Group Testing: A Probably Approximately Correct Analysis and Recovery Algorithms for pooled RT-qPCR

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Sameera Bharadwaja H.

under the Guidance of

Prof. Chandra R. Murthy
and
Dr. Shashank S. Agashe

Electrical Communication Engineering
Indian Institute of Science, Bangalore
Bangalore – 560 012 (INDIA)

August 2024

To My Family

Acknowledgments

I will be eternally grateful to my advisor *Prof. Chandra R. Murthy* for his continual support and encouragement throughout my PhD journey. Every meeting during my study has been fruitful to a significant degree. His dedication, passion, and interest in research were infectious. I am happy that some of these qualities have enabled me to grow personally and professionally, further motivating much of this dissertation. It has been a pleasure working under his supervision. I am thankful for all our discussions during the progress meetings. Watching him effortlessly perform multiple tasks very efficiently and perfectly was a treat. I learned how to formulate problems, different perspectives from which one could address the problem, and quite a bag full of tips and tricks to present the results and write papers, to name a few. Thank you for being an excellent mentor and guide during my PhD study!

I am equally thankful to *Dr. Shashank S. Agashe*, my co-advisor from my organization, who, despite his primary responsibility as the head of the Smart Equipment Solution Group (SESG) at Samsung Semiconductor India Research (SSIR), Bangalore, has always been able to carve out time for any discussions related to my progress. He helped me maintain the required healthy work-life balance at SSIR, which enabled me to complete my PhD journey. I also thank my mentor *Prof. Manoj Choudhary*, former senior director at SSIR, now the vice chancellor at Gati Shakti Vishwavidyalaya, Vadodara, India, who supported me during the initial stages of the coursework till my comprehensive examination.

I thank my organization, Samsung Semiconductor India Research (SSIR), Bangalore, for its fantastic initiative to upskill its employees and help them grow professionally. I thank SSIR MD *CEVP Balajee Sowrirajan* for letting me pursue a doctoral degree at the Indian Institute of Science (IISc), Bangalore. I want to thank all the members of SSIR learning & development (L&D) and the finance team for helping me with all the administrative work from SSIR. In addition, I thank my group members, *Gaurav Kumar, Lakshmi, Harish, Naveen*, and *Rajesh*, for their moral support and all the time spent together in the office. I want to extend my gratitude to my project team, *Sachin*, *Argho*, *Deepanshu*, *Gururajan*, *Gururaja*, *Raju*, *Rajaneesh*, *Kumar* and *Kavya*, who took over the responsibilities of the project while I was out of the office for the coursework. I want to thank all my group members from SESG for their support during my PhD journey.

I want to thank IISc for providing a fantastic infrastructure and environment that is fit for honing one's intellectual ability. I thank all the institute faculty, *Prof. Chandra R. Murthy, Prof. B. Sundar Rajan, Prof. Parimal Parag, Prof. Sundeep P. Chepuri* and *Prof. Utpal Mukherji*, whose courses guided me during my initial days of returning to academia as a student after more than a decade of industrial assignments. I thank my comprehensive examination panel and the annual review committee, *Prof. Chandra Sekhar Seelamantula*, *Prof. Joy Kuri*, *Prof. Neelesh B. Mehta* and *Prof. Rahul Singh*, for their constructive feedback during the progress review meetings. I thank ECE office members *Mr. R. Srinivas Murthy*, *Ms. Suma T. C.*, *Mr. Jaseem* and *Ms. Rajani B.* for their prompt administrative support during my coursework and visits to the campus.

I thank *Dr. Ribhu Chopra*, who introduced me to *Prof. Chandra Murthy's* amazing research lab group, the Signal Processing for Communications (SPC) lab. I remember my visit to the campus to meet Ribhu, followed by walking through the streets of Malleshwaram to MTR, by which Ribhu was able to motivate me enough to take more concrete steps towards getting myself enrolled. There was no looking back and in retrospect, I will forever be grateful for *that* meeting.

I am incredibly grateful to the alums and current members of the SPC lab: *Anubhab*, *Sai*, *Kanchan*, *Chirag*, *Yashvanth*, *Dheeraj*, *Lekshmi*, *Unni*, *Pradip*, *Abhinav*, *Soumendu*, *Chandrasekhar*, *Rupam*, *Niladri*, *Krishna* and *Prabhat* for making my initial days relatively seamless and more enjoyable. I will miss long discussions, technical and otherwise, with *Sai*, *Chirag* and *Anubhab* over WhatsApp and in person. *Anubhab*, *Sai* and *Kanchan* have always supported my countless requests for lab PC checks or restarts. *Chirag*, *Anubhab* and *Sai* kept my morale high during the preparation phase for the comprehensive examination and the pandemic. I want to thank *Dr. Sarvendranath Rimalapudi*, the TA for the Foundations of Machine Learning (FML) course. Without him, assignments involving numerous theorems and lemmas would have been impossible to understand. I thank all my batch mates from all the courses for making it intellectually enjoyable.

I am immensely grateful to my parents, *Hayavadana* and *Parimala*, who worked very hard and have supported me unconditionally. My loving wife, *Brinda*, and daughter, *Samhita*, have been patient throughout my PhD journey. I thank them for their support and sacrifices to help me complete this journey. My younger brother, *Surg. Lt. Sanjeeva* (Indian Navy) has motivated me immensely. I thank all my relatives, cousins, and grandparents for their support and love.

Abstract

The goal of group testing, also called pool testing, is to *successfully* identify a set of k *defectives* from a population of n *items* using only m ($< n$) *group tests*. In each group test, a subset of the n items is tested together as dictated by a *pooling protocol*. The outcome of a group test is negative if and only if none of the defective items participate in that group test; it is positive otherwise. When $k \ll n$, group testing can help significantly reduce the number of tests needed. A decoding algorithm then estimates the *defective item set* using the group test outcomes and the knowledge of the pooling protocol.

From its first introduction by Dorfman in 1943, group testing has been adopted in various applications like infectious disease detection, multiple-access protocols, cognitive radio, and product testing, to name a few. The primary motivating factors for its widespread success include improving the test reliability and reducing the testing cost. On the other hand, the theoretical study involving the design of pooling protocols, development of decoding algorithms, deriving sufficiency (achievability), and converse bounds on the number of tests under various scenarios can be found in a rich literature spanning over half a century. Recently, group testing has gained a renewed interest in the eyes of both practitioners and theorists alike due to the outbreak of COrona-VIrus Disease (Covid-19) pandemic.

In the first part of this thesis, we focus on deriving sufficiency bounds on the number of tests for Boolean non-adaptive group testing algorithms, namely, Combinatorial Orthogonal Matching Pursuit (COMP) and Definite Defectives (DD) with random pooling. The term *non-adaptive* means that all the group tests are conducted in a single stage, possibly in parallel. The term *random pooling* refers to the fact that the strategy used to determine which item participates in which group test is determined by a probability distribution.

We view the group testing problem through the lens of a function learning problem and formulate it in a probably approximately correct (PAC) analysis framework. This enables us to characterize our sufficiency bounds by a *confidence parameter* and an *approximation error tolerance parameter*. In practical settings with finite resources, one is often interested in ensuring that the probability of the error incurred by a function learned using a finite number of randomly drawn samples exceeding a threshold remains below a small number, which we call the *confidence parameter*. Also, approximate defective set recovery is sufficient in many applications, wherein the number of errors that can be tolerated is quantified by the *approximation error tolerance parameter*. Our resulting sufficiency bounds provide a finer perspective of the random-pooling-based group testing algorithms by separately accounting for the randomness in the pooling protocol and the defective set identification errors.

In the second part of this thesis, we focus on developing recovery algorithms for Covid-19 infected sample detection with pooled Reverse Transcriptase (quantitative) Polymerase Chain Reaction (RT-qPCR) assay. The quantitative output from the RT-qPCR test is called the cycle threshold (CT), a quantity inversely related to the amount of the viral load in the sample. For a healthy sample, the CT returned is, in theory, infinity. Existing recovery algorithms suffer from challenges related to the non-linear nature of the RT-qPCR model, existence of infinities in the feasible set of the optimization problem, and are sensitive to the unknown PCR efficiency factor, q. We develop two iterative algorithms to address the above-mentioned gaps: 1) alternating direction method of multipliers CT (ADMM-CT) and 2) block coordinate descent CT (BCD-CT). At the heart of these algorithms lie gradient descent (GD-CT) and iterative mirrored hard thresholding (IMHT-CT) algorithms for individual sample CT estimation and a projected gradient descent (PGD) method for estimating q. Lastly, we present empirical results demonstrating the advantage of using quantitative measurements in non-adaptive pool testing in terms of the testing rate and, hence, the cost on publicly available Covid-19 data on the number of tests conducted and also compile the best rates achievable for a given prevalence rate.

In summary, we address these two aspects of group testing in this thesis: 1) theoretical analysis of Boolean non-adaptive group testing algorithms and 2) developing recovery algorithms to detect Covid-19 using pooled RT-qPCR. The key takeaways are as follows:

- Unlike the traditional PAC framework, our formulation allows for deriving bounds under zero error or exact recovery scenario. Further, we can *choose the data distribution from which the samples are drawn* for function learning based on our knowledge of the hypothesis space from which the target function is learned.

- Our bounds consider both the *randomness in the test matrix* and the *approximation error probability*. This makes existing bounds a special case of our PAC bounds. In addition, we characterize a *lower bound on the cumulative distribution of the approximation errors*. In deriving these bounds, we characterize the expected stopping time and the tail probability for the subset coupon collector problem (SCCP).

- Pooled RT-qPCR based Covid-19 detection algorithms that *jointly estimate* both individual sample CTs and q are robust to model uncertainties.

- Finally, we collate *best optimal designs given a prevalence rate* to help the group testing practitioners make the best use of the theoretical results available on group testing.

We show the *tightness of our bounds* and *benchmark* the performances of our algorithms with several existing comparable methods to validate the utility of our algorithms.

Glossary

Acronym	Definition
ADMM	Alternating Direction Method of Multipliers
ADMM-CT	Alternating Direction Method of Multipliers - Cycle Threshold
BCD	Block Coordinate Descent
BCD-CT	Block Coordinate Descent - Cycle Threshold
BP	Basis Pursuit
CB	Counting Bound
CBP	Combinatorial Basis Pursuit
CCP	Coupon Collector Problem
CDC	Center for Disease Control and Prevention
cDNA	Complementary DeoxyriboNucleic Acid
CDF	Cumulative Distribution Function
CGT	Combinatorial Group Testing
CoCo	Coupon Collector
COMP	Combinatorial Orthogonal Matching Pursuit
CoMa	Column Matching
Covid-19	COrona VIrus Disease - 2019
CS	Compressed Sensing
CSG	Computational-Statistical Gap
CT	Cycle Threshold
CVX	A Convex Optimization Toolbox
DD	Definite Defectives
DNA	DeoxyriboNucleic Acid
ETI	Expected number of Tests per Infected
FN	False Negative
FNR	False Negative Rate

FP	False Positive
FPR	False Positive Rate
GD	Gradient Descent
GES	Generalized Euler Squares
GR	Gaussian Randomization
GT	Group Testing
ICMR	Indian Council of Medical Research
IHT	Iterative Hard Thresholding
IMHT	Iterative Mirrored Hard Thresholding
KTS	Kirkman Triple System
LAD	Least Absolute Deviation
LASSO	Linear Absolute Shrinkage and Selection Operator
LiPo	Linear Programming decoding
LS	Least Squares
MINLP	Mixed-Integer Non-Linear Programming
ML	Machine Learning
MSE	Mean Squared Error
NC	Negative Control
NCBP	Noisy CBP
NCOMP	Noisy COMP
NMSE	Normalized Mean Squared Error
NN-LAD	Non-Negative Least Absolute Deviation
NN-LASSO	Non-Negative Linear Absolute Shrinkage and Selection Operator
NN-LS	Non-Negative Least Squares
NN-OMP	Non-Negative Orthogonal Matching Pursuit
NP-hard	Non-Deterministic Polynomial-Time hard
OMP	Orthogonal Matching Pursuit
PAC	Probably Approximately Correct
PC	Positive Control
PCR	Polymerase Chain Reaction
PDF	Probability Distribution Function
PDS	Probable Defective Set
PGD	Projected Gradient Descent

PGT	Probabilistic Group Testing
RNA	RiboNucleic Acid
RT-qPCR	Reverse Transcriptase (quantitative) Polymerase Chain Reaction
SARS-CoV-2	Severe Acute Respiratory Syndrome related CoronaVirus 2
SBL	Sparse Bayesian Learning
SCCP	Subset Coupon Collector Problem
SCOMP	Sequential Combinatorial Orthogonal Matching Pursuit
SSS	Smallest Satisfying Set
STS	Steiner Triple System
WHO	World Health Organization

Notation

Vectors and matrices are denoted by boldface small and capital letters, respectively. Sets are denoted by calligraphy letters. The rest of the notation is listed below.

Vectors

$\mathbf{x} = [x_1, \ldots, x_n]$	:	A vector $\mathbf{x}$ whose elements are $x_1, \ldots, x_n$
x_j	:	jth element of a vector $\mathbf{x}$
$\mathbf{0}$	:	An all-zero vector
$\mathbf{1}$	:	An all-ones vector
$\|\mathbf{x}\|_0$	:	l_0 norm of a vector $\mathbf{x}$
		That is, the count of non-zero elements in $\mathbf{x}$.
$\|\mathbf{x}\|_1$	:	l_1 norm of a vector $\mathbf{x}$
$\|\mathbf{x}\|_2$	:	l_2 norm of a vector $\mathbf{x}$
$\|\mathbf{x}\|_\infty$	:	l_∞ norm of a vector $\mathbf{x}$,
		That is, the absolute value of the largest component of $\mathbf{x}$.

Matrices

$\mathbf{I}_n$	:	Identity matrix of dimension $n \times n$
$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$	:	A matrix $\mathbf{A}$ whose columns are $\mathbf{a}_1, \ldots, \mathbf{a}_n$
a_{ij}	:	(i, j)th entry of a matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	:	Inverse of a matrix $\mathbf{A}$
$\mathbf{A}^T$	:	Transpose of a matrix $\mathbf{A}$
$\lvert\lvert\lvert\mathbf{A}\rvert\rvert\rvert_2$	:	Spectral norm of a matrix $\mathbf{A}$
$\|\mathbf{A}\|_F$	:	Frobenius norm of a matrix $\mathbf{A}$
$\texttt{trace}\,(\mathbf{A})$	:	Trace of a matrix $\mathbf{A}$
$\mathbf{A} \succcurlyeq 0$	:	A matrix $\mathbf{A}$ is positive semi-definite.

Probability

$\mathbb{E}\left[.\right]$	:	Expectation of a random variable/vector
$\mathtt{var}\left(.\right)$	:	Variance of a random variable/vector
$\mathcal{N}(\mu, \sigma^2)$	:	Normal distribution with mean μ and variance σ^2
Unif(a, b)	:	Uniform distribution in range $[a, b]$
$\mathcal{B}(p)$	:	Bernoulli distribution with parameter, $p \in [0, 1]$
Bin(n, p)	:	Binomial distribution with parameters, $n \in \mathbb{N}$ and $p \in [0, 1]$
a.s.	:	Almost sure convergence
i.i.d.	:	Independent and identically distributed
$x \sim \mathcal{D}$	:	x is drawn from a distribution $\mathcal{D}$

Bachmann-Landau

$f(n) = O(g(n))$	:	f is bounded above by g asymptotically.
		$\exists k > 0\ \exists n_0, \forall n > n_0 : \lvert f(n)\rvert \leq kg(n)$ OR $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$.
$f(n) = \Theta(g(n))$	:	f is bounded above and below by g asymptotically.
		$\exists k_1 > 0\ \exists k_2 > 0\ \exists n_0, \forall n > n_0 : k_1 g(n) \leq f(n) \leq k_2 g(n)$.
$f(n) = o(g(n))$	:	f is dominated by g asymptotically.
		$\forall k > 0\ \exists n_0, \forall n > n_0 : \lvert f(n)\rvert < kg(n)$ OR $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

Set

$\lvert\mathcal{A}\rvert$	:	Cardinality of the set $\mathcal{A}$
$\mathcal{A} \cup \mathcal{B}$	:	Union of the set $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \cap \mathcal{B}$	:	Intersection of the set $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \backslash \mathcal{B}$	:	Set difference: set of elements in $\mathcal{A}$ that are not in $\mathcal{B}$
$\mathcal{A}'$	:	Complement of the set $\mathcal{A}$
$\mathcal{A} \subseteq \mathcal{B}$	:	$\mathcal{A}$ is subset of $\mathcal{B}$
$\mathcal{A} \supseteq \mathcal{B}$	:	$\mathcal{A}$ is superset of $\mathcal{B}$
$j \in \mathcal{X}$	:	j is an element of the set $\mathcal{X}$
$j \notin \mathcal{X}$	:	j is not an element of the set $\mathcal{X}$
$\emptyset$	:	Null set or empty set
$[n]$	:	A set $\{1, 2, \ldots, n\}$

Field

$\mathbb{N}$	:	Field of natural numbers
$\mathbb{Z}$	:	Field of integers
$\mathbb{Z}_+$	:	Field of positive integers
$\mathbb{R}$	:	Field of real numbers
$\mathbb{R}_+$	:	Field of positive real numbers

Miscellaneous

$\mathbb{R}^{m\times n}$	:	The set of real-valued $m \times n$ matrices
$\mathbb{R}^{n\times 1}$	:	The set of real-valued $n \times 1$ vectors
$\mathbb{N}^{m\times n}$	:	The set of natural number-valued $m \times n$ matrices
$\mathbb{N}^{n\times 1}$	:	The set of natural number-valued $n \times 1$ vectors
$\{0,1\}^{m\times n}$	:	The set of binary-valued $m \times n$ matrices
$\{0,1\}^{n}$	:	The set of binary-valued $n \times 1$ vectors
$a \sim b$	:	$a \in \mathbb{R}$ is of same order as $b \in \mathbb{R}$.
$\lceil x \rceil$	:	The smallest integer greater than or equal to the scalar $x \in \mathbb{R}$
$\lfloor x \rfloor$	:	The greatest integer less than or equal to the scalar $x \in \mathbb{R}$
$\log_M(x)$	:	Logarithm of x using the base $M \in \mathbb{Z}_+$
$\ln(x)$	:	Natural Logarithm of x, i.e., logarithm of x to base e
$a \vee b$	:	Logical OR-ing operation between $a, b \in \{0,1\}$.
$\binom{n}{k}$	:	Number of ways to choose k items out of n items.
$\nabla_{\mathbf{x}} f(\mathbf{x})$	:	Gradient of $f(\mathbf{x})$ w.r.t. $\mathbf{x}$.
f^{**}	:	Convex Biconjugate of function $f(x)$
$n!$	:	Factorial of $n \in \mathbb{Z}_+$ with $0! \triangleq 1$.

Thesis-Specific Notation

The following notations are common across chapters. Any notation used only in a specific chapter is explicitly defined in the corresponding chapter.

n	:	Number of items in the population to be tested
m	:	Number of the group tests
m_S	:	Sufficient number of the group tests
k	:	Number of *defective* items ($\ll n$)
$\mathbf{A}$	:	Pooling or Test Matrix of size $m \times n$
$\mathcal{K}$	:	The defective item set
$\hat{\mathcal{K}}$	:	Estimate of the defective item set
ϵ	:	The approximation error tolerance probability in PAC analysis
g_ϵ	:	Number of false positive (FP) errors that can be tolerated
d_ϵ	:	Number of false negative (FN) errors that can be tolerated
δ	:	The confidence parameter in PAC analysis
σ_ϵ	:	The standard deviation of the measurement noise in the pooled RT-qPCR
$\boldsymbol{\epsilon}$	:	The measurement noise vector in the pooled RT-qPCR

Contents

List of Figures

List of Tables

1 | Introduction

Chapter Highlights

This chapter presents a brief overview of the group testing problem. Group testing deals with the problem of detecting a set of *defective* items from a population by performing *group tests*. Each group test is conducted on a pool or group containing a subset of items from the population. The outcomes of the group tests are used along with the knowledge of how the items were pooled to recover an estimate of the set of defective items. Within the purview of group testing, several different testing protocols, including how the tests are conducted, designing various pooling strategies, algorithms to recover the defective set, analysis of the algorithms in terms of the number of group tests required under various settings such as the recovery tolerance, the error probability definition, relationship between the number of defective items and the population size under test, to name a few, have been of interest in the literature. This thesis focuses on non-adaptive group testing algorithms.

