaussian code-division multiple-access channels. *IEEE Transactions on Information Theory*, 41(4):1083–1096, 1995. DOI: 10.1109/18.391251.

[191] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001. DOI: 10.1007/978-3-662-04565-7.

[192] T. Wadayama. Nonadaptive group testing based on sparse pooling graphs. *IEEE Transactions on Information Theory*, 63(3):1525–1534, 2017. DOI: 10.1109/TIT.2016.2621112. See also [193].

[193] T. Wadayama. Comments on 'nonadaptive group testing based on sparse pooling graphs'. *IEEE Transactions on Information Theory*, 64(6):4686–4686, 2018. DOI: 10.1109/TIT.2018.2827463.

[194] S. D. Walter, S. W. Hildreth, and B. J. Beaty. Estimation of infection rates in populations of organisms using pools of variable size. *American Journal of Epidemiology*, 112(1):124–128, 1980. DOI: 10.1093/oxfordjournals.aje.a112961.

[195] C. Wang, Q. Zhao, and C.-N. Chuah. Optimal nested test plan for combinatorial quantitative group testing. *IEEE Transactions on Signal Processing*, 66(4):992–1006, 2018. DOI: 10.1109/TSP.2017.2780053.

[196] J. Wang, E. Lo, and M. L. Yiu. Identifying the most connected vertices in hidden bipartite graphs using group testing. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2245–2256, 2013. DOI: 10.1109/TKDE.2012.178.

[197] J. K. Wolf. Born again group testing: Multiaccess communications. *IEEE Transactions on Information Theory*, 31(2):185–191, 1985. DOI: 10.1109/TIT.1985.1057026.

[198] S. Wu, S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan. Achievable partition information rate over noisy multi-access boolean channel. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1206–1210, 2014. DOI: 10.1109/ISIT.2014.6875024.

[199] S. Wu, S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan. Partition information and its transmission over boolean multi-access channels. *IEEE Transactions on Information Theory*, 61(2):1010–1027, 2015. DOI: 10.1109/TIT.2014.2375211.

[200] W. Xu, M. Wang, E. Mallada, and A. Tang. Recent results on sparse recovery over graphs. In *45th Asilomar Conference on Signals, Systems and Computers*, pages 413–417, 2011. DOI: 10.1109/ACSSC.2011.6190031.

[201] Y. Xuan, I. Shin, M. T. Thai, and T. Znati. Detecting application denial-of-service attacks: A group-testing-based approach. *IEEE Transactions on Parallel and Distributed Systems*, 21(8):1203–1216, 2010. DOI: 10.1109/TPDS.2009.147.

[202] N. Zaman and N. Pippenger. Asymptotic analysis of optimal nested group-testing procedures. *Probability in the Engineering and Informational Sciences*, 30(4):547–552, 2016. DOI: 10.1017/S0269964816000267.

[203] W. Zhang and L. Huang. On or many-access channels. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2638–2642, 2017. DOI: 10.1109/ISIT.2017.8007007.