The first part of the thesis presents a probably approximately correct (PAC) learning-based reformulation of the group testing problem. The PAC-learning view of group testing enables us to derive sufficiency bounds on the number of tests under both *exact* and *approximate recovery* conditions at various confidence levels. The proposed PAC bounds reduce to the sufficiency bounds found in the literature under exact recovery conditions asymptotically. Further, an order-wise analysis of the PAC bounds under both exact and approximate recovery scenarios is presented. Experiments show a good agreement between the simulation results and our theoretical bounds.

The second part of the thesis presents novel algorithms for detecting Covid-19 using pooled reverse Transcriptase (quantitative) polymerase chain reaction (RT-qPCR) testing. The formulation of the group testing problem with the RT-qPCR model poses challenges related to non-linearity and the introduction of infinities in the feasible set of the optimization problem. Convergence results for the proposed algorithms are discussed. The proposed algorithms converge to a local optimum, and numerical simulations show the effectiveness of the proposed algorithms.

1.1 Background

At the height of World War II, in 1943, the US Public Health Service and the Selective Service System were engaged in a program to identify all the syphilitic men before inducting prospective soldiers into the army [1]. The conventional process at the time was to subject each prospective inductee to a Wassermann complement fixation test [2]. The test was divided into two stages: 1) draw a blood sample from the personnel; 2) subject the collected sample to a lab analysis to detect syphilitic antigen. The presence of an antigen is a good indicator of the infection.

In order to overcome the scarcity of the reagents required for the test and thereby ensure a sufficient supply of reagents to screen all the personnel, Prof. Robert Dorfman proposed a procedure where the tests are conducted on the *pooled* blood samples in his report titled *"The detection of defective members of large populations"* [1]. The idea of *pool testing* as described by Dorfman starts by pooling a set of, say, five blood samples together and subjecting it to a single Wassermann-type test. If there are no infected blood samples in a pool, the test outcome will be negative for the presence of the antigen, and all individuals participating in the test are declared healthy. On the other hand, a positive pool test means that at least one infected sample is present in the pool. In the second stage, the individual samples from the positive pools are tested again individually to identify the infected samples.

In the context of the above-described testing procedure, Dorfman presented the answer to the following two questions: 1) Does this testing protocol result in fewer chemical tests for a population size n under the screening process compared to the individual testing? 2) What is the most efficient size of the pool, where the term *efficiency* refers to the most savings obtainable in the number of tests relative to the individual testing? The answer to both questions is that the amount of savings in the number of tests and the optimal pool size are a function of the prevalence rate of the disease. In particular, group testing is optimal, and the benefits are higher, at a lower prevalence rate. Beyond a threshold prevalence rate, individual testing is optimal, i.e., the best pool size is to use a single sample per test.

Over the past 8 decades, group testing has been extensively studied, under different testing protocols, analyzing the necessary and sufficient number of tests, considering the effects of dilution and pooling errors, to name a few. In the subsequent sections of this chapter, we present a brief overview of group testing and describe the various applications, benefits, and challenges of group testing, followed by the focus of this thesis.

1.2 A Brief Overview of Group Testing

In general, identifying a set of k *defectives* from a population of n *items* is an interesting problem. A näive solution is to test the items individually, which requires n tests. This approach is inefficient if n is very large or there are constraints on time-to-test, cost budget, or testing hardware and resource constraints.

An alternative is to pool items together and run $m < n$ tests in parallel, with each test pooling a subset of the items together. These are collectively called group testing algorithms (or pool testing algorithms) and work as follows. The outcome of a group test is negative if and only if none of the defective items participate in that group test; it is positive otherwise. When $k \ll n$, which holds in many applications like identifying a rare disease from a set of blood samples, testing a population for an infection in the early stage of an epidemic, identifying defective industrial products in a high-yield production line, etc., this approach can help significantly reduce the number of tests needed.

Group testing methods can be categorized into *adaptive* and *non-adaptive* types [3]. In adaptive group testing, the tests are performed in multiple stages wherein the pool/group design in the current stage depends on the test outcomes from the previous stage. In contrast, in the non-adaptive method, all the required tests are performed in a single stage, followed by the application of a suitable decoding algorithm [4–6] to recover the individual *item status* (i.e., defective or non-defective). A few important decoding algorithms available in the literature include Combinatorial Orthogonal Matching Pursuit (COMP),[1] Definite Defectives (DD), Sequential COMP (SCOMP), Smallest Satisfying Set (SSS), Linear Programming (LiPo) algorithm [7]. In non-adaptive group testing, the information about which item(s) participate in which group test(s) is stored in a pooling/testing matrix.

Group testing was first introduced by Dorfman in 1943 during world war II to test US military personnel for Syphilis [1]. Dorfman style of testing is an adaptive group testing method. Since then, multiple researchers have analyzed various theoretical aspects pertaining to group testing, including intelligent ways to design the test matrix, pool size, lower bound (also called the converse bound) on the number of tests required below which no error-free decoding is possible and achievability bound (also called upper bound or the

[1] The COMP algorithm has been discussed in the literature under different names. For example, the authors in [6] analyze the COMP algorithm, but under different names: Column Matching (CoMa) and Combinatorial Basis Pursuit (CBP). The authors use a Bernoulli test design with CoMa and a near-constant row-weight design with the CBP algorithm. However, CoMa and CBP are mathematically equivalent. As stated by the authors, the naming convention in that work is motivated by the compressed sensing literature. Further, separate names help to couple the test design with the decoding procedure and the performance analysis. However, we follow the definition in [7], where the decoding procedure dictates the algorithm's name, and the test design is decoupled from the decoding procedure.

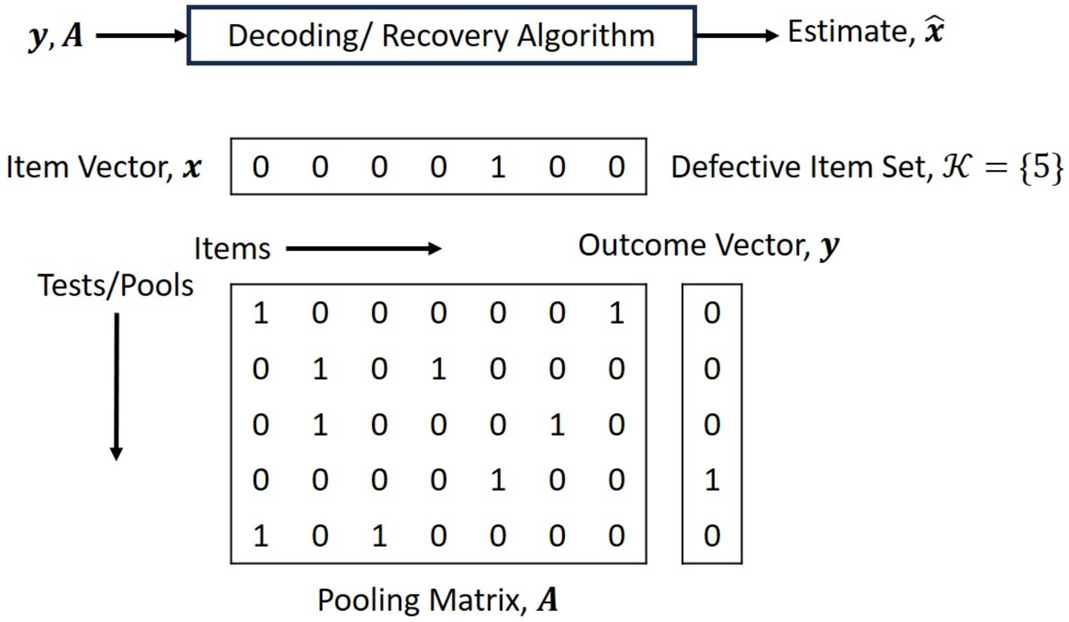
sufficiency bound) under various scenarios. An interested reader is referred to the recent extensive survey by Aldridge et al. [7]. A discussion on the theoretical results that are relevant to the thesis is presented in Section 2.1.1 and Section 2.1.2.

1.2.1 Definitions

As mentioned above, group testing can be categorized into adaptive and non-adaptive protocols. In this section, we summarize the commonly used terminologies related to non-adaptive group testing, which is the focus of this thesis.

1. *Group Test:* A *test* which is conducted on a pool or group of items. Here, a *test* can mean chemical analysis, electrical procedure, mechanical operation, or a series of digital operations, to name a few, which helps detect the presence or absence of specific characteristics of interest. Often, *group testing* refers to the act of performing one or more group tests according to a given pooling protocol. Group testing could also refer to a *decoding algorithm*, which is a way of mapping the outcome of several group tests to the respective items/individuals (see below for the definition). The specific meaning will be clear from the context.

2. *Pooling Matrix:* A binary matrix of size $m \times n$, where n denotes the number of items in the population and $m (< n)$ denotes the number of group tests. The (i, j)th element of the matrix takes a value of 1 or 0, and it indicates whether the jth item participates in the ith group test or not, respectively. The pooling matrix is also referred to as the test matrix. The term test design refers to the way the pooling matrix is constructed. In this thesis, we focus on random pooling designs, where the entries of the pooling matrix are chosen according to a probability distribution.


$\boldsymbol{y}, \boldsymbol{A}$
Decoding/ Recovery Algorithm
Estimate, $\widehat{\boldsymbol{x}}$
Item Vector, $\boldsymbol{x}$
Defective Item Set, $\mathcal{K} = \{5\}$
Items
Outcome Vector, $\boldsymbol{y}$
Tests/Pools
Pooling Matrix, $\boldsymbol{A}$

$$\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Figure 1.1: A toy example illustrating the idea of non-adaptive group testing.

3. *Item Vector:* A binary vector of size $n \times 1$ whose jth element takes the value 1 or 0, and indicates the defective or non-defective status of that item, respectively. Equivalently, the information in the item vector can be represented by a *defective item set* containing the indices of defective items.

4. *(Test) Outcome Vector:* A binary vector of size $m \times 1$ whose ith element indicates the outcome of the ith group test, and takes the value 1 or 0 depending on whether the outcome is positive or negative, respectively.

5. *Decoding Algorithm:* A procedure that takes m group test outcomes along with the pooling matrix as the inputs and outputs an estimate of the defective item set or an estimate of the items' statuses.

The Figure 1.1 illustrates these concepts with a toy example containing $n = 7$ items and $m = 5$ group tests. The item vector indicates that the 5th item is defective. In Figure 1.1, the pooling matrix is binary-valued. For example, the 3rd row indicates that the 2nd and 4th items participate in the 3rd group test. The outcome vector has a single positive group test (4th test). Finally, the decoding or recovery algorithm is shown as a block at the top, which takes the outcome vector and the pooling matrix as an input and outputs an estimate of the item vector.

1.3 Applications of Group Testing

A wide-variety of applications benefit from group testing. We discuss some of these in this section.

1.3.1 Random Access and Short Packet Communications

Some of the earliest works in [8–10] discuss the relationship between the group testing and the problem of random multiple-access communications. In [11], the authors show that a group testing-based protocol is superior to time-division multiple access (TDMA) for any user-packet-generation probability, p, provided there is at least one user in the so-called high power group. Thereafter, the group testing algorithm was extended to accommodate a heterogeneous set of users with unequal user activity probabilities [12]. Further, the authors in [13] extended the *monotonicity theorem* of group testing to a case where the group test outcomes can take one of the K values and apply the results to the collision resolution problem in a random multiple-access communication channel. In short, the monotonicity theorem states that the expected number of group tests under an optimal algorithm is non-decreasing in a parameter, p, which denotes the probability that an item, independently, is defective.

The work in [14] establishes an equivalence between solving a random access problem using an irregular repeated slotted Advocates of Linux Open-source Hawaii Association (ALOHA) and coded slotted ALOHA with the non-adaptive group testing problem. The ID of the transmitting device is implicitly stored in the transmission patterns of the data packets, thereby drawing an analogy with how a pooling matrix is constructed in non-adaptive group testing. For a population size of n and with at most k devices active at any given time, the protocol sequence length scales as $\Theta(k \log n)$. A sequential screening algorithm for activity detection in a massive random access channel with access delay constraint is developed in [15]. In [16], the same authors propose a group testing-based user activity detection algorithm under access delay and energy constraints such as those seen in fast-fading random access channels. The energy constraints can be translated to the *maximum number of tests each item can participate in* [17]. Also, non-adaptive group testing has been used to detect, localize, and track multiple individuals on a fiber sensor floor web [18].

Recently, the COMP algorithm has been applied by the authors in [19] to resolve packet collisions, thereby improving the success rate of the iterative interference cancellation process in the context of coded slotted ALOHA protocol. Also, the authors in [20] show that it is possible for a base station (BS) in a multiple-user multiple-input-multiple-output (MU-MIMO) setup to jointly identify and decode $\leq K$ (out of N) non-cooperative self-scheduling devices simultaneously aided by a group testing-based algorithm with vanishing probability of error using $O(K \log(NM))$ antennas at BS, where M denotes the number of possible messages.

1.3.2 Cognitive Radios

The authors in [21] apply group testing algorithms to find a spectrum hole of a specified bandwidth in a wideband channel. The sampling strategy exploits the sparsity in the frequency-domain. In particular, the primary user's spectral occupancy is sparse. The authors in [22] apply group testing to detect a group of active sub-carriers in a wavelet-packet modulation-based multi-carrier channel. The authors show that the computational complexity of the detection algorithm can be reduced by group testing compared to classical spectrum sensing techniques. A two-stage algorithm for wideband spectrum sensing using poly-phase filter banks is developed in [23].

1.3.3 Sensor Networks

The authors in [24] propose a group testing-based reactive trigger node identification algorithm for defending from reactive jamming attacks in wireless sensor networks (WSN) across different *attack models*. The authors in [25] extend the group testing algorithms to a distributed setting and propose a solution to detect defective sensors in a sensor network. Combinatorial group testing has been applied along with Kalman filtering to detect faulty sensors in a network when the occurrence of the faults is relatively rare [26].

A variation of group testing called *generalized group testing* has been applied to detect status of the sensors and retrieve information in a WSN efficiently, i.e., using a minimal number of channel accesses [27, 28] and to derive scheduling policies for sensors' adopting cooperative transmissions under a correlated source model [29]. The authors in [30] apply group testing with a Latin Square-based pooling matrix to detect invalid signatures in a WSN.

1.3.4 Electrical and Electronic Device/Product Testing

One of the earliest work by Sobel and Groll discuss an application of group testing in product testing [31]. After that, the authors in [32] have applied group testing to detect and locate shorts in electric circuits or groups of nets. More recently, group testing has been applied to built-in self-test (BIST) diagnosis of digital logic systems to detect faulty scan cells, faulty logic blocks, erroneous test vectors, etc., by the authors in [33]. Due to the parallelism, the test results can be obtained faster in non-adaptive group testing. The authors in [34] use this fact to improve the throughput of a genetic algorithm to perform autonomous repair and refurbishment of re-programmable logic devices.

1.3.5 Compute, Storage and Distributed Systems

In distributed systems, the files are replicated and stored across multiple physical devices to meet the reliability and availability requirements. Under a system model where it is guaranteed that $\leq k$ pages or partitions out of n partitions of a file can differ across the copies, the authors in [35] apply group testing algorithms to efficiently detect the pages/partitions which are different. A suitable checksum is assumed to be available per page (or partition, as applicable) for the comparison.

The authors in [36] use pooling matrix designs from group testing to detect collusion attacks where a malicious or illegal distributor of watermarked digital content attempts to delete or modify the embedded fingerprint used to decode the primary buyer. Tracing algorithms based on group testing where a set of buyers/users are pooled and tested for collusion attack is discussed in [37–39].

The authors in [40, 41] apply group testing to solve the problem of verifying web services (WS) in a dynamic service-oriented architecture (SOA) where new services can be composed at run-time using existing WS. The verification checks whether the WS behavior is normal, and knowing the WS's state helps establish trust in the users [42].

To realize efficient data transmissions in an authenticated manner in the context of the Internet of Things (IoT), the authors in [43, 44] propose a group testing-based method for message authentication using an aggregate message authentication code (MAC). Further, the authors in [45] and [46] apply group testing to detect application denial-of-service (DoS) attacks. Group testing algorithms have been applied to detect malicious users [47]; to detect anomalous patterns like attacks, hot IPs, and the status of hosts and services in distributed systems [48, 49].

The authors in [50] propose a variant called sum-observation group testing, where the group test outcomes contain the count of defective items that have participated in the test for detecting heavy hitters in communication networks. A hierarchical group testing-based solution for the byzantine attack, i.e., where the worker node in a distributed computing pipeline sends an incorrect result to the central master, is addressed by the authors in [51]. Further, a failed link detection in a network using group testing has been proposed along with the theoretical bounds in [52, 53]. Similarly, fault detection in an all-optical network using non-adaptive combinatorial group testing has been proposed in [54]. The authors in [55] apply group testing to accelerate the rational unified process (RUP), where testing is performed throughout the software (product) development process.

The authors in [56] note that the non-uniformity in the photo response noise associated with an imaging sensor or a camera can be used to identify the specific camera used for capturing an image. However, storing the fingerprints of possibly thousands of camera devices available in a database is both memory and time-consuming while matching and retrieving, rendering the idea impractical. The authors propose to use group testing with carefully constructed composite fingerprints to overcome this bottleneck. Recently, the authors in [57] present and analyze a distributed group testing algorithm applied to a network intrusion detection problem, where the goal is to identify unauthorized or malicious activities within a computer network.

1.3.6 Infectious Disease Detection and Medical Screening

After the success of group testing to screen Syphilis by Dorfman [1], the same idea has been applied successfully to screen various infectious diseases. In particular, the authors in [58] apply a pooled testing strategy to detect Malaria using a polymerase chain reaction (PCR) assay. Other applications include detecting viral infections like acute human immunodeficiency virus (HIV), hepatitis B and C, and West Nile virus, to name a few (see the references in [58]).

Recently, group testing has regained popularity owing to the Covid-19 pandemic [59, 60]. The authors in [61] concluded that the detection limit is $1 - 3$ Ribonucleic acid (RNA) copies per μl of the sample. Covid-19 detection using a multi-stage group testing has been proposed by [60, 62], where the individual samples are classified as having *{no, low, medium, high}* infection level. The authors in [63] apply non-adaptive group testing algorithms, whereas the authors in [64, 65] apply compressed sensing (CS) algorithms for Covid-19 detection. Further, group testing has been used to improve test reliability and throughput via possibly non-binary pooling matrices [66]. We discuss more such works from the literature about the application of group testing for Covid-19 detection in Section 4.1.3.

Another exciting area that benefits from group testing is expanding a cancer staging system called tumor lymph node metastasis, where groups of cancer patients' records are formed. Then, statistical testing is applied to detect the difference among the groups [67].

On similar lines, group testing has been helpful in drug and Deoxyribonucleic acid (DNA) library scanning [68–71]. Lastly, it is possibly applicable to monkeypox detection [72] and to scale-up testing to detect Khosta 2, a type of sarbecovirus [73].

1.3.7 Other Applications of Group Testing

In addition to the above-mentioned areas, group testing has found a use in improving the latency of single-touch detection on a capacitive touch screen [74]; to enable an efficient continual face authentication on the cloud [75]; in wavelet-based image compression [76,77] and video compression [78]. Further, group testing has increased the storage density of topological DNA-based data storage [79, 80]. Similarly, group testing has been applied to DNA microarrays to perform a large number of DNA-DNA hybridization experiments, where oligonucleotide probes could bind to multiple targets simultaneously [81]. This approach overcomes the need for designing sequence-specific probes, which has become a significant challenge due to the presence of closely related target families in the sample. Further, combinatorial group testing has also been applied to check the integrity of the hard disks by hashing [82]. The proposed method improves the storage space required to store the hash values of the sectors as compared to classical solutions.

Another exciting area of research where group testing has been found useful is in identifying the k most connected vertices (kMCV) in hidden bipartite graphs and applying the results in various applications like spatial databases, graph databases, and bioinformatics. In a hidden bipartite graph, some edges are not explicitly given, and edge probe tests are required to detect the presence of edges. A single invocation of a group test can reveal as much information as multiple invocations of 2-vertex tests [83].

The authors in [84] have applied group testing to the problem of beam alignment (BA) in an uplink multi-path communication between user equipment (UE) and BS. They show that a group testing-based algorithm for BA improves (minimizes) the expected BA duration in both noiseless and 5G millimeter wave (mmWave) settings.

Large-scale image classification tasks where the number of object categories runs into millions, whereas the number of categories (of the objects) that appear in a given image is low ($\sim 10 - 50$) can be made more efficient by reducing the number of *one-vs.-rest* classifiers required, using group testing [85]. The authors in [86] use group testing in neural network inference to test (infer) multiple images in batches or groups to determine rare but inappropriate images, for example, in an image moderation task. Recently, neural group testing has been applied to wireless/sensor networks in [87].

1.4 Group Testing: Benefits and Challenges

The discussion in the above section shows that group testing has been found useful in a wide range of applications. When the number of defective items in a population is low as compared to the population size, and further, when there is a constraint on time-to-test/ resources, group testing has been known to be useful in improving the throughput of the testing procedure with optimal use of available resources and man-power [58, 88].

In addition to the throughput improvement, group testing applied to product testing can improve the reliability of the verification process [31]. Under an assumption that the unsampled and, hence, untested products display similar statistical properties and, further in the interest of time-to-test, products from an industrial manufacturing line are sampled and subjected to quality checks. If the products can be tested in pools or groups, then group testing enables higher coverage and reliability of the overall quality assurance process. In the early stages of a pandemic or breakdown of a contagious infection in the population, group testing can support in realizing a robust population monitoring process and help arrest the spread of the disease early on [88].

Owing to multiple advantages offered in a wide range of applications where group testing has been applied successfully, understanding the theoretical implications and results that characterize the group testing protocols and the decoding algorithms has been of critical interest. In adaptive group testing, work on understanding how to decide the optimal pool size and number of stages in which the group test is conducted can be found in [89, 90]. In the non-adaptive variant, designing an optimal pooling matrix such that it guarantees recovery of a certain number of defective items, guarantees recovery under constraints on the number of tests that an item can participate and the number of items a test could accommodate has been studied [7]. In addition, a lower bound (also called converse bound) on the number of tests required below which no group testing algorithm can guarantee a vanishing probability of decoding error has been studied [4, 7, 90, 91]. Lastly, the expected number of tests and asymptotic results for the achievability or sufficiency bounds on the number of tests for different decoding algorithms have been studied in the literature [4, 6, 90, 92].

In practice, the number of items to be tested is finite. Therefore, one is often interested in non-asymptotic analysis-based bounds. Non-asymptotic bounds are presented in the literature for a maximum-likelihood-based algorithm, which is computationally intractable even when the problem size is moderate [93, 94]. Furthermore, in practical settings with finite resources, one is often interested in ensuring that the probability of the error incurred by the decoding algorithm remains below a small number, which is called confidence. Also, one could be interested in the approximate recovery of the defective items where a certain amount of decoding errors are tolerated [92]. As far as we know, there is no work in the literature where sufficiency bounds for practical group testing algorithms like COMP and DD are presented under both exact and approximate recovery scenarios parameterized by a confidence parameter in a single framework. We bridge this gap by viewing group testing as a function learning problem, which enables us to apply probably approximately correct (PAC) analysis to derive sufficiency bounds. Our bounds enable us to quantify the sensitivity of the group testing algorithms to both confidence level and error tolerance separately.

On the other hand, group testing algorithms for Covid-19 detection and similar applications where the system model exhibits non-linearity and the feasibility set of the decoder's optimization problem includes both finite values and infinities have been of interest. We develop Covid-19 detection algorithms for the case when the efficiency parameter of the PCR test is unknown. In addition, from the practitioner's point of view, it is interesting to understand when group testing is useful, namely, how does one read the theoretical results and use them in practice?

The above aspects form the core of this thesis. We next introduce, without delving into mathematical details, the research problems explored in this thesis and our original contributions regarding the methodologies and algorithms developed to address them. More elaborate discussions are relegated to the dedicated chapters.

1.5 Scope and Contributions of the Thesis

This thesis focuses on two central themes: 1) theoretical analysis of the group testing algorithms where we derive sufficiency bounds for well-known practical group testing algorithms under both exact and approximate recovery conditions and also parameterized by a desired confidence level and 2) iterative decoding algorithms for recovering a set of Covid-19 infected individuals and estimating *the state of the infection* under a non-linear system model and unknown PCR efficiency factor. In the context of Covid-19 detection using reverse Transcriptase quantitative PCR (RT-qPCR), the state of the infection is quantified by a parameter called the *cycle threshold* (CT). That is, the decoding algorithm must not only find the set of infected individuals, it also needs to estimate the CT values of each infected individual.

In the next two subsections, we briefly overview each of these and highlight our main contributions.

1.5.1 PAC Analysis for Group Testing Algorithms

In Chapter 2, we reformulate the group testing problem from a function learning view. We establish an equivalence between the average error probability in the exact recovery case and the prediction error by the learned function. Then, in Chapter 3, we relate the false positive (FP) and false negative (FN) errors to the approximation error of the function learning problem. Further, we use the PAC framework to derive sufficiency bounds for well-known practical group testing algorithms. We summarize the main contributions leading to novel sufficiency bounds for the group testing algorithms using PAC analysis below.

1. We reformulate the defective set identification problem in non-adaptive group testing with random pooling as a function learning problem. We relate the notion of

recovery error in group testing with the prediction error in the function learning problem. Specifically, we show that the exact recovery of the defective set by a decoding algorithm is equivalent to prediction with zero error in the function learning problem. Further, we relate the number of false positives (similarly, the number of false negatives) to the approximation error probability used in the PAC formulation. This establishes a connection between the parameters used in the PAC formulation and the conventional metrics of interest in group testing.

2. We apply the PAC analysis to derive a sufficiency bound on the number of tests in exact and approximate recovery conditions for the binary group testing algorithms: COMP and DD. The COMP algorithm is analyzed under a Bernoulli test design and a near-constant row-weight design, whereas DD is analyzed under the Bernoulli design. We use the terms COMP-B and COMP-R to distinguish between the analysis of the COMP algorithm with Bernoulli design and the COMP algorithm with near-constant row-weight design, respectively. The PAC analysis serves as a common framework, where exact recovery results in the literature emerge as a special case. We extend the existing results on the coupon collector problem to handle *collection of only a subset of coupons* and apply them to derive the COMP-R bound. Further, we optimize the Chernoff design parameter to get a tighter bound for COMP-R. Finally, we derive the order-wise behavior of the PAC bounds for all the three cases which leads to easily-interpretable and insightful sufficiency bounds.

3. The PAC-based analysis allows us to trade off the *accuracy* of defective set recovery with the *confidence* with which the decoded set meets that accuracy requirement. We present a visualization of this trade-off in a sufficient number of tests *contour/surface*,

which shows its dependence on the approximation error tolerance and the probability of failure to meet the required error tolerance. Also, PAC analysis enables one to characterize a lower bound on the cumulative distribution of the approximation errors.

Numerical results show the agreement between the theoretical bounds and the simulated values. Using the PAC framework for analyzing practical group testing algorithms opens up means to accommodate both exact and approximate recovery and further account for the randomness in the test matrix, in a common framework. Finally, as mentioned earlier, PAC bounds enable us to study the sensitivity of the group testing algorithms to both confidence level and error tolerance, separately.

1.5.2 Recovery Algorithms for Covid-19 Group Testing

As mentioned earlier, group testing has regained practical importance due to the Covid-19 pandemic. Owing to the non-linear nature of the PCR system model and stability issues encountered by the optimization algorithms due to the presence of infinities in the feasible set, the recovery performance of the existing group testing algorithms for Covid-19 detection is low. We develop a set of novel iterative recovery algorithms to overcome this bottleneck in Chapter 4. In particular, these recovery algorithms estimate the individual sample CT values. Then, in Chapter 5, we develop robust recovery algorithms to estimate the individual sample CT values when the PCR efficiency factor is unknown. Now, we summarize our main contributions in that part of this thesis.

1. We develop two novel algorithms to accurately recover the individual sample CT vector given the pooled CT vector and the pooling matrix, from the non-linear

RT-qPCR model. Further, to address the unknown PCR efficiency factor issue, we develop two novel algorithms that can jointly recover the individual sample CT vector and estimate the unknown efficiency factor. The developed recovery algorithms are robust to noise and varying machine parameters.

2. A case study describing the advantage of using quantitative measurements in non-adaptive pool testing in terms of the testing rate and, hence, the cost, is presented using publicly available data on the number of tests conducted. The best rates achievable for a given prevalence rate using deterministic testing matrix designs are compiled. In practice, the prevalence rate estimate obtained from the previous day can be used to decide the optimum (testing) rate design for each day.

The performance of the proposed algorithms is evaluated using a simulated model of the RT-qPCR process for Covid-19 to conclude that they outperform related algorithms in the literature under practical settings (e.g., unknown machine-specific parameters, CT measurement noise, etc.) in terms of the normalized mean-squared error (NMSE) and the sparsity level up to which the algorithms guarantee zero recovery errors.

1.6 List of Publications

Journal Articles

1. S. Bharadwaja H. and C. R. Murthy, "Recovery Algorithms for Pooled RT-qPCR Based Covid-19 Screening," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4353-4368, Aug. 2022.

2. S. Bharadwaja H. and C. R. Murthy, "Probably Approximate Correction Analysis

for Group Testing," *Longer version of ISIT 2022 paper. Under review at IEEE Transactions on Information Theory. Submitted in June 2024.*

Conference Proceedings

1. S. Bharadwaja H., M. Bansal and C. R. Murthy, "Approximate Set Identification: PAC Analysis for Group Testing," *IEEE International Symposium on Information Theory (ISIT)*, Espoo, Finland, 2022, pp. 2237-2242.

Publications not included in this Thesis

1. Thoota, S. S., Marti, D. G., Demir, Ö. T., Mundlamuri, R., Palacios, J., Yetis, C. M., Thomas, C. K., Bharadwaja, S. H., Björnson, E., Giselsson, P. and Kountouris, M., Murthy C. R., González-Prelcic N., Widmer J., "Site-Specific Millimeter-Wave Compressive Channel Estimation Algorithms with Hybrid MIMO Architectures," *ITU Journal on Future and Evolving Technologies (ITU J-FET) special issue on AI & ML solutions in 5G and future networks*, vol. 2, issue. 4, pp. 9-26, July 2021.

2. S. Katyal, S. Bharadwaja H., C. R. Murthy, "Deep Unfolding-Based Channel Estimation and Soft Symbol Decoding With Low-Resolution ADCs," *European Signal Processing Conference (EUSIPCO)*, Lyon, France, August 2024.

2 Probably Approximately Correct Formulation for Group Testing

Chapter Highlights

This chapter sets the stage for the PAC analysis of non-adaptive group testing algorithms. We start by formally describing the non-adaptive group testing problem. The notion of *exact recovery* of defective items and the average probability of error under a combinatorial setting is then defined. A summary of various results on sufficiency bounds for the number of tests is presented. The concept of *approximate recovery* and its applications are stated, followed by how various works in the literature have derived the sufficiency bound to accommodate this relaxation in the recovery condition. The gaps in the existing analysis methods are presented to motivate a novel reformulation of the group testing problem.

We present the PAC learning view of the group testing problem. In particular, the group test outcomes are viewed as an output of a k-literal OR-ing function. The group testing algorithm aims to learn a k-literal OR-ing function that satisfies a certain notion of *correctness* at a certain *confidence* level. PAC analysis enables us to determine a sufficient number of tests for a target level of correctness and confidence. The bounds derived using our PAC learning formulation enable the characterization of the sufficient number of tests as a function of two parameters in addition to the number of items, n and the number of defectives, k: 1) the approximation error tolerance, ϵ, and 2) the confidence, $1 - \delta$. This chapter presents a Lemma that relates PAC learning to group testing in the exact recovery case. This chapter forms the foundation for the next chapter, where we present how the approximation error tolerance manifests in terms of false positives and false negatives depending on the group testing algorithm under consideration, followed by a derivation of corresponding sufficiency bounds on the number of tests.

2.1 Introduction

We describe the system model for random pooling-based non-adaptive group testing [21]. Here, the decision about which items will participate in which group test is predetermined and is encoded in a random binary *test matrix* denoted by $\mathbf{A} \in \{0,1\}^{m \times n}$ [7]. The (i, j)th element of $\mathbf{A}$, denoted by a_{ij}, takes the value 1 or 0 depending on whether the jth item participates in the ith group test or not, respectively. The *item vector* is denoted by $\mathbf{x} \in \{0,1\}^n$, whose jth entry, denoted by x_j, takes the value 1 if the jth item is defective, and 0 otherwise. In addition, the support of $\mathbf{x}$ is denoted by the set $\mathcal{K}$ such that $|\mathcal{K}| = k$, where $|\cdot|$ denotes the cardinality of a set. Finally, the *outcome* of the ith group test is $y_i = \vee_{j=1}^{n} a_{ij} x_j,\ i \in [m]$, where $\vee$ denotes the Boolean OR-ing operation. Thus, the outcome is 1 if the group test includes one or more defective items and is 0 otherwise.

Once the group test outcomes $(y_i, i \in [m])$ are collected, a decoding algorithm aims to output an item vector estimate denoted by $\hat{\mathbf{x}} \in \{0,1\}^n$. The estimate of the defective item set, $\hat{\mathcal{K}}$ is then simply the support of $\hat{\mathbf{x}}$, i.e., the set of indices corresponding to the nonzero entries in $\hat{\mathbf{x}}$.

Now, when *exact set identification* is considered, the decoding algorithm succeeds when $\hat{\mathcal{K}} = \mathcal{K}$ and fails otherwise. Let $\mathbb{P}_{\mathbf{A}}\left(\hat{\mathcal{K}} \neq \mathcal{K}\right)$ denote the probability of error given the set of defectives, $\mathcal{K}$. Then, the average probability of error under a combinatorial setting can be written as [7]

$$\mathbb{P}(\text{err}) = \mathbb{P}_{\mathbf{A},\mathcal{K}}(\hat{\mathcal{K}} \neq \mathcal{K}) = \frac{1}{\binom{n}{k}} \sum_{\mathcal{K}:|\mathcal{K}|=k} \mathbb{P}_{\mathbf{A}}(\hat{\mathcal{K}} \neq \mathcal{K}). \tag{2.1}$$

Characterizing $\mathbb{P}(\text{err})$ as a function of k, n, and m is helpful in using group testing for practical applications. For instance, one can ask: *Given k and n along with an upper* *bound on $\mathbb{P}(err)$, what is the sufficient number of tests, m_S, required by a group testing algorithm?* In the following subsection, we discuss the existing work toward answering this question before presenting our main contributions for this chapter.

2.1.1 Prior Work

Under $\delta' \equiv \mathbb{P}(\text{err}) = n^{-\delta}, \delta > 0$ and $k = o(n)$, the authors in [5, 6] show that an upper bound on the number of tests is $ek(1+\delta)\log n$ and $2ek(1+\delta)\log n$ for COMP algorithm with Bernoulli design, i.e., COMP-B (also called CoMa in [6]) and COMP algorithm with a near-constant row-weight design, i.e., COMP-R (also called CBP in [5]), respectively.[1] Further, the LiPo decoder requires no more than $O(k \log n)$ tests under the same conditions with the constant factor associated with the asymptotic expression being a function of $(1{+}1/k)$, $\log k/\log n$ and δ [6]. Similarly, the sufficient number of tests for DD and SCOMP algorithms is also $C_a k \log n$, with a decoder-dependent constant $C_a > 0$ [4]. In addition, a lower bound on the required number of tests is $(1-\delta')k\log(n/k)$ (See [6] and references therein, e.g., [95, 96].). In the above works, the entries of the test matrix are chosen i.i.d. from a Bernoulli distribution with parameter $p \in (0,1)$, denoted by $\mathcal{B}(p)$. Further, the author in [97] presents an improved converse bound for this Bernoulli-design group testing.

Other test matrix designs have been considered and contrasted with the Bernoulli design. The authors in [98] draw inspiration from spatially-coupled low-density parity check (LDPC) codes for designing their test matrix. For any $0<\beta<1$, $k \sim n^{\beta}$, $\epsilon' > 0$, the authors derive lower and upper bounds on the number of tests as $(1{-}\epsilon')m_{\text{inf}}$ and $(1{+}\epsilon')m_{\text{inf}}$, respectively, where $m_{\text{inf}} = \max\{\beta/\log^2 2, (1-\beta)/\log 2\}k\log n$. The authors in [99] show

[1] Throughout the thesis, log denotes the natural logarithm, unless specified otherwise.

similar results for the number of tests below which any group testing algorithm fails and above which the SCOMP and/or DD algorithms succeed.

The authors in [100] provide bounds on the number of tests for disjunct test matrices. In [94, 101], the authors show that a (near-)constant tests-per-item design with $m(\log 2)/k$ tests-per-item (chosen with replacement) requires 23.4% fewer tests (correspondingly, the testing rate improves by $\approx 30\%$) than the Bernoulli design, when coupled with COMP or DD algorithms. The authors in [102] show that, in the sub-linear sparsity regime (i.e., $k = \Theta(n^\beta)$, $\beta \in (0,1)$), the DD algorithm under a constrained design with at most a fixed number of tests-per-item and $\rho = O\left((n/k)^\beta\right)$, $\beta \in [0,1)$ items-per-test, yields an improved achievability result compared to the COMP algorithm under an unconstrained design. The work in [103] presents an analysis of constrained design (e.g., constraints on the number of tests per item or the *item-divisibility* and on the number of items per test) and shows that even a small *amount of constraint* can have a significant effect on the information-theoretic bound. Recently, the authors in [104] considered both a constant column-weight design (termed Δ-divisible item setting) and a constant row-weight design (termed Γ-sized test setting) while analyzing the achievable number of tests of various decoding algorithms. Under the $k \sim n^\beta$ regime and $\Gamma = \Theta(1)$ assumption, the DD algorithm succeeds w.h.p. for $m \geq \max\{2, 1+\lfloor \beta/(1-\beta) \rfloor\}\, n/\Gamma$ [104, Theorem 4.10]. In contrast to the sub-linear regime, the works in [95, 105] consider very sparse regime, where $k = O(1)$ and the works in [106–108] consider a linear regime, i.e., $k = pn$ for $p \in (0,1)$.

The work by the authors in [109, 110] considers a case when there are inhibitors, pretenders, or unreliable items in the population. In contrast to the (unconstrained) non-adaptive protocol considered here, adaptive techniques (see [7, 90, 111–114] and references therein) and group testing models, where block length, run-length, geometric and graph constraints, community-awareness, correlation (as opposed to items being defective independently) have been considered for theoretical analysis in the literature (see [112,114–119] and references therein). Also, noisy group testing has been analyzed under different test designs (see [111, 120, 121] and references therein).

Converse results for both adaptive and non-adaptive testing algorithms using finite blocklength results and directed information theory formulations are presented in [122]. Furthermore, results in the literature can be categorized based on what error metric is used and how it is characterized. In a probabilistic group testing (PGT), one is interested in determining the bounds on the number of tests such that $\mathbb{P}(\text{err}) \to 0$ as $n \to \infty$, whereas in the combinatorial group testing (CGT), the criterion changes to $\mathbb{P}(\text{err}) = 0$ for any n [123]. Other metrics of interest include number of false positives and false negatives, to name a few. More discussions on this are deferred to Section 2.1.2.

The authors in [93] present a sufficiency condition on m for a weakened version of the maximum likelihood decoder in the non-asymptotic regime, i.e., when the problem dimensions are finite. The non-asymptotic bounds are more practically useful: for instance, in a typical RT-qPCR based Covid-19 testing kit, one can accommodate ~ 96 or 384 (pooled) samples.[2] However, the analysis of the maximum likelihood-based decoder is dependent on the decoder having the knowledge of k and is typically computationally intractable even for moderate-sized problems [7, 94]. Therefore, practical and computationally simple group testing algorithms like COMP and DD are attractive.

[2] PCR Plastics - PCR Plate Configurations: https://www.thermofisher.com/in/en/home/life-science/cloning/cloning-learning-center/invitrogen-school-of-molecular-biology/pcr-education/pcr-qpcr-plastics/pcr-qpcr-plastics-considerations.html

2.1.2 Approximate Defective Set Identification

In many group testing applications, *approximate defective set* recovery is sufficient [7]. Here, the estimated defective set can contain missed defective items or false positive items. As an example, consider the scenario where one needs to quickly identify a few people with a particular antigen in their blood when the incidence of the antigen in the population is low. We need not identify all the defective items (i.e., all the people with the antigen in their blood), a subset is sufficient. In such a scenario, false negatives can be tolerated. Alternatively, consider the problem of shipping out a small set of non-defective industrial products in urgency [31]. In this case, the goal is to ensure fast scrutiny of which items are non-defective and ship them out quickly. It is sufficient to detect the required number of non-defective items to be shipped, which may be only a fraction of the full non-defective set. In order words, false positives can be tolerated.

In such scenarios, one could reduce the sufficient number of tests by exploiting the relaxation on the required accuracy of the recovered defective/non-defective set. In this direction, the authors in [124] present an achievability bound on the number of tests when up to αk misses out of k defectives are allowed given n items, as $O\left(e^{(1-\alpha)k\log n}/H(e^{-\alpha})\right)$, where $H(\cdot)$ denotes the binary entropy. The authors in [93, 125] consider approximate defect set identification and derive both achievability and converse bounds on m in the *sub-linear regime*, i.e., $k = \Theta(n^{\beta})$, $\beta \in (0,1)$. The proof uses information spectrum methods and thresholding techniques from channel coding theory. For achievability, the authors use a maximum likelihood-like recovery algorithm. Similarly, the authors in [92] consider a list-decoding algorithm for approximate recovery. The results in all these scenarios show that allowing $\lfloor \alpha k \rfloor$ defectives to be missed relaxes the converse bound on m by at most a multiplicative factor $1-\alpha$, where $\alpha \in (0,1)$.

The authors in [126] show that the probability that the maximum of the false positive (FP) or false negative (FN) errors incurred is no more than $\alpha k, \alpha \in (0,1)$ approaches one when $m \geq (1+\alpha')(1-\alpha)\log_2\binom{n}{k}$ but the same probability approaches zero when $m < (1-\alpha')\log_2\binom{n}{k}$, for arbitrary $\alpha' > 0$, as $n \to \infty$. A treatment of this *all-or-nothing* phenomenon is also presented in [127]. The authors in [128] use sparse graph codes and present a decoder called SAFFRON which can recover $(1-\epsilon')k$ defectives with probability $1-k/n^r$ with $2(1+r)C'k\log_2 n$ tests, where C' is a function of $\epsilon' > 0$ and $r \in \mathbb{Z}_+$.

The available bounds on the sufficient number of tests for approximate defective set recovery also depend on the underlying test matrix design, i.e., the distribution from which the test matrix is drawn. For example, in the sub-linear regime, with an i.i.d. Bernoulli$(\log 2/k)$ test matrix, $k\log(n/k)/\log^2 2$ tests are sufficient for the error probability (appropriately defined to account for the maximum number of FPs and FNs allowed) to approach zero [129]. Similarly, under a doubly-regular design with column-weight r and row-weight s, it is known that the false negative rate (FNR) is minimized for larger r and smaller s, whereas the number of tests is rn/s [130]. Therefore, they fix different (nonzero) values for the allowed FNR and numerically analyze the number of tests. They define the false positive rate (FPR) $= \mathbb{E}[|\hat{\mathcal{K}} \setminus \mathcal{K}|]/|[n] \setminus \mathcal{K}|$ and FNR $= \mathbb{E}[|\mathcal{K} \setminus \hat{\mathcal{K}}|]/|\mathcal{K}|$, where $[n] \triangleq \{1,\ldots,n\}$ as a measure of the average approximation errors.

Recently, the authors in [131] addressed a question related to the *computational-statistical gap* (CSG) in the non-adaptive group testing paradigm, where one is interested in $(1-o(1))$−approximate recovery in the $k = o(n)$ regime, for the $\mathcal{B}(\log 2/k)$ test design [129, Sec. IIA]. The CSG is defined as the gap between the number of tests above which recovery is information-theoretically possible and the number of tests required by the currently best-known efficient algorithms to succeed [131]. The authors provide evidence that the gap can be closed when $m \geq (1+\alpha)k\log_2(n/k)$, $\alpha \in (0,1)$ and $n, k \to \infty$, enabling $(1-\alpha')m$ tests to contain at least $\lfloor(1-\alpha)k\rfloor$ defectives asymptotically almost surely, for any $\alpha' > 0$. The authors show that the absence of CSG implies that a practical local search routine succeeds in solving the smallest satisfying set (SSS) estimator under the said regime, which otherwise has a combinatorial complexity.

In contrast to the usual error probability or the FP/FN rates, the author in [132] considers a new metric called the expected number of tests per *infected* individual found (ETI). One of the questions posed in this work involves tolerating partial recovery and analysis of the SAFFRON[3] algorithm's ETI, which is shown to be $2e\log_2(n/k)$. Lastly, [7, 105, Section 5.1] presents bounds on m for COMP and DD under partial recovery conditions, albeit without proof. For instance, for $m \geq (1+\eta)ek\log(n/k)$, where $\eta > 0$, the average number of FPs output by the COMP algorithm behaves as $o(k)$, and, therefore, the probability of getting more than γk false positives, for a fixed $\gamma \in (0,1)$, tends to zero. A similar result is presented in the context of the DD algorithm for FN errors. The authors in [2] also briefly discuss partial defective set recovery in group testing.

In practical settings with finite resources, one is often interested in ensuring that the probability of the error incurred by a function learned using a finite number of randomly drawn samples exceeding a threshold remains below a small number, called *confidence*. We use the PAC formulation [133, 134] to bridge this gap.

We have presented some of the main results from the vast field of research on group

[3] SAFFRON: Serial estimate of the Alpha Fraction that is Futilely Rationed On true Null hypotheses.

testing relevant to this work. A collection of extensive results and deeper discussions pertaining to various decoder rules, testing protocols, and designs, including the bounds on the number of tests under different constraints, reliability criteria, various measurement and mixing models can be found in [7] and references therein. With this background, we now summarize the motivation for applying PAC analysis to group testing algorithms.

2.1.3 Motivation and Contributions

To the best of our knowledge, a rigorous treatment of sufficiency bounds on m for *approximate* recovery using practical algorithms like COMP and DD accounting for the *randomness* in the test matrix and at nonzero confidence levels is not available in the literature. In order to bridge this gap, we start by reformulating the group testing problem and view these algorithms through the lens of function learning and PAC analysis. In turn, this allows us to shed light on the relationship between PAC-learning and the exact recovery bounds available in the literature [4, 6]. A fundamental difference between PAC learning [133, 134] and our problem is as follows. In group testing, we can *choose* the data distribution from which the samples are drawn for function learning based on our knowledge of the hypothesis space from which the target function is to be learned (in the context of group testing, function learning corresponds to identifying defective items.) For example, in i.i.d. Bernoulli test matrix designs, we can choose the probability p with which a given item participates in a given test. As we will see, this additional degree of freedom allows us to optimize the PAC learning process and obtain tighter and more general bounds on the sufficient number of group tests.

It is worth mentioning that viewing group testing as a function learning problem has other potential applications: blind chemistry, where one is interested in determining which

k out of n reactants in a chemical reaction has the potential to create a particular (useful) detectable compound [2]; identification of key design variables for improving an automobile's fuel efficiency; key-species identification in a complex biological ecosystem; reactions of bacteria in gut micro-biome to a given drug, etc [2, 135]. Also, multi-label classification with a large number of the number of class labels $n \sim 10^3 - 10^6$ and a small number of classes per input (e.g., an image) is $k \ll n$ can be performed by a set of n *one-vs.-rest* classifiers. That is, the jth classifier is trained to output 1 if the jth object is detected in the input image, and 0, otherwise for each $j \in [n]$. The authors in [85] propose to use a (k, e)-disjunct matrix and encode the n-dimensional label vector using an m-dimensional vector, where $m \ll n$. The m-length binary vector is constructed by learning m one-vs.-rest classifiers. Then, the recovery of the original n-length label vector from the m-length binary vector is solved as a classical group testing problem. The test design in [85] can tolerate up to $\lfloor e/2 \rfloor$ misclassifications. On a similar note, the authors in [136] show how group testing can be used in a binary classification problem by posing it as an exact rule-learning problem.

We note that the PAC formulation is one of the key tools used to analyze machine learning algorithms [133]. For example, PAC analysis aids in developing a lower bound on the probability that the error rate of the above-mentioned multi-label classifier lies below a certain error threshold. However, the existing analysis of group testing algorithms does not conform to this notion of characterizing the distribution of the error rates over the randomness of the test matrix. By formulating group testing as a PAC learning problem and developing corresponding achievability bounds, we are able to bridge this gap. Furthermore, since the PAC formulation allows one to characterize the decoding error rate *and* the confidence in a unified framework, it is well suited for applications where the target error tolerance is small but nonzero, as in the examples discussed above.

In this chapter, we set the stage for PAC analysis of group testing algorithms. Then, in the next chapter, we use the PAC analysis to characterize the sufficient number of tests, m, required for *approximate* recovery of the defective set with *high-probability* using the PAC learning framework. Our main contributions in this chapter are as follows:

1. We reformulate the defective set identification problem in non-adaptive group testing with random pooling as a function learning problem. It can be seen that the non-adaptive group testing problem is equivalent to learning a k-literal OR-ing function, where k denotes the number of defective items in the population of n items.

2. A PAC model for group testing is established where we show that the average probability of error definition in the combinatorial group testing model setting is equivalent to that of the PAC bound given in (2.6) when the approximation tolerance probability, ϵ, is equal to 0.

The technical novelty of this part of the thesis is that it investigates the group testing problem from a new perspective, i.e., the PAC framework. Traditional PAC analysis methods require $\epsilon > 0$ and show that the sample complexity for PAC learnability varies as $1/\epsilon$ [133]. In the context of group testing, we obtain the exact recovery bounds by setting $\epsilon = 0$. Therefore, the proof techniques used in classical PAC analysis can not be directly applied to group testing. We first bridge this gap between the traditional PAC framework and the analysis of group testing algorithms. We show an equivalence between the performance characterization in PAC learning and group testing in the exact recovery scenario, through Lemma 2.1. As we shall see in Chapter 3, we relate the allowed number of FP errors, g_ϵ and the FN errors, d_ϵ, with the approximation error tolerance probability, ϵ, of the PAC framework in the approximate recovery scenario (see (3.1), (3.12) and (3.28)).

2.2 PAC Learning View of Group Testing

In this section, we cast the group testing problem as a function learning problem, also termed as learning from examples [137]. Here, a target function $f \in \mathcal{C}$ is learnt using m *training examples* $(\mathbf{a}_i, f(\mathbf{a}_i)), i \in [m]$, with the inputs $\mathbf{a}_i$ drawn independently from a distribution $\mathcal{D}$. The training examples are fed to the learner, which then outputs an estimate of f, also called a hypothesis, and denoted by h. The error between h and f evaluated on unseen *test* data is

$$e(h, f) = \mathbb{P}_{\mathbf{a}\sim\mathcal{D}}(h(\mathbf{a}) \neq f(\mathbf{a})). \tag{2.2}$$

However, the quantity $e(h, f)$ is random because the m training examples are drawn from $\mathcal{D}$. Therefore, we can ask how many training examples are sufficient to ensure that

$$\mathbb{P}(e(h, f) > \epsilon) \leq \delta, \tag{2.3}$$

where $\delta \in (0, 1)$ and $1 - \delta$ is called the *confidence parameter*. Obviously, it is desirable to have small ϵ and δ.

2.2.1 PAC Model for Group Testing

Given a test matrix, $\mathbf{A}$, and the corresponding test outcomes, y_i, $i \in [m]$, consider the problem of learning a hypothesis that can predict the group test outcomes with high confidence. More formally, we consider the ith row of the test matrix, $\mathbf{a}_i, i \in [m]$ as the input, and the outcome of the ith group test, y_i, as a *label* associated with the ith training

Table 2.1: Group Testing as a PAC Learning Problem

	Target	**Training Examples**	**Hypothesis**
Group testing case	$x(\cdot)$	$(\mathbf{a}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}, y_i = x(\mathbf{a}_i))$	$\hat{x}(\cdot)$
PAC learning case	$f(\cdot)$	$(\mathbf{a}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}, f(\mathbf{a}_i))$	$h(\cdot)$

example: $(\mathbf{a}_i, y_i)$. Since the entries of $\mathbf{x}$ corresponding to the non-defective items are 0, we can write y_i as

$$\begin{aligned} y_i &= a_{ij_1}x_{j_1} \vee \ldots \vee a_{ij_k}x_{j_k} \\ &= a_{ij_1} \vee a_{ij_2} \vee \ldots \vee a_{ij_k} \triangleq x(\mathbf{a}_i), \end{aligned} \tag{2.4}$$

where $j_1, j_2, \ldots, j_k$ are the indices of $\mathbf{x}$ corresponding to the defective items. Thus, $x(\mathbf{a}_i)$ is a k-literal logical OR-ing function, and is our target function to learn. In computer science, this problem is referred to as the k-disjunctive function learning problem [138]. The target function space, denoted by $\mathcal{C}$, consists of all k-literal OR-ing functions, where k literals are picked without replacement from n literals in accordance with (2.4).

The relationship between non-adaptive random pooling-based group testing and the function learning problem is summarized in Table 2.1. One could set the distribution from which the entries of the test matrix is drawn as $\mathcal{D} = \mathcal{B}(p)$. Another well-studied random pooling design is to uniformly and independently sample s items with replacement in each group test [5].

In the notation of group testing, (2.2) can be written as

$$e(\hat{x}(\cdot), x(\cdot)) = \mathbb{P}_{\mathbf{a}\sim\mathcal{D}}(\hat{x}(\mathbf{a}) \neq x(\mathbf{a})), \tag{2.5}$$

which denotes the error probability on future group tests, i.e., after m training samples are observed and the mapping $\hat{x}(.)$ is learnt. Thus, in the PAC learning view, we seek to determine the number of training examples, m, and a mapping from the training examples to a hypothesis, $\hat{x}(\cdot)$, such that with a confidence probability $1-\delta$, the error between $x(\cdot)$ and $\hat{x}(\cdot)$ is at most ϵ [137], [133, Chapter 2], i.e.,

$$\mathbb{P}(e(\hat{x}(\cdot), x(\cdot)) > \epsilon) \leq \delta \tag{2.6}$$

holds true, where $e(\cdot,\cdot)$, as defined in (2.5), is a random variable.

A fundamental difference between the PAC formulation of the group testing presented here as compared with the classical PAC-learning problem is that the data distribution, $\mathcal{D}$, is choosable. For example, the distribution can be set based on the hypothesis class, $\mathcal{C}$, i.e., based on the sparsity parameter, k. Even though the group testing *algorithms* considered in this work do not use the knowledge of k during the defective set recovery process, the design of the testing matrix and hence, the test data distribution, depends on this knowledge.

We demonstrate that, when $\epsilon = 0$, the bounds on m derived via the PAC model reduce to the exact recovery results derived in [4, 6]. Note that, in classical group testing, the goal is to correctly identify the defective set, whereas in the PAC learning view of group testing, we seek to learn a hypothesis satisfying (2.6). The following Lemma relates the PAC learning to group testing in the exact recovery case.

Lemma 2.1. *Let $\mathcal{D}$ be a distribution such that $\mathbb{P}_{\mathcal{D}}(a_j = 1) \in (0,1)$, $j \in [n]$ and a_js are independent. Let $\mathcal{C}$ denote the set of all k-literal OR-ing functions in n-dimensional space, where $k < n$. Let $\hat{x} : \{0,1\}^n \rightarrow \{0,1\}$ (correspondingly $\hat{\mathcal{K}}$) be a function in $\mathcal{C}$ that is learnt using a set of m training samples. Then, $\hat{\mathcal{K}} = \mathcal{K}$ if and only if $\mathbb{P}_{\mathbf{a}\sim\mathcal{D}}(\hat{x}(\mathbf{a}) \neq x(\mathbf{a})) = 0$.*

The proof of Lemma 2.1 is presented in Appendix A.1. Lemma 2.1 says that provided the marginal probability of every entry of the vector $\mathbf{a}$ is bounded in the open interval $(0, 1)$ and the entries of $\mathbf{a}$ are drawn independently, the notions of recovery in (2.1) and (2.6) when $\epsilon = 0$ are equivalent.

2.3 Chapter Summary

In this chapter, we have set the stage for analyzing non-adaptive group testing algorithms using a PAC framework. We formally described the non-adaptive group testing problem, followed by an overview of some well-known binary group testing algorithms available in the literature. Then, some important results on achievability bounds available in the literature for exact and approximate recovery scenarios were summarized. The concept of approximate recovery and its applications were stated.

Next, the PAC learning view of the group testing problem was formally described along with the PAC model for group testing. In particular, the group test outcomes are viewed as an output of a k-literal OR-ing function. The goal of the group testing algorithm is then to learn a k-literal OR-ing function that satisfies a certain notion of correctness at a certain confidence level. Lemma 2.1 shows the equivalence between the PAC learning and the group testing problem when $\epsilon = 0$, i.e., under exact recovery case. We emphasize that the PAC formulation presented in this chapter is different from the classical PAC analysis since the data distribution can be chosen based on the hypothesis space containing the target function. With this foundation, we derive sufficiency bounds for well-known binary group testing algorithms in the next chapter.

3 Probably Approximately Correct Bounds for Group Testing

Chapter Highlights

In this chapter, we derive a sufficiency bound for well-known non-adaptive binary group testing algorithms. The first part of this chapter presents a PAC analysis for an algorithm that makes only FP errors. This section is comprised of two parts: 1) the COMP algorithm with Bernoulli test matrix design (COMP-B) and 2) the COMP algorithm with near-constant row-weight test matrix design (COMP-R). In both cases, we establish the relationship between the approximation error probability, ϵ, and the number of FPs tolerated. Then, the main results, i.e., the sufficiency bounds, are stated. Next, special cases of the bound, including the exact recovery scenario obtained by setting $\epsilon = 0$, are discussed. The optimal Bernoulli parameter, p, in the case of the COMP-B bound and the importance of optimizing the Chernoff parameter in the case of the COMP-R bound are then discussed. Also, we extend the results on the expected stopping time and the tail probability for the stopping time in the context of the coupon collector problem (CCP) to *collecting a subset of coupons*. The new results are used to derive the COMP-R bound.

The second part of this chapter presents a PAC analysis with FN errors. In particular, the DD algorithm with Bernoulli test matrix design is analyzed. As before, we establish the relationship between the approximation error probability, ϵ, and the number of FNs tolerated, followed by the main result. By setting $\epsilon = 0$, we see that the derived bound agrees with the exact recovery bound available in the literature under the sub-linear regime. An order-wise analysis is performed for all three cases. We observe that our bound is $\propto \log(C_d/\delta)$ with a constant, $C_d = 2$ for the COMP-R algorithm and $C_d = 1$ for COMP-B and DD algorithms when we fix n, k and ϵ. Further, the sufficient number of tests is $\propto (\log(1/\epsilon) + 1/\epsilon)$ when we fix n, k and δ.

Finally, we show that the novel theoretical bounds *agree* with that obtained by the simulations by a suitable numerical experiment. In other words, we comment on the tightness of the derived bounds. We then analyze the effect of approximation error tolerance on the bounds, followed by the side effects of performing insufficient number of tests. Lastly, we pictorially illustrate how *testing rate surface* and *sufficient tests contours* behave as a function of ϵ and δ.

3.1 Introduction

The previous chapter discussed the motivation for applying PAC analysis to the non-adaptive group testing problem. Recall from Section 2.2.1 that two parameters characterize the PAC-based analysis of group testing: 1) the approximation error tolerance, ϵ, and 2) the confidence, $1 - \delta$. Further, recall that n denotes the number of items, k $(< n)$ denotes the number of defective items in the population, and $\mathcal{K}$ denotes the defective set with $|\mathcal{K}| = k$. In this chapter, we apply the PAC analysis to well-known binary group testing (decoding) algorithms under the following three scenarios [4, Chapter 2] [6]

1. Combinatorial Orthogonal Matching Pursuit (COMP) algorithm with Bernoulli test matrix design (COMP-B);
2. COMP algorithm with a near-constant row-weight (random) test matrix design (COMP-R);
3. Definite Defectives (DD) algorithm with a Bernoulli test matrix design,

and derive a sufficiency bound for the number of tests, m, in each case as a function of n, k, ϵ, and δ. This enables us to perform a finer analysis of the group testing algorithms by separately accounting for the randomness in the test matrix $\mathbf{A} \in \{0, 1\}^{m \times n}$ and the defective set $\mathcal{K}$.

We note that exact recovery bounds can be obtained by setting $\epsilon = 0$. As a result, the exact defective set recovery results found in the literature emerge as a special case of our analysis. In other words, we develop a common framework to arrive at sufficiency bounds on the number of tests for both exact and approximate set recovery with high probability.

Recall from Section 2.1.2 that the available bounds on the sufficient number of tests for approximate defective set recovery also depend on the underlying test matrix design, i.e., the distribution from which the test matrix is drawn. In our thesis, we are interested in the number of FP or FN errors allowed at a given n, k, and with the confidence parameter, δ. Lastly, we work mainly in the non-asymptotic regime, which is of practical interest, as mentioned in Section 2.1.1.

Note that, under the group testing model described in Section 2.1, COMP only make FP errors, i.e., $\hat{\mathcal{K}} \supseteq \mathcal{K}$, while DD only makes FN errors, i.e., $\hat{\mathcal{K}} \subseteq \mathcal{K}$, where $\hat{\mathcal{K}}$ denotes an estimate of $\mathcal{K}$ output by the group testing algorithm [4]. COMP and DD are some of the most popularly studied algorithms in the group testing literature [7]. As stated above, the COMP algorithm is appropriate when one can tolerate FP errors, i.e., when the cost of FP error is low relative to the cost of FN errors. Similarly, the DD algorithm is suitable in scenarios when FN errors can be tolerated (see Section 2.1.2 for examples of such applications). Since the PAC formulation for group testing is a new introduction, we analyze the sufficient number of tests with the COMP and DD decoding algorithms. These algorithms show *near-optimal* behavior in terms of the sufficient number of tests asymptotically as the number of items goes to infinity, in the exact recovery case. In this chapter, we answer the following question: *When the confidence parameter, δ, and the approximation error tolerance, ϵ, are nonzero, what is the scaling of the number of tests given a number of FN or FP errors (corresponding to the nonzero ϵ) that is allowed?* Towards this end, we analyze these algorithms separately in the sequel. Before we proceed to present the main results, we summarize the contributions of this chapter below:

1. We use the PAC-learning formulation of the non-adaptive group testing presented

in Chapter 2 for deriving a sufficiency bound on the number of tests in both exact and approximate recovery conditions for the well-known and popular binary group testing algorithms: COMP and DD. In contrast to existing works, we optimize the design parameters to get a tighter bound on the sufficient number of tests for COMP-R. Further, the existing results on the coupon collector problem do not apply to approximate recovery. Therefore, we also extend the analysis of the coupon collector problem to handle *collection of only a subset of coupons*. The new results are then used to develop measurement bounds for approximate recovery of the defective set using the COMP algorithm with near-constant row-weight test matrix design.

2. We derive the order-wise behavior of the PAC bounds for large n and k. When we fix n, k, and ϵ, the sufficient number of tests obtained by the PAC analysis is $\propto \log(C_d/\delta)$ with a constant, $C_d = 2$ for COMP-R and $C_d = 1$ for COMP-B and DD. Further, the sufficient number of tests is $\propto (\log(1/\epsilon) + 1/\epsilon)$ when we fix n, k and δ.

3. We relate the number of FP errors tolerated, denoted by g_ϵ (similarly, the number of FN errors tolerated, d_ϵ) to the approximation error probability ϵ used in the PAC formulation. This way of relating the theoretical results to the practical metrics of interest makes the results appealing to practitioners also.

4. The PAC-based analysis allows us to trade-off the *accuracy* of defective set recovery with the *confidence* with which the decoded set meets that accuracy. We present a visualization of this trade-off in the form of a sufficient number of tests *contour/surface*,

which shows its dependence on the approximation error tolerance and the probability of failure to meet the required error tolerance. Also, PAC analysis enables one to characterize a lower bound on the cumulative distribution of the approximation errors (see (2.6)).

The technical novelty in this part of the thesis is that the existing results for the coupon collector problem (CCP) [139] cannot be directly applied to derive COMP-R bounds in the approximate recovery case. Therefore, we derive the expressions for the expected stopping time and the tail probability bound for a *subset coupon collection problem* (SCCP), where one is interested in acquiring only a subset of coupons to complete the collection, in Lemma 3.2. We note CCP finds applications beyond group testing, e.g., in animal species census or biodiversity sampling, electrical fault detection and network failure analysis, and data compilation, to name a few [140–142]. Extensions to the CCP problem continue to be an active area of research [143, 144]. To our knowledge, SCCP analysis has not been directly addressed in the literature, making it a potentially useful contribution of independent interest.

Further, we optimize the Chernoff parameter by bounding the error probability in two independent parts in the COMP-R analysis, thereby obtaining a tighter bound as compared to that in the literature for the exact recovery case [6].

Lemma 3.1 and Lemma 3.3(c) forms the basis for deriving the sufficiency bound for COMP-B and DD algorithms: the exact computation of the probabilities that g non-defective items are hidden and d defectives items are unidentified (respectively) is characterized. Also, the order-wise analysis of DD bound involves simplification of an implicit equation in (3.30) and solving the transcendental equation in (3.33). Lastly, we give the

Table 3.1: PAC-based bounds vs. existing bounds on the sufficient number of tests for exact recovery for COMP and DD algorithms, where $\kappa(\gamma') \triangleq \max\{\gamma', 1-\gamma'\}$ with $k = n^{1-\gamma'}$. Also, we set $\delta' \triangleq C_\delta n^{-\delta}$, where the constant $C_\delta = 2$ for COMP-R and $C_\delta = 1$ for COMP-B and DD with C_η denoting an algorithm-specific constant.

Algorithm	**PAC Bound with** $\epsilon = 0$	**Existing Bound**
COMP-B	$e(k+1)\left(\log n + \log\left(\frac{1}{\delta}\right)\right)$	$ek\left(\log n + \log\left(\frac{1}{\delta'}\right)\right)$ [6, Thm. 4]
COMP-R	$\frac{ek}{1-\frac{C_\eta}{\sqrt{n \log n}}}\left[\log(n-k) + \log\left(\frac{2}{\delta}\right)\right]$	$2ek\left(\log n + \log\left(\frac{2}{\delta'}\right)\right)$ [6, Thm. 3]
DD	$ek\left(\log\left(\frac{n}{k}\right) + \frac{\log\left(\frac{1}{\delta}\right)}{\log\left(\frac{n}{k}\right)} + 2\right)$	$ek\left(\kappa\left(\gamma'\right)\log n + \log\left(\frac{1}{\delta'}\right)\right)$ [4, Thm. B.3]

sufficiency bounds for the COMP and DD algorithms in terms of the allowed number of false positives or false negatives. In case the practitioner is unable to perform the required number of tests or they perform additional tests, the expressions in (3.3), (3.13) and (3.30) can be used to determine the increase or decrease in the FP or FN errors to be expected, respectively, and at different confidence levels.

Before we proceed forward, we summarize simpler, order-wise versions of our PAC bound for COMP and DD algorithms in Table 3.1 along with the existing bounds, in the exact recovery scenario, for quick reference. We note that the bounds derived in the thesis are more general, as they are applicable even under approximate recovery scenarios. From Table 3.1, we see that our COMP-B and DD bounds are similar to the ones in the literature, whereas the COMP-R bound is tighter by a constant factor since $C_\eta/\sqrt{n \log n} \ll 1$ for large n. Further, the existing bounds implicitly use a vanishing error probability $\delta' = n^{-\delta}$, whereas our bounds do not have such assumptions. Often, in practice, a fixed but small error probability is acceptable.

Also, the order-wise versions of the general bounds are summarized in Table 3.2 for reference. We note that there are no equivalent existing bounds in the general scenario where

Table 3.2: The general order-wise PAC bounds for COMP and DD algorithms, where $\delta \in (0,1)$, $\epsilon \geq 0$.

Algorithm	**Order-Wise PAC Bound**
COMP-B	$2ke \left[\log\left(\frac{n}{ke\epsilon+1}\right)+1+\frac{\log\left(\frac{1}{\delta}\right)}{ke\epsilon+1}\right]$
COMP-R	$2ke\left[\log\left(\frac{n}{ke\epsilon+1}\right)+1+\log\left(\frac{2}{\delta}\right)\left[\frac{1}{ke\epsilon+1}+\frac{e}{2k}\right]\right]$
DD	$ke\left(\log\left(\frac{n}{k}\right)+\frac{\log\left(\frac{1}{\delta}\right)}{(ke\epsilon+1)\log\left(\frac{n}{k}\right)}+\frac{\log\left(\frac{ke}{ke\epsilon+1}\right)}{\log\left(\frac{n}{k}\right)}\right)$

the effects of both δ and ϵ are accounted for, with COMP and DD decoding algorithms. However, as stated earlier, the authors in [7] provide some insight into how FP or FN errors scale for COMP and DD algorithms (see Section 2.1.2). In particular, they show that when the number of tests, $m \geq (1+\eta)ek\log(n/k)$, where $\eta > 0$, the average number of FPs (or FNs) in COMP (or DD) algorithm behaves as $o(k)$. Finally, the lower bound based on the counting argument states that $k\log(n/k)$ tests are necessary [4, 7]. From Table 3.1 and Table 3.2, we see that the derived sufficiency bounds follow a similar trend in terms of their dependence on n and k.

3.2 PAC Analysis with False Positive Errors

In this section, we develop a PAC analysis for the case where only false positive errors occur, i.e., when $\mathcal{K} \subseteq \hat{\mathcal{K}}$. This is the case with the COMP algorithm. As mentioned earlier, COMP has been known under various names in the literature. For example, the authors in [5, 6] couple the test matrix design with a decoding procedure. In particular, they called COMP with the Bernoulli test design as a Column Matching (CoMa) algorithm. Also, COMP with a near-constant row-weight test design is called the Combinatorial Basis

Pursuit (CBP) algorithm. One can view the CoMa algorithm as a *column-wise* decoding view of the decoding procedure, while the CBP algorithm as a *row-wise* decoding view of the decoding procedure [6]. The two algorithms are mathematically equivalent in the sense that they always output the same $\hat{\mathcal{K}}$. More recently, the authors in [7] suggest decoupling the test matrix design from the decoding algorithm. Therefore, as mentioned earlier, we refer to the COMP algorithm with Bernoulli design as COMP-B and the COMP algorithm with near-constant row-weight design as COMP-R. However, we stick to the column-wise vs. row-wise decoding procedures for the analysis and the derivation of the PAC bound. Consequently, the upper bound analysis and takeaways are different in the two cases.

3.2.1 The COMP Algorithm: Bernoulli Test Design (COMP-B)

We start by briefly summarizing the column-wise decoding view of the COMP algorithm [6]. We declare an item as defective if the ones in the column of the test matrix corresponding to that item are a subset of the ones in the outcome vector. Otherwise, the item is declared as non-defective. The algorithm never classifies a defective item as a non-defective. However, the estimate may contain false positives, which occur when non-defective items do not participate in any of the negative outcome tests in the training phase. Such items are also called *hidden non-defectives*.

Suppose the hypothesis output by COMP-B has G hidden non-defective items. Then, from (2.5), the probability that $\hat{x}(\cdot) \equiv \hat{x}$ differs from $x(\cdot) \equiv x$ for the next group test, defined as $\mathbb{P}_{\mathbf{a}_i \sim \mathcal{B}(p)}(\hat{x}(\mathbf{a}_i) \neq x(\mathbf{a}_i))$, is a function of G. Suppose we want the error between $\hat{x}$ and x to be at most ϵ. In turn, this requires $G \leq g_\epsilon$, where g_ϵ can be computed from

$$\mathbb{P}_{\mathbf{a}_i \sim \mathcal{B}(p)}\left(\hat{x}(\mathbf{a}_i) \neq x(\mathbf{a}_i)\right) = (1 - (1-p)^G)(1-p)^k \leq \epsilon$$

$$\Rightarrow g_\epsilon = \left\lfloor \frac{\log\left(1-\epsilon/\left(1-p\right)^k\right)}{\log(1-p)} \right\rfloor. \tag{3.1}$$

Thus, the bound in (2.6) reduces to $\mathbb{P}(G > g_\epsilon) \leq \delta$. In order to proceed further, we need the following Lemma.[1]

Lemma 3.1. *The probability that a fixed set of g $(1 \leq g < k)$ non-defective items remain hidden in all m tests is given by*

$$\mathbb{P}_g^h(m) = \left(1-(1-p)^k + (1-p)^{g+k}\right)^m. \tag{3.2}$$

From the above Lemma, we obtain the following sufficiency condition on m for COMP-B under the PAC model:

Theorem 3.1. *The sufficient number of tests such that the predicted outcome based on the estimated defective set does not agree with the true outcome on future group tests with probability at most ϵ and confidence parameter $1-\delta$ is*

$$m_S = \frac{\log\binom{n-k}{g_\epsilon+1} + \log\frac{1}{\delta}}{\log\left(1/(1-(1-p)^k + (1-p)^{g_\epsilon+k+1})\right)}, \tag{3.3}$$

with g_ϵ as given by (3.1).

From (3.3), we observe that our bound on the number of tests is a function of g_ϵ (equivalently, ϵ) and δ in addition to n and k. In contrast, the existing bounds in the literature do not capture the effect of both ϵ and δ. When ϵ tends to zero, our PAC bound reduces to the bound given in the literature [6] and is discussed in a paragraph after (3.6). Also, as we shall see in Section 3.2.1.2, the PAC bound shows $k \log n$ dependency, similar to the counting bound [4].

[1] All the proofs to the results present in this chapter can be found in Appendix B.

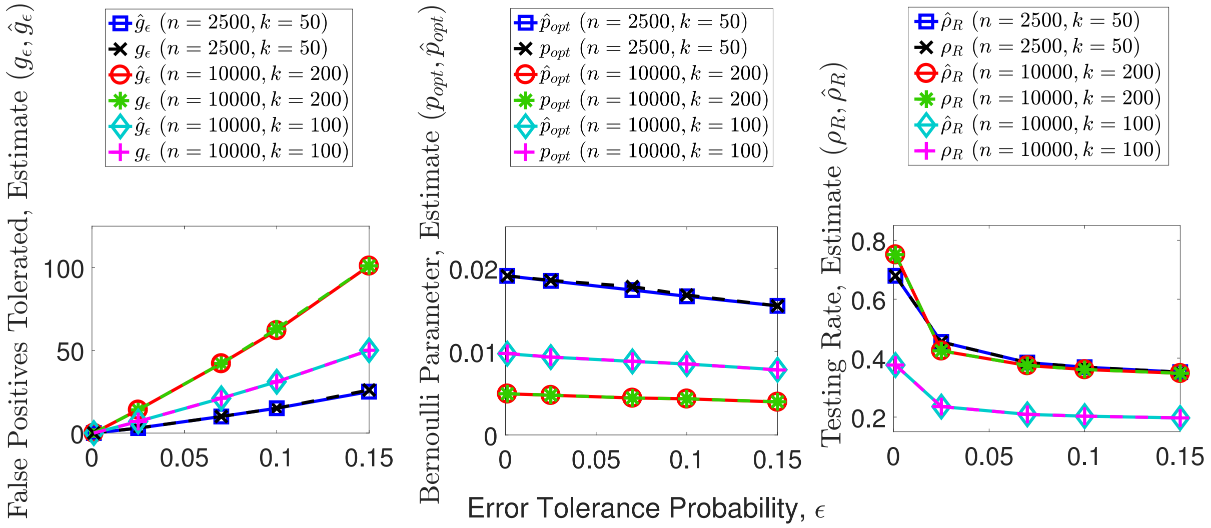


Figure 3.1: Comparison of the solution of MINLP using grid-search vs. the implicit equations for m_S, p_{opt} and g_ϵ at $\delta = 0.01$ over various approximation error tolerance, ϵ, when $(n, k) \in \{(2500, 50), (10000, 200), (10000, 100)\}$.

3.2.1.1 Optimum Bernoulli Parameter

We can determine the optimum value of p for which (3.3) is minimized by solving the following mixed-integer non-linear program (MINLP):

$$\begin{aligned}\hat{m}_S, \hat{p}_{\text{opt}}, \hat{g}_\epsilon = \underset{\substack{m \in \mathbb{Z}_+,\ p \in (0,1) \\ g_\epsilon \in \mathbb{Z}_+ \cup \{0\}}}{\arg\min}\ m \\ \text{s.t. } (1 - (1-p)^{g_\epsilon})(1-p)^k \le \epsilon \\ \text{and } \binom{n-k}{g_\epsilon + 1}\left(1 - (1-p)^k + (1-p)^{g_\epsilon + 1 + k}\right)^m \le \delta. \qquad (3.4)\end{aligned}$$

The constraints above correspond to the error probability (ϵ) and confidence $(1 - \delta)$ requirements. Since the problem does not admit a closed-form solution, we solve (3.4) using a grid-search over m, p and g_ϵ, ensuring that we are varying them over the feasible range.

In order to obtain insights, we also minimize (3.3) using a heuristic fixed-point iteration method. Note that (3.3) holds for any $g_\epsilon \in \mathbb{Z}_+$. Further, when $g_\epsilon = 0$ (i.e., for exact recovery), p_{opt} in (3.5) reduces to $1/(k+1)$ or $O(1/k)$ for large k. We can use $1/(k+1)$ as an initial value for p and find the largest number of false positive errors g_ϵ for which the ϵ-constraint is satisfied from (3.1). Then, for the given g_ϵ, we seek the p for which the term $\left(1-(1-p)^k+(1-p)^{g_\epsilon+1+k}\right)$ is minimized, since this results in the smallest m for which the δ-constraint is satisfied. Differentiating this term and setting it equal to zero yields the optimum value of p as

$$p_{\text{opt}} = 1 - \sqrt[g_\epsilon+1]{k/(k+g_\epsilon+1)}. \tag{3.5}$$

We now iterate between (3.5) and (3.1) to obtain the optimal g_ϵ and p_{opt}.

Although it appears hard to prove analytically, we find that this procedure converges, and yields the globally optimal solution. We compare the numerical solution, $\hat{g}_\epsilon$ and $\hat{p}_{\text{opt}}$, obtained by solving (3.4) with that obtained using the fixed-point procedure, namely, g_ϵ and p_{opt}. Also, we compare the value of the testing rate $\rho_R \triangleq m_S/n$ computed from Theorem 3.1 along with (3.5) and (3.1) with that obtained by solving (3.4), i.e., $\hat{\rho}_R \triangleq \hat{m}_S/n$. The computed values and the solution of the MINLP are plotted in Figure 3.1 over various values of the approximation error tolerance, ϵ, with $\delta = 0.01$ for a collection $(n,k) \in \{(2500,50),(10000,200),(10000,100)\}$. It can be observed that the expression for p_{opt} as given in (3.5) is consistent with the expression for g_ϵ in (3.1) which further yields the optimum m_S as given by (3.3).

3.2.1.2 Order-Wise Analysis of (3.3)

We note that the $\log\binom{n-k}{g_\epsilon+1}$ term appearing in the expression for m_S is similar to the $\log\binom{n-k}{\tau}$ term obtained in [7, 93], where τ denoted the number of false positives. Further, the achievability bound in [93] is characterized by the conditional mutual information term in the denominator. In this sub-section, we elucidate the behavior of the denominator in (3.3) for large n and k, resulting in an order-wise characterization of the achievability bound.

First, we can relate the COMP-B bound for the exact recovery case in [6] to our PAC bound. From (3.1), we see that $\epsilon = 0$ implies $g_\epsilon = 0$. In the *sub-linear* regime, $k = \Theta(n^\beta), \beta \in (0, 1)$, the numerator in (3.3) can be upper bounded by $\log n + \log(1/\delta)$. Let $z \triangleq 1 - (1-p)^k + (1-p)^{k+1}$. Using $\log(1/\tilde{z}) \geq 1 - \tilde{z},\ \tilde{z} > 0$, the denominator in (3.3) can be lower bounded as follows:

$$\log\left(1/\tilde{z}\right) \geq p(1-p)^k = \frac{1}{ek} - O\left(\frac{1}{ek^2}\right) \geq \frac{1}{e(k+1)} \tag{3.6}$$

where $p = 1/k$ [6], and Laurent's series is used in the penultimate step as k gets large. Combining the reduced expressions, we obtain that $m = e(k+1)\left(\log n + \log(1/\delta)\right)$ are sufficient for exact recovery. From the sufficiency result for m in [6, Theorem 4], with $\delta' \triangleq n^{-\delta}$, $m = ek(\log n + \log(1/\delta'))$ is sufficient, which is similar to our bound at $\epsilon = 0$.

Second, we analyze how the achievability bound in (3.3) behaves as as a function of δ and ϵ when n and k grow large. Define $z \triangleq (1-p)^k - (1-p)^{g_\epsilon+k+1}$. Using $\binom{n-k}{g_\epsilon+1} \leq (e(n-k)/(g_\epsilon+1))^{g_\epsilon+1}$ along with the fact that $\log(1/(1-z)) \geq z,\ z \in [0,1]$ in (3.3), we see that

$$m_S = \frac{(g_\epsilon+1)\log\left(\frac{n-k}{g_\epsilon+1}\right) + (g_\epsilon+1) + \log\left(\frac{1}{\delta}\right)}{(1-p)^k\left(1-(1-p)^{g_\epsilon+1}\right)}, \tag{3.7}$$

tests are sufficient to ensure no more than g_ϵ errors with confidence $1-\delta$. Set $p = 1/k$ in (3.7). Using $(1-x)^r \leq e^{-xr}$ for $x \in [0,1], r \geq 0$ and $1 - e^{-y} \geq y/2$, for $y \in [0,1]$, we lower bound the second factor in the denominator of (3.7) with $x = p$, $r = g_\epsilon + 1$ and

$y = (g_\epsilon + 1)/k$ as:

$$1 - (1-p)^{g_\epsilon + 1} \geq \frac{g_\epsilon + 1}{2k}. \tag{3.8}$$

Using $(1-p)^k \rightarrow 1/e$ for large k and $\log(n-k) < \log n$, along with (3.8) in (3.7), we get

$$\begin{aligned} m_S &= 2ke \left[\log\left(\frac{n}{g_\epsilon + 1}\right) + 1 + \frac{\log\left(\frac{1}{\delta}\right)}{g_\epsilon + 1} \right] \\ &= 2ke \left[\log\left(\frac{n}{ke\epsilon + 1}\right) + 1 + \frac{\log\left(\frac{1}{\delta}\right)}{ke\epsilon + 1} \right], \end{aligned} \tag{3.9}$$

where we have used $p = 1/k$ in (3.1) and $\log(1-x) \rightarrow -x$, $x \rightarrow 0$ along with $(1-p)^k \rightarrow 1/e$ as n and k grow, to get $g_\epsilon = ke\epsilon$. From (3.9), we see that $m_S \propto \log(1/\delta)/\epsilon$ for very small δ. Further, we see that the dependency of our bound on $\epsilon > 0$ is $\propto (\log(1/\epsilon) + 1/\epsilon)$. On the other hand, we get $m_S \approx 2ke\left(\log n + 1 + \log(1/\delta)\right)$ by setting $\epsilon = 0$, i.e., for the exact recovery case, accounting for the confidence parameter δ.

3.2.2 The COMP Algorithm: Near-Constant Row-Weight Test Design (COMP-R)

Under the row-wise decoding view [6], we declare all the items participating in the group tests with negative outcomes as non-defective, and the remaining items as defective. It is clear that the COMP-R makes only false positive errors. In contrast to COMP-B, the analysis technique is different in the COMP-R case. As we shall see from the numerical results, the COMP-B bound is tighter (i.e., it follows the simulated behavior more closely) than the COMP-R bound, in the exact recovery case. On the other hand, the COMP-R bound is tighter in the approximate recovery case, i.e., as g_ϵ increases. Therefore, it is of interest to study COMP-R bounds separately. We also note that the COMP-B analysis yields an analytical expression for the optimum parameters of the test design, as seen earlier.

The analysis of the COMP algorithm with near-constant row-weight design is related to the theory of Coupon Collector Problem (CCP), where the goal is to collect distinct *coupons* to obtain a set of all available coupons [6]. More precisely, there is one of n distinct types of coupon, say, inside each cereal box. How many cereal boxes should a person purchase in order to collect all the n coupons?

The expected stopping time, i.e., the average number of purchases required to succeed (with replacements, as coupons can repeat across purchases) is nH_n, where $H_n \triangleq \sum_{i=1}^{n} 1/i$ denotes the nth Harmonic number for any $n \in \mathbb{N}$ and $H_0 \triangleq 0$. A well known asymptotic approximation for H_n is $H_n \approx \log n + \gamma + 1/2n + O(1/n^2) \approx O(\log n)$, where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant. Therefore, the expected stopping time is $O(n \log n)$ for sufficiently large n. One can bound H_n as $\log n + \gamma < H_n < \log(n+1) + \gamma$. Also, the probability that the stopping time exceeds $\chi n \log n$ is at most $n^{-\chi+1}$, for $\chi > 1$ [139, 145].

Before we present the main result, on similar lines to [6], we relate the COMP-R decoding procedure to the CCP as follows. The COMP-R procedure collects items from a sequence of tests. Consider an s-length test vector whose entries index the items being pooled in a given test. The s items are chosen with replacement.[2]

Following [6], it is clear that there is a natural bijection between the s-length vector and n-length row vector $\mathbf{a}_i$ of the ith group test. Since the probability of an item occurring at any location of the s-length vector is uniform and independent, and this property holds

[2] A given item can be potentially picked more than once apart from being picked once or not picked at all. By assigning the number of times an item is picked in the ith test to the (i,j)th entry of the test design matrix, the authors in [146] present a multi-group testing model using standard (not Boolean) arithmetic. However, in our work, the (i,j)th entry of the testing matrix is set to one irrespective of whether the jth item is picked once or more than once in ith test, and is set to zero otherwise.

across the tests, the items in any sub-sequence of m' tests with negative outcomes may be viewed as the outcome of a process of selecting a single chain of sm' coupons. Since the outcome of a single test with s items is negative with probability $((n-k)/n)^s$, in order for COMP-R to succeed with m tests, in the exact recovery case, we require

$$ms\left(\frac{n-k}{n}\right)^s \geq (n-k)H_{n-k} \geq (n-k)\left[\log(n-k)+\gamma\right]. \tag{3.10}$$

For approximate recovery, i.e., g_ϵ errors, it suffices to *stop collecting* items participating in negative outcome tests once we collect $n-k-g_\epsilon$ non-defective items. Lemma 3.2 presents the expected stopping time and tail probability in this case.

Lemma 3.2. *For the coupon collector problem with w distinct coupons, with each coupon being picked in an equally likely and independent fashion, if any subset containing $w-g$ distinct coupons are sufficient to complete the collection, then*

(a) The expected stopping time is $w[\log w + \gamma - H_g]$, where γ is the Euler-Mascheroni constant and H_g is the gth Harmonic number as defined earlier.

(b) For any $\chi > 1$, the stopping time exceeds $\chi w[\log w + \gamma - H_g]$ with probability at most $w^{(g+1)(-\chi+1)}\frac{e^{(g+1)\chi[H_g-\gamma]+g}}{(g+1)^{(g+1)}}$.

From Lemma 3.2(a), the RHS of (3.10) can be modified with $g = g_\epsilon$ and $w = n-k$ to obtain

$$ms\left(\frac{n-k}{n}\right)^s \geq (n-k)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]. \tag{3.11}$$

With this background, we are ready to present the result on the sufficient number of tests required by COMP-R in the PAC framework. Since COMP can only make false positive errors, the PAC equation (2.6) takes the form $\mathbb{P}(G > g_\epsilon) \leq \delta$.

We note that an error in the prediction occurs when none of the k defectives out of n

items participate in the group test and at least one of G hidden non-defectives participates, making the predicted outcome 1 whereas the true outcome is 0. On similar lines as (3.1), the largest value of g_ϵ such that the error between $\hat{x}$ and x is at most ϵ can be computed as

$$\mathbb{P}_{\mathbf{a}_i \sim \mathcal{S}}(\hat{x}(\mathbf{a}_i) \neq x(\mathbf{a}_i)) = \left(1-\frac{k}{n}\right)^s \left[1-\left(1-\frac{G}{n-k}\right)^s\right] \leq \epsilon,$$

where $\mathcal{S}$ denotes a uniform distribution over all s-length vectors. This yields

$$g_\epsilon = \left\lfloor (n-k)\left[1-\left(1-\frac{\epsilon}{\left(1-\frac{k}{n}\right)^s}\right)^{1/s}\right]\right\rfloor. \tag{3.12}$$

The following theorem characterizes the sufficient number of tests required by COMP-R in the PAC framework.

Theorem 3.2. *Suppose s items are chosen with replacement in each group test. A sufficient number of tests such that the predicted outcome based on the estimated defective set does not agree with the true outcome on future group tests with probability at most ϵ and confidence parameter $1-\delta$ is*

$$m_S = \frac{\chi(n-k)}{(1-\eta)s\left(\frac{n-k}{n}\right)^s}\left[\log(n-k)+\gamma-H_{g_\epsilon}\right], \tag{3.13}$$

where g_ϵ is given by (3.12),

$$\chi = \frac{\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1}+\frac{g_\epsilon}{g_\epsilon+1}+\log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}{\log(n-k)+\gamma-H_{g_\epsilon}}, \tag{3.14}$$

$\eta = (-C+\sqrt{C^2+4C})/2 \in (0,1)$, *with*

$$C \triangleq \frac{\log\left(\frac{1}{(1-c)\delta}\right)}{\left(\frac{n-k}{s}\right)\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1}+\frac{g_\epsilon}{g_\epsilon+1}+\log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}, \tag{3.15}$$

and $c \in (0,1)$ is a design parameter.

In the subsequent subsections, we understand the implications of Theorem 3.2 in more detail. A quick observation shows the dependency of ϵ and δ in addition to n and k, which the bounds in the literature do not capture. As we shall see, when we set $\epsilon = 0$, our PAC bound is tighter by a constant amount as compared to the bound in [6]. Further discussions are deferred to Section 3.2.2.3 and Section 3.4.4. Like the COMP-B bound, our COMP-R bound shows a $k \log n$ dependency, similar to the lower (counting) bound [4].

3.2.2.1 Order-Wise Analysis - Exact Recovery Case

We now discuss the order-wise behavior of (3.13) for large n and k, and specialize the result to the case when $\epsilon = 0$. Note that differentiating (3.10) with respect to s and setting the derivative equal to 0 yields $s^* = 1/\log(n/(n-k))$. Using this value of s^*, we obtain the following corollary.

Corollary 3.1. *With χ and η as specified in Theorem 3.2 computed at $s = s^* \triangleq 1/\log(n/(n-k))$, a sufficient number of tests such that the predicted outcome based on the estimated defective set does not agree with the true outcome on future group tests with probability at most ϵ and confidence parameter $1-\delta$ is*

$$m_S = \frac{\chi k}{(1-\eta)} \left(\frac{n}{n-k}\right)^{s^*} \left[\log(n-k) + \gamma - H_{g_\epsilon}\right], \tag{3.16}$$

with g_ϵ as given by (3.12).

Recall that $g_\epsilon = 0$ corresponds to the exact recovery case as considered in the literature [5]. Also, for reasonably large n, $(n/(n-k))^{s^*} \to e$ from below. Substituting for χ

from (3.14) with $g_\epsilon = 0$ into (3.16) leads to

$$m_S = \frac{ek}{(1-\eta)}\left[\log(n-k) + \log\left(\frac{2}{\delta}\right)\right]. \tag{3.17}$$

We now set $s = s^*$, $c = 1/2$, and $g_\epsilon = 0$ and reduce (3.15) to

$$C = \frac{s^* \log\left(\frac{2}{\delta}\right)}{(n-k)\left[\log\left(\frac{2}{\delta}\right) + \log(n-k)\right]} \tag{3.18}$$

$$\approx \begin{cases} \frac{s^*}{(n-k)} & , \ \delta \ll \frac{2}{n-k} \\ \frac{s^* \log\left(\frac{2}{\delta}\right)}{(n-k)\log(n-k)} & , \ \delta \gg \frac{2}{n-k} \end{cases} \tag{3.19}$$

Using the fact that $s^*/(n-k) \ll 1$ holds for large n and in the sub-linear regime along with (3.18) and (3.19), a Taylor series approximation yields

$$\eta \approx \begin{cases} \sqrt{\frac{s^*}{(n-k)}} & , \ \delta \ll \frac{2}{n-k} & (3.20a) \\ \sqrt{\frac{s^* \log\left(\frac{2}{\delta}\right)}{(n-k)\log(n-k)}} & , \ \delta \gg \frac{2}{n-k} & (3.20b) \\ \sqrt{\frac{s^* \log\left(\frac{2}{\delta}\right)}{(n-k)\left[\log\left(\frac{2}{\delta}\right) + \log(n-k)\right]}} & , \ \delta \sim \frac{2}{n-k} & (3.20c) \end{cases}$$

where, in (3.20c), $a \sim b$ is used to signify that a and b are of the same order.

From (3.20c), we see that η scales as $O\left(1/\sqrt{n \log n}\right)$ for large n. From the sufficiency result for m in [6, Theorem 3], with $\delta' \triangleq 2n^{-\delta}$, $m = 2ek(\log n + \log(2/\delta'))$ is sufficient, which is similar to our PAC bound (3.17) when $\epsilon = 0$ (also see the discussion below (3.27)). Explicitly computing the optimum η as in (3.20a), (3.20b) and (3.20c) instead of using a nominal value $\eta = 1/2$ [6] yields approximately a factor of 2 improvement in the testing rate when $g_\epsilon = 0$.

3.2.2.2 Order-Wise Analysis - Approximate Recovery Case

We first discuss the behavior of η when $g_\epsilon > 0$, for large n and k. When $g_\epsilon > 0$, we have $1/2 \leq g_\epsilon/(g_\epsilon+1) < 1$ and further, dropping $g_\epsilon/(g_\epsilon+1)$ in the denominator of (3.15) only increases the value of C and hence η. Therefore, dropping the $g_\epsilon/(g_\epsilon+1)$ term in the expression for C does not violate the sufficiency of the bound in (3.16). As before, we set $c = 1/2$ and $s = s^*$ and observe that $s^*/(n-k) \ll 1$ holds for large n and therefore, using a Taylor series approximation in the sub-linear regime, the Chernoff parameter, η, can be approximated as

$$\eta \approx \sqrt{\frac{s^* \log\left(\frac{2}{\delta}\right)}{(n-k)\left[\frac{\log\left(\frac{2}{\delta}\right)}{g_\epsilon+1} + \log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}}. \tag{3.21}$$

From (3.21), we observe that η scales as $O(1/\sqrt{n\log n})$ for any fixed $\epsilon \geq 0$, similar to its behavior in the exact recovery scenario. Further, (3.21) reduces to (3.20c) at $\epsilon = 0$.

We now characterize the order-wise behavior of our bound. We start by substituting (3.14) in (3.16), to get

$$m_S = \frac{ek}{(1-\eta)}\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1} + \frac{g_\epsilon}{g_\epsilon+1} + \log\left(\frac{n-k}{g_\epsilon+1}\right)\right], \tag{3.22}$$

where $(n/(n-k))^{s^*} \to e$ from below as n grows large. Using $\eta = (-C + \sqrt{C^2+4C})/2$ from Theorem 3.2, we get

$$\frac{1}{1-\eta} \leq 2 + C. \tag{3.23}$$

Using $\log(1/x) \to 1 - (1/x)$ for $x \to 1$ with $x = (n-k)/n$, we get $s^* \to (n-k)/n$, for large n. Therefore, we use

$$C = \frac{\log\left(\frac{1}{(1-c)\delta}\right)}{k\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1} + \frac{g_\epsilon}{g_\epsilon+1} + \log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}, \tag{3.24}$$

along with (3.23) in (3.22) to get

$$m_S = 2ke\left[\log\left(\frac{n-k}{g_\epsilon+1}\right) + \frac{g_\epsilon}{g_\epsilon+1} + \frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1}\right] + e\log\left(\frac{1}{(1-c)\delta}\right). \tag{3.25}$$

Observe that $g_\epsilon/(g_\epsilon+1) < 1$ and $\log(n-k) < \log n$, when $n > k > 0$. Setting $c = 1/2$ in (3.25), we get

$$m_S = 2ke\left[\log\left(\frac{n}{g_\epsilon{+}1}\right){+}1{+}\log\left(\frac{2}{\delta}\right)\left[\frac{1}{g_\epsilon{+}1}{+}\frac{e}{2k}\right]\right]. \tag{3.26}$$

As mentioned earlier, $(n/(n-k))^{s^*} \to e$ for large n. Also, $(1-x)^{(1/x)-1} \to 1/e$ as $x \to 0$ with $x = k/n$ and $(1-x)^n \to 1 - nx$ for $x < 1$ and $|nx| \ll 1$. Therefore, $g_\epsilon = ke\epsilon$ from (3.12). Substituting for g_ϵ in (3.26), we arrive at

$$m_S = 2ke\left[\log\left(\frac{n}{ke\epsilon{+}1}\right){+}1{+}\log\left(\frac{2}{\delta}\right)\left[\frac{1}{ke\epsilon{+}1}{+}\frac{e}{2k}\right]\right]. \tag{3.27}$$

For a given n, k, and δ, we see that the dependency of our COMP-R bound on $\epsilon > 0$ is $\propto (\log(1/\epsilon) + 1/\epsilon)$, similar to the COMP-B bound. On the other hand, we get $m_S = 2ke[\log n + 1 + (1 + e/2k)\log(2/\delta)]$ for the exact recovery case by setting $\epsilon = 0$, showing that the analysis in this subsection is inclusive of the exact recovery case.

3.2.2.3 Utility of Optimizing the Chernoff Parameter

We now discuss the effectiveness of the approximation to the Chernoff parameter, η, for both $g_\epsilon = 0$ and $g_\epsilon > 0$. To this end, recall the definition of the *testing rate*, $\rho_R \triangleq m_S/n$. Note that ρ_R denotes the sufficient number of tests per item.

Figure 3.2 shows the testing rates, ρ_R over various values of the *log confidence* parameter, $\log(1/\delta)$ for both exact recovery and approximate recovery cases. For illustration, we choose $g_\epsilon = 5$ for our discussion on the approximate recovery case in this subsection. The

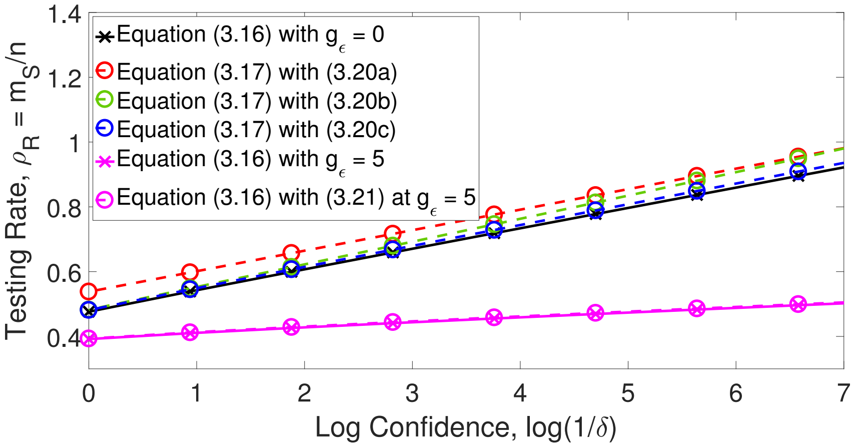


Figure 3.2: Comparison of the sufficiency bound given in (3.16) and (3.17) with $n = 2500$, $k = 50$, $s = s^*$ and $c = 1/2$.

plots are generated with $n = 2500$ and $k = 50$, corresponding to an error tolerance of 10% in the approximate recovery case.

From Figure 3.2, we see that the approximations in (3.20a)–(3.20c) (the red, green, and blue curves, respectively) match well with the exact expressions from Corollary 3.1 (the black curve) under the exact recovery. Similarly, the approximation in (3.21) (the magenta dashed curve) matches well with the exact expression from Corollary 3.1 (the magenta solid curve). A closer observation under the exact recovery condition shows that the approximation in (3.20b) (the green curve) closes in on the exact bound at higher δ where the condition, $\delta \gg 2/(n-k)$, is valid. Further, we observe that the testing rate has almost halved when we allow 5 errors as compared to the exact recovery case at very high confidence, $\log(1/\delta) = 7$. Lastly, the rate of increase of ρ_R with $\log(1/\delta)$ is lower when $g_\epsilon > 0$ compared to when $g_\epsilon = 0$, i.e., the slope of the magenta curve is lower than that of the black curve in Figure 3.2. For example, if $g_\epsilon = 0$, i.e., we wish to guarantee exact recovery, and if we perform $0.6n$ tests, the set output by the algorithm will fail to match the defective set for about 15% of the random test matrices. In contrast, if $g_\epsilon = 5$, with

$0.5n$ tests, a randomly drawn test matrix will fail with probability less than 0.1%. As we will see, such observations hold for the other algorithms also.

3.3 PAC Analysis with False Negative Errors

In this section, we illustrate how PAC analysis can be used to quantify the approximation tolerance when only false negative errors occur. Specifically, we consider the DD algorithm whose defective set estimate satisfies $\hat{\mathcal{K}} \subseteq \mathcal{K}$ [4].

3.3.1 The DD Algorithm: Bernoulli Test Design

The DD algorithm proceeds in two stages: 1) a COMP-like method to eliminate all the items participating in negative tests to obtain a *probable defective set (PDS)* comprising the remaining items, and 2) an item in the PDS is declared a *definite defective* if it is the sole item participating in any positive outcome test, after eliminating the items identified as non-defective in the first stage [4]. Therefore, the DD algorithm never classifies a non-defective item as a defective. However, it may make false negative errors.

We characterize the number of false negatives by counting the number of *unidentified defectives* i.e., the defective items that remain unidentified in the training phase due to their participation in only those group tests which have other (definite) defectives participating in them. Suppose the hypothesis, i.e., the outcome $\hat{x}(\cdot)$ of the DD algorithm, has $D \leq k$ unidentified defectives, and we want the error in the testing phase to be at most ϵ. This requires $D \leq d_\epsilon$, where d_ϵ can be obtained from

$$\begin{aligned}\mathbb{P}_{\mathbf{a}_i \sim \mathcal{B}(p)}\left(\hat{x}(\mathbf{a}_i) \neq x(\mathbf{a}_i)\right) = (1-(1-p)^D)(1-p)^{k-D} \leq \epsilon \\ \Rightarrow d_\epsilon = \left\lfloor \frac{\log(1+\epsilon/(1-p)^k)}{\log(1/(1-p))} \right\rfloor. \qquad (3.28)\end{aligned}$$

Here, the PAC bound in (2.6) reduces to $\mathbb{P}(D > d_\epsilon) \leq \delta$.

We use a similar approach as in [4] till Lemma 3.3 below, with the definition of the success probability modified to accommodate the approximate recovery condition. Think of each group test as a *ball*, and *bin* the balls (group tests) into $k+2$ bins. The first k bins correspond to the k defectives, i.e., a ball will fall into the ith bin, $i = 1, 2, \ldots, k$, if only the ith defective item participates and no other defective item participates in the corresponding group test. The ball will fall into the $(k+1)$th bin, if the corresponding group test has more than one defective item participating. Lastly, the ball will fall into the $(k+2)$th bin, if the outcome of the corresponding test is negative, i.e., when no defectives participate. We construct a vector $\mathbf{B}' = (B_1, B_2, \ldots, B_k, B_+, B_-)$ containing the number of tests of each type that are conducted in the training phase comprising m independent group tests. If the entries of each group test are drawn i.i.d. from $\mathcal{B}(p)$, the probability vector associated with the bins is $\mathbf{q} = (\underbrace{q_1, \ldots, q_1}_{k \text{ bins}}, q_+, q_-)$, [4] with

$$\begin{aligned} q_1 &= \mathbb{P}(\text{Only one defective participates}) = p(1-p)^{k-1}, \\ q_- &= \mathbb{P}(\text{None of the defectives participates}) = (1-p)^k, \\ q_+ &= \mathbb{P}(\text{More than 1 defectives participate}) = 1 - kq_1 - q_-, \end{aligned}$$

and hence $\mathbf{B}'$ follows the multinomial distribution [4]

$$\mathbb{P}_{m;\mathbf{q}}(b_1, \ldots, b_k, b_+, b_-) = \frac{m!}{b_+!\ b_-!\ \prod_{i=1}^{k} b_i!} q_1^{b_1+\cdots+b_k} q_+^{b_+} q_-^{b_-}, \tag{3.29}$$

where $\sum_{i=1}^{k} b_i + b_+ + b_- = m$.

Recall that the PDS is a set containing all the defective items as well as $0 \leq g \leq n-k$ hidden non-defective items. An item participating in a positive test can be declared definite defective if and only if none of the other items in the PDS participate in that test [4]. The probability of the event where all g hidden non-defectives do not participate in a given test is $(1-p)^g$. Using this, we divide ith bin into two (sub-) bins containing L_i and $B_i - L_i$ balls, with corresponding probabilities $(1-p)^g q_1$ and $(1-(1-p)^g)q_1$, respectively, for $i \in [k]$. We call the bin with L_i balls as the ith singleton bin: any ball in this bin corresponds to a group test where only the ith defective item (and no other item in the PDS) participates. Therefore, if the $L_i \geq 1$, we can declare the ith item as a definite defective. Thus, the new bin vector is $\mathbf{B} = (B_+, B_-, L_1, L_2, \ldots, L_k, B_1 - L_1, B_2 - L_2, \ldots, B_k - L_k)$ with the associated probabilities $\mathbf{q} = (q_+, q_-, \underbrace{(1-p)^g q_1, \ldots, (1-p)^g q_1}_{k \text{ bins}}, \underbrace{(1-(1-p)^g)q_1, \ldots, (1-(1-p)^g)q_1}_{k \text{ bins}})$. We now have:

Lemma 3.3. *Consider the DD algorithm run using the outcomes of m tests, with the test matrix drawn from $\mathcal{B}(p)$. Then,*

(a) The probability that there are r negative test outcomes is

$\mathbb{P}(B_- = r) = \binom{m}{r} q_-^r (1-q_-)^{m-r}$.

(b) If G denotes the number of hidden non-defectives in the PDS, we have

$\mathbb{E}[G] = \bar{g} = (n-k)(1-p(1-p)^k)^m$.

(c) Given a set of d unidentified defectives after the second stage, conditioned on $G = g$, we have $\mathbb{P}(\cap_{i=1}^d \{L_i = 0\} | G = g) = (1 - dp(1-p)^{k-1+g})^m$.

We note that the relation in Lemma 3.3(a) gives the marginal distribution of B_-. Further, Lemma 3.3(c) specifies the probability that a given set of d singleton bins are empty, conditioned on there being g hidden non-defectives. We now present a sufficiency bound on m for the DD algorithm:

Theorem 3.3. *A sufficient number of tests such that the predicted outcome based on the estimated defective set does not agree with the true outcome on the future group tests with probability at most ϵ and confidence $1-\delta$ is implicitly given by the value of m_S that satisfies*

$$\binom{k}{d_\epsilon+1}(1-(d_\epsilon+1)p(1-p)^{k-1+\bar{g}+\tilde{g}})^{m_S} \leq \delta, \tag{3.30}$$

where d_ϵ is given by (3.28), $\bar{g} = (n-k)(1-p(1-p)^k)^{m_S}$ and $\tilde{g} \geq 0$ is a parameter.

In the above, $\tilde{g}$ is a parameter that arises in proving the sufficiency condition in (3.30). Due to the complicated form of $\mathbb{P}(G = g)$ in (B.19), it is hard to analytically derive its precise value, but since the last expression in (B.22) is monotonically increasing in $\tilde{g}$, the inequality will be satisfied for some non-negative $\tilde{g}$. In our simulations, we have seen that choosing $\tilde{g} = \lceil \bar{g} \rceil - \bar{g}$ is sufficient to ensure that the inequality (B.22) holds, and hence the m_S obtained from (3.30) is indeed sufficient.

The dependency of ϵ and δ can be seen in (3.30), in addition to n and k. When $\epsilon = 0$, our bound matches the bound in the literature [4]. Lastly, the $k \log n$ dependency is similar to the counting bound [4]. Further discussion is relegated to Section 3.3.1.1.

3.3.1.1 Order-Wise Analysis of (3.30)

We now discuss the behavior of m_S for nonzero ϵ as n and k grow. We use $p = 1/k$ and $\tilde{g} = 1$ in our analysis. First, we observe that,

$$\bar{g} = (n-k)\left(1 - 1/ke\right)^m \rightarrow ne^{-m/ke}, \tag{3.31}$$

where we use $(1 + a/x)^x \rightarrow e^a$ from below for large x with $x = m$ and $a = -m/ke$. Using

$\binom{k}{d_\epsilon+1} \leq (ek/(d_\epsilon+1))^{d_\epsilon+1}$ and (3.31) in (B.23), we get

$$\left(\frac{ek}{d_\epsilon+1}\right)^{d_\epsilon+1}\left(1-(d_\epsilon+1)\frac{1}{k}\left(1-\frac{1}{k}\right)^{k+\bar{g}}\right)^m \leq \delta$$

$$\left(\frac{ek}{d_\epsilon+1}\right)^{d_\epsilon+1}\left(1-\frac{d_\epsilon+1}{ke}\left(1-\frac{1}{k}\right)^{ne^{-m/ke}}\right)^m \leq \delta$$

$$m\log\left(1-\frac{d_\epsilon+1}{ke}\left(1-\frac{1}{k}\right)^{ne^{-m/ke}}\right) \leq \log\left(\delta\left(\frac{d_\epsilon+1}{ke}\right)^{d_\epsilon+1}\right), \tag{3.32}$$

where we use the fact that $(1-1/k)^k \to 1/e$ for large k along with (3.31) in penultimate step. We now observe that $(1-x)^a \approx (1-ax)$ holds when x is small and for fixed a. It also holds when x becomes smaller at a *faster rate* than the growth of a. Therefore, the condition stated above is valid for $m > ke\log n$ with $x = 1/k$ and $a = ne^{-m/ke}$. Further, for $ne^{-m/ke}/k < 1$ to hold, we require $m > ke\log(n/k)$. Since we have $m > ke\log n$, the above condition on m holds too, for $k \geq 1$. Using these conditions in (3.32), we get

$$m\log\left(1-\frac{d_\epsilon+1}{ke}\left(1-\frac{ne^{-m/ke}}{k}\right)\right) \leq \log\left(\delta\left(\frac{d_\epsilon+1}{ke}\right)^{d_\epsilon+1}\right)$$

$$m\left(1-\frac{ne^{-m/ke}}{k}\right) \geq ke\left[\frac{\log\left(\frac{1}{\delta}\right)}{d_\epsilon+1}+\log\left(\frac{ke}{d_\epsilon+1}\right)\right], \tag{3.33}$$

where we use $\log(1-x) \to -x$ for small x with $x = \frac{d_\epsilon+1}{ke}\left(1-\frac{ne^{-m/ke}}{k}\right) \ll 1$ for large n and m in the last step. The solution to the transcendental equation in (3.33) subject to $m > ke\log n$ gives the sufficient number of tests, m_S required for the DD algorithm to succeed with confidence $1-\delta$ when d_ϵ false negative errors are allowed. Define

$$D \triangleq ke\left[\frac{\log\left(\frac{1}{\delta}\right)}{d_\epsilon+1}+\log\left(\frac{ke}{d_\epsilon+1}\right)\right]. \tag{3.34}$$

Note that as m increases, D/m decreases. Using (3.34) in (3.33), we get

$$\begin{aligned} -\frac{m_S}{ke} + \log\left(\frac{n}{k}\right) &\leq \log\left(1 - \frac{D}{m_S}\right) \\ \frac{m_S^2}{ke} - m_S \log\left(\frac{n}{k}\right) - D &\geq 0, \end{aligned} \tag{3.35}$$

where the last step uses $\log(1-x) \to x$ for small x with $x = D/m$. Therefore, we get

$$m_S = \frac{ke}{2}\left[\log\left(\frac{n}{k}\right) + \sqrt{\log^2\left(\frac{n}{k}\right) + \frac{4D}{ke}}\right]. \tag{3.36}$$

Since $m_S > ke \log n$, the solution to m in (3.36) is consistent. Further using $\sqrt{1+x} \approx 1 + x/2$ with $x = 4D/ke\log^2(n/k)$ in (3.36), we get

$$\begin{aligned} m_S &= \frac{ke}{2}\log\left(\frac{n}{k}\right)\left[2 + \frac{2D}{ke\log^2\left(\frac{n}{k}\right)}\right] \\ &= ke\log\left(\frac{n}{k}\right) + \frac{D}{\log\left(\frac{n}{k}\right)}. \end{aligned} \tag{3.37}$$

From (3.28), we see that for $p = 1/k$ and large k, we get $d_\epsilon \approx ke\epsilon$ using $1/(1-x) \approx 1+x$, $(1-p)^k \to 1/e$ and $\log(1+x) \approx x$ for small x. Substituting for D from (3.34) in (3.37) along with $d_\epsilon = ke\epsilon$, we get

$$m_S = ke\left[\log\left(\frac{n}{k}\right) + \frac{\log\left(\frac{1}{\delta}\right)}{(ke\epsilon+1)\log\left(\frac{n}{k}\right)} + \frac{\log\left(\frac{ke}{ke\epsilon+1}\right)}{\log\left(\frac{n}{k}\right)}\right]. \tag{3.38}$$

As in the COMP-B and COMP-R case, we see that the behavior of the DD bound for a fixed n, k and δ is $\propto (\log(1/\epsilon) + 1/\epsilon)$ for any $\epsilon > 0$. Lastly, setting $\epsilon = 0$, we get $m_S = ke\left(\log(n/k) + \frac{1}{\log(n/k)}[\log(1/\delta) + \log k + 1]\right)$ in the exact recovery case. Noting that $1/\log(n/k) < 1$ along with $\log k/\log(n/k) < 1$ for $k \ll n$, we get the following $m_S = ke\left(\log(n/k) + \frac{\log(1/\delta)}{\log(n/k)} + 2\right)$ when $\epsilon = 0$. Using [4, Theorem B.3], we get $m = (\kappa(\gamma') + \delta)ek\log n$ is sufficient, where $\kappa(\gamma') \triangleq \max\{\gamma', 1-\gamma'\}$ with $k = n^{1-\gamma'}$. Using

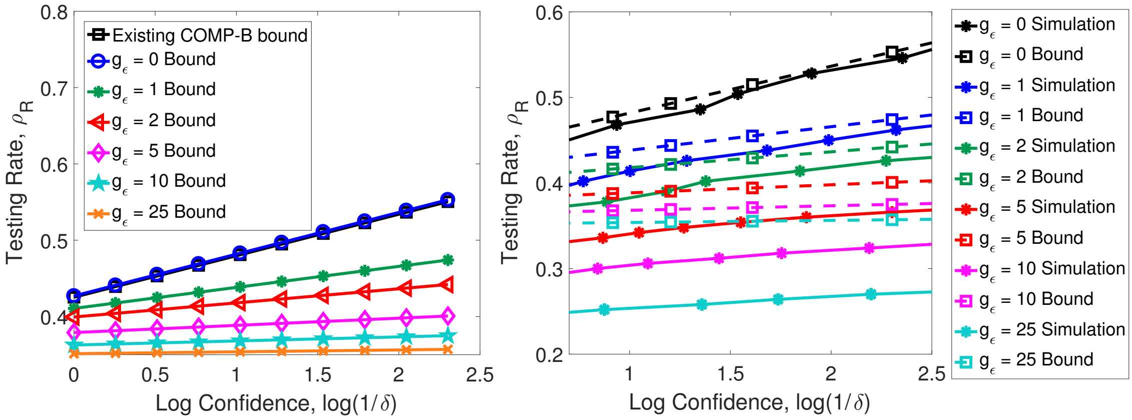


Figure 3.3: (Left) Comparison of the sufficiency bound in [6, Theorem 4] and theoretical PAC bounds (3.3) on the testing rate; (Right) theoretical and simulated testing rates at different error tolerance values, for COMP-B.

$\delta' = n^{-\delta}$, $m = ek(\kappa(\gamma')\log n + \log(1/\delta'))$ is sufficient for exact recovery, which is similar to our bound.

3.4 Simulation Results

3.4.1 Tightness of the Bounds

We simulate exact and approximate set identification scenarios for the COMP-B, COMP-R and DD algorithms[3], and compare them with our theoretical bounds and the existing results [4, 6]. We consider $n = 2500$ items, out of which $k = 50$ are defective. Further, we use Bernoulli parameter $p = 1/k$ for generating the plots with COMP-B and DD algorithms and $s = s^*$ with the weight parameter, $c = 1/2$, in the COMP-R algorithm (see Theorem 3.2). The simulated curves are obtained by averaging over 1,000 Monte Carlo runs.

[3] As mentioned earlier, COMP-B and COMP-R denote the use of Bernoulli and near-constant row-weight test designs with the COMP algorithm. We use the Bernoulli test design with the DD algorithm in our analysis, and therefore, we do not add a separate qualifier to denote the test design.

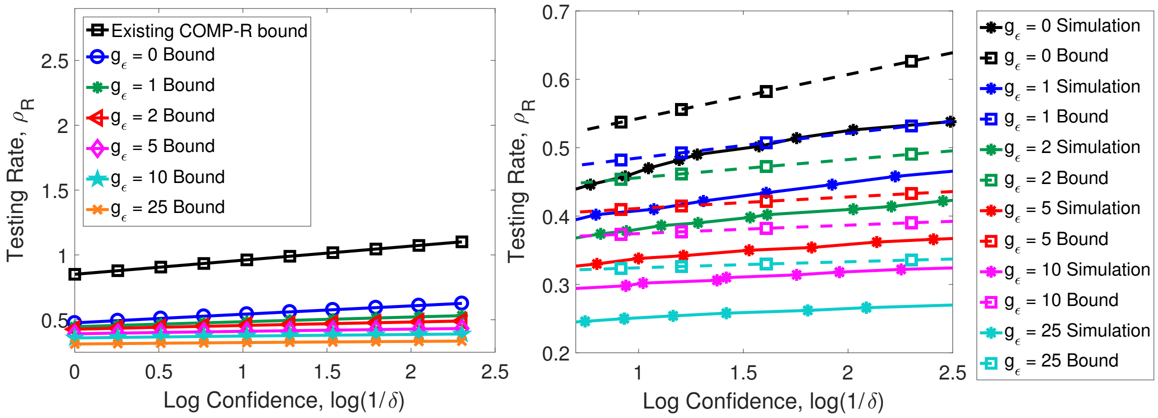


Figure 3.4: (Left) Comparison of the sufficiency bound in [6, Theorem 3] and the theoretical PAC bounds (3.13) on the testing rate; (Right) theoretical and simulated testing rates at different error tolerance values, for COMP-R.

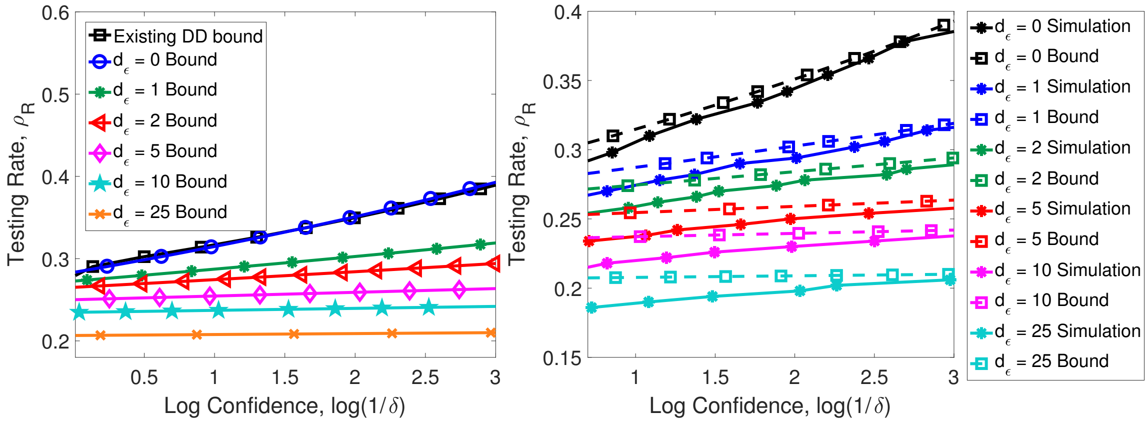


Figure 3.5: (Left) Comparison of the sufficiency bound in [4, Lemma A.8] and the theoretical PAC bounds (3.30) on the testing rate; (Right) theoretical and simulated testing rates at different error tolerance values, for the DD algorithm.

Figure 3.3, Figure 3.4 and Figure 3.5 show the relationship between the testing rate, $\rho_R = m_S/n$ and $\log(1/\delta)$, a notion similar in behavior to the confidence parameter, $1-\delta$, for COMP-B, COMP-R and DD algorithms, respectively. In particular, 1) the *left* subplots illustrate how our PAC bound compare with the existing bounds whereas 2) the *right* subplots compare the theoretical PAC bounds with the simulation curves.

From the left subplots of Figure 3.3 and Figure 3.5, we note that the testing rate obtained from the PAC analysis equations when $\epsilon = 0$ (blue curves) match with the existing upper bounds under exact recovery (black curves) for both COMP-B and DD algorithms. In contrast, from left subplot of Figure 3.4, we note that there is a difference of $\approx 0.4 - 0.5$ in testing rates between the PAC bound when $g_\epsilon = 0$ (blue curve) and the existing sufficiency condition under the exact recovery (black curve). This difference can be attributed to the choice of the Chernoff parameter η in the two cases. The authors in [5] choose $\eta = 1/2$, whereas, we optimize η to obtain the minimum number of tests such that the probability of error is lower than δ. We defer further discussion on optimizing η to Section 3.4.4. We note that the sufficient number of tests is lower when we allow $\epsilon > 0$ in both COMP (COMP-B and COMP-R) and DD algorithms. Also, the right subplots validate the PAC upper bound derived in the paper, and we note that the bound is particularly tight in the case of the DD algorithm (Figure 3.5.)

We see that allowing for a few missed/false positive items in the set output by the algorithm can help significantly reduce the sufficient number of tests needed. As mentioned earlier in Section 2.1.2, allowing for a few false positives (as in the COMP algorithm) can also be useful in applications where the goal is to identify most of the non-defective items [147], since these algorithms do not miss defective items. Similarly, as in the rare antigen identification example mentioned in Section 2.1.2, allowing for a small number of missed defectives is useful when it is important to quickly identify some of the defective items, and we see that by not requiring that *all* the defective items be found, the DD algorithm can significantly reduce the number of group tests that need to be conducted, while retaining a high confidence in the outcome.

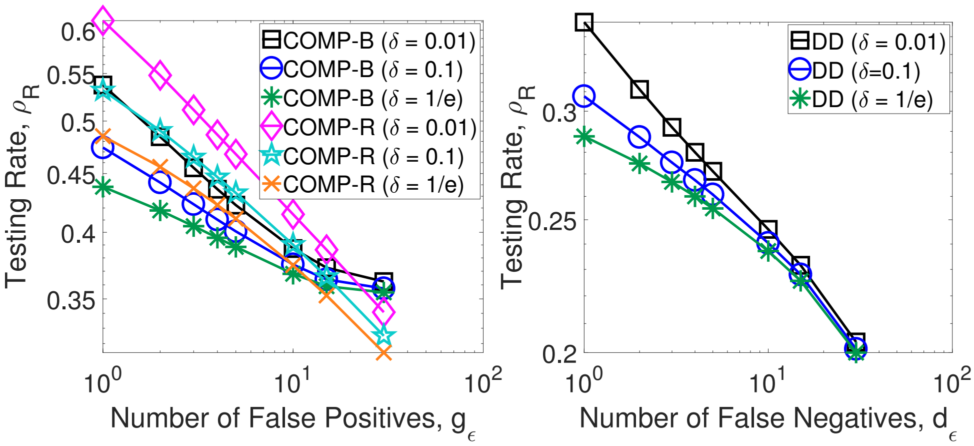


Figure 3.6: The sufficiency bound for COMP-B and COMP-R (left subplot) and DD (right subplot) as the number of errors allowed varies, for different values of δ.

Lastly, we observe from the right subplots of Figure 3.3 and Figure 3.4 that the COMP-B bound is tight for exact recovery whereas the COMP-R bound follows the slope of the simulated curves slightly better at higher g_ϵ and lower confidence.

3.4.2 Effect of Approximation Error Tolerance on the Bounds

In this subsection, we discuss the achievability bounds for COMP-B, COMP-R, and DD, as a function of the number of errors allowed. Figure 3.6 shows how the testing rate, $\rho_R = m_S/n$, of COMP-B and COMP-R (left subplot) and DD (right subplot) varies as the number of FPs and FNs, respectively, for different values of the parameter δ. In the plots, the m_S is computed by setting $n = 2500$, $k = 50$ using Theorem 3.1, Theorem 3.2, and Theorem 3.3 for COMP-B with $p = 1/k$, COMP-R with $s = s^*$, and DD with $p = 1/k$, respectively.

From Figure 3.6, we see that the testing rate, ρ_R, (and thus the sufficiency bound on m) is proportional to $(\log(1/\tau) + 1/\tau)$, where $\tau = g_\epsilon + 1$ in the COMP-B and COMP-R cases, and $\tau = d_\epsilon + 1$ in the case of DD. From the left subplot, we see that although the testing rate for COMP-R as computed from our bounds for $g_\epsilon = 0$ is higher than for COMP-B, the slope for COMP-R is steeper as g_ϵ increases. Therefore, the testing rate obtained from the COMP-R bound at, say, $g_\epsilon = \{25, 30\}$ is lower than that obtained from the COMP-B bound. For example, at $g_\epsilon = 30$, the testing rate of COMP-B is $\rho_R \approx 0.3574$ and that of COMP-R is $\rho_R \approx 0.325$ at a $\delta = 0.1$. The difference seen can be attributed to the analysis procedure and the way the test matrix is constructed. A similar observation can be made from the right subplots of Figure 3.3 and Figure 3.4. From the right subplots of Figure 3.3 and Figure 3.4, we see that the COMP-B bound matches very well with the simulation curve at $g_\epsilon = 0$ whereas the COMP-R bound is relatively loose. On the other hand, the COMP-B bound is looser than the COMP-R bound relative to their respective simulation curves at $g_\epsilon = 25$. Thus, the analysis used in deriving the COMP-B bound is useful for low approximation error scenarios, whereas that used in the COMP-R is useful when we allow for higher approximation error. This can be attributed to the use of the union-bound argument (see (B.2)). The slope of the factor $\binom{n-k}{g_\epsilon+1}$ is bounded between $[(c_l/x)^x(\log(c_l/x) - 1), (c_u/x)^x(\log(c_u/x) - 1)$ where $x = g_\epsilon + 1$, $c_u = e(n-k)$ and $c_l = n - k$. That is, the slope increases with g_ϵ, making the COMP-B bound looser relative to the COMP-R bound at higher g_ϵ when $g_\epsilon + 1 \ll (n-k)/2$.

Further, from the right subplot of Figure 3.6, the testing rate of the DD algorithm is lower than that of COMP-B and COMP-R algorithms, due to the dependency of m_S on $\kappa(\gamma') \log n = \log k$ in the former case as against a dependency on $\log n$ in COMP-B/COMP-R.

3.4.3 Effect of Performing an Insufficient Number of Tests

In this subsection, we present the utility of our bounds from an alternative viewpoint:

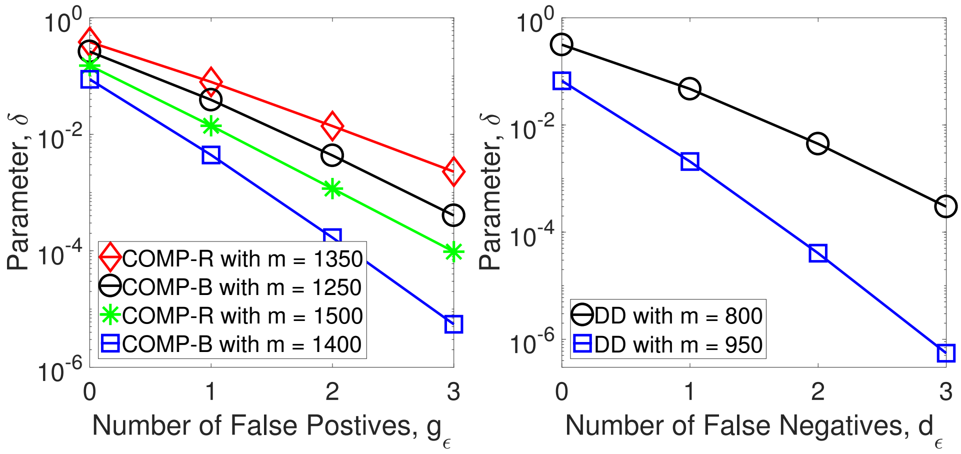


Figure 3.7: Illustration of the trade-off between the confidence parameter and the approximation error tolerance for COMP-B, COMP-R and DD algorithms.

what guarantees can be provided for a given n and k, if the number of group tests performed is insufficient to guarantee exact recovery with high confidence? To illustrate this, in Figure 3.7, we plot the parameter, δ, as a function of g_ϵ for COMP-B and COMP-R algorithms, and as a function of d_ϵ for DD algorithm, at two different values of m for each algorithm. The plot is generated by setting $n = 2500$, $k = 50$, $p = 1/k$, $c = 1/2$, $s = s^*$ in Theorem 3.1, Theorem 3.2 and Theorem 3.3 for the purpose of discussion. However, similar observations can be made across different values of n, k, m etc.

From Figure 3.7, we see that allowing a small number of false positive/negative errors allows one to obtain significantly higher confidence, i.e., a lower δ, for a given number of tests. For example, to ensure a confidence of $\approx 91\%$, with the COMP-B algorithm, 1400 tests are sufficient as per (3.3) with $n = 2500$, $k = 50$ and $p = 1/k$ for exact recovery. However, if only 1250 tests could be conducted, one can provide a confidence of $\approx 73\%$ for exact recovery, while tolerating a single false positive error yields a confidence of $\approx 96\%$ ($> 91\%$). If two false positives are allowed, the confidence goes well above 99%. Similar conclusions can be drawn from the COMP-R and DD algorithms.

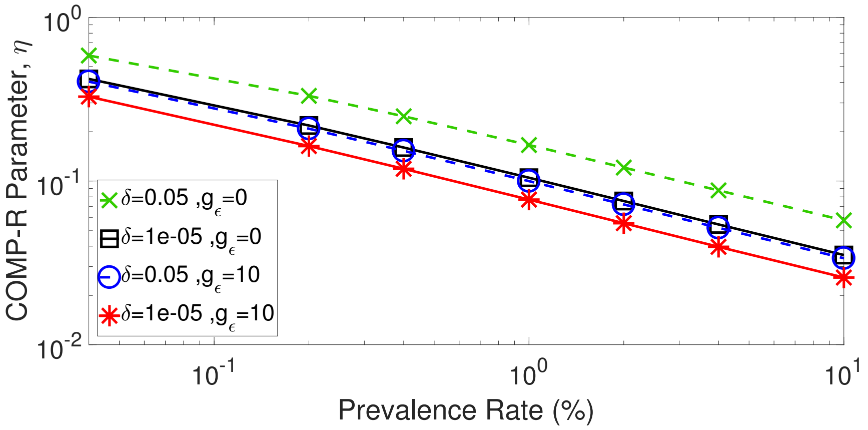


Figure 3.8: The COMP-R parameter, η, computed at different g_ϵ and δ with $n = 2500$, $c = 1/2$, $s = s^*$, as the prevalence rate (%) is varied.

This example illustrates that the PAC bounds can provide a range of guarantees when the number of group tests performed is insufficient to guarantee exact recovery with high confidence. One can either tolerate a small number of errors or choose to operate at lower confidence levels.

3.4.4 Effect of Optimizing η in COMP-R Analysis

In this subsection, we present empirical evidence that optimizing the Chernoff parameter η yields tighter bounds on the number of tests. For this experiment, we set $n = 2500$, $s = s^*$ and $c = 1/2$ in the COMP-R algorithm (see Theorem 3.2). From (3.13), it is clear that m_S decreases as η is decreased. At the same time, choosing an η such that (B.10) is upper bounded by δ ensures that the sufficiency condition is satisfied. We illustrate the effect of varying the prevalence rate (%), i.e., $100 \times k/n$ at $\delta \in \{10^{-5}, 0.05\}$ and $g_\epsilon \in \{0, 10\}$ in Figure 3.8. We see that η computed using (B.13) decreases as the prevalence rate increases irrespective of the choice of δ and g_ϵ. For a fixed δ, the value of η is higher under exact recovery (i.e., when $g_\epsilon = 0$) to that when $g_\epsilon = 10$, and, consequently, the sufficient

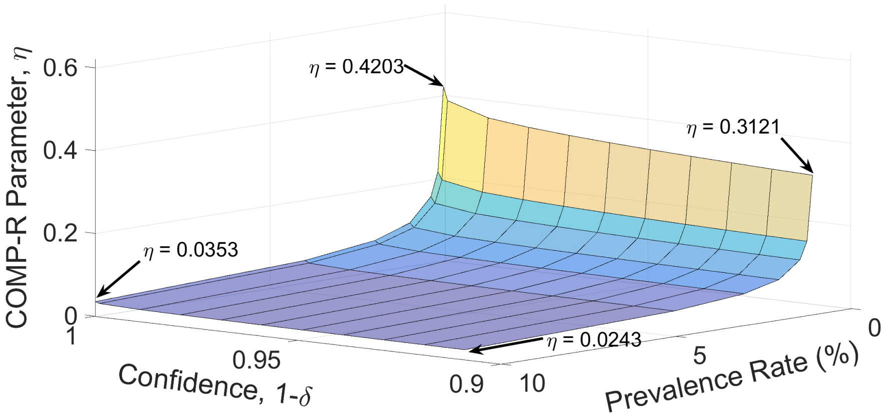


Figure 3.9: The surface of the COMP-R parameter, η, computed at $g_\epsilon = 0$ with $n = 2500$, $c = 1/2$, $s = s^*$, as the prevalence rate (%) and δ are varied.

number of tests is lower in the latter case. For a fixed g_ϵ, as the confidence, $1-\delta$, decreases, η decreases, thereby lowering the sufficient number of tests. The gap between the values of η is higher across the above δ and g_ϵ values when the prevalence rate is lower, in fact, close to 0%, and as we move towards right in Figure 3.8, the gap decreases.

Next, in Figure 3.9, we show the effect of varying δ and the prevalence rate using a surface plot of η, for $g_\epsilon = 0$. When the prevalence rate is very low, $\eta = 1/2$ is optimum. Otherwise, the optimum value of η is much lower. Hence, computing η using (B.13) leads to a tighter sufficiency bound, an improvement by a factor of 2 (see left subplot of Figure 3.4).

We conclude our discussion on the parameter η by empirically showing its variation as n varies in Figure 3.10. The optimum value of η decreases as n increases. The variation across the prevalence rate remains similar to the observations made earlier. Therefore, in conclusion, the computation of the optimum Chernoff parameter, η, can significantly improve the sufficiency bound on m.

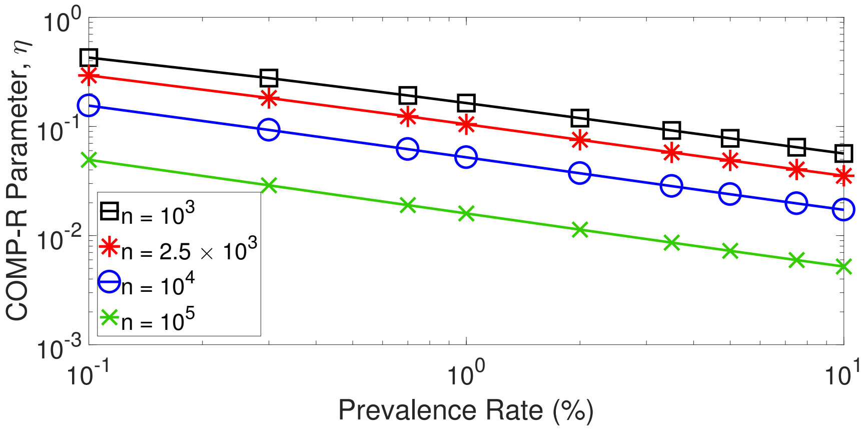


Figure 3.10: The COMP-R parameter, η, as the prevalence rate (%) is varied, computed for different values of n, with $g_\epsilon = 0$, $\delta = 10^{-5}$, $c = 1/2$, $s = s^*$.

3.4.5 Numerical Analysis of the DD Tuning Parameter, $\tilde{g}$

We present an alternate view of Theorem 3.3, where the sufficient number of tests for DD algorithm to succeed with confidence $1 - \delta$ when ϵ approximation errors are allowed is given implicitly by (3.30). We set $p = 1/k$ and relate ϵ to the allowed number of false negative errors, d_ϵ, by (3.28). We start by posing the problem of solving for the sufficient number of tests in Theorem 3.3 as an optimization problem:

$$\begin{aligned} m_S, \tilde{g} = & \underset{m\in\mathbb{Z}_+,\ \tilde{g}'\in(0,\infty)}{\arg\min} \quad m \\ & \text{s.t.} \sum_{g=0}^{n-k} (1-(d_\epsilon+1)p(1-p)^{k-1+g})^m \mathbb{P}(G=g) \le (1-(d_\epsilon+1)p(1-p)^{k-1+\bar{g}+\tilde{g}'})^m \\ & \text{and} \binom{k}{d_\epsilon+1}(1-(d_\epsilon+1)p(1-p)^{k-1+\bar{g}+\tilde{g}'})^m \le \delta, \end{aligned} \tag{3.39}$$

where $p = 1/k$, $d_\epsilon < k$ and $\mathbb{P}(G = g)$ is given by (B.19). We solve the optimization problem in (3.39) using a grid-search over a suitable range of m and $\tilde{g}'$ to ensure that we are operating in the feasible range.

Figure 3.11 shows a scatter plot of the optimum $\tilde{g}$ obtained by solving (3.39) with $p = 1/k$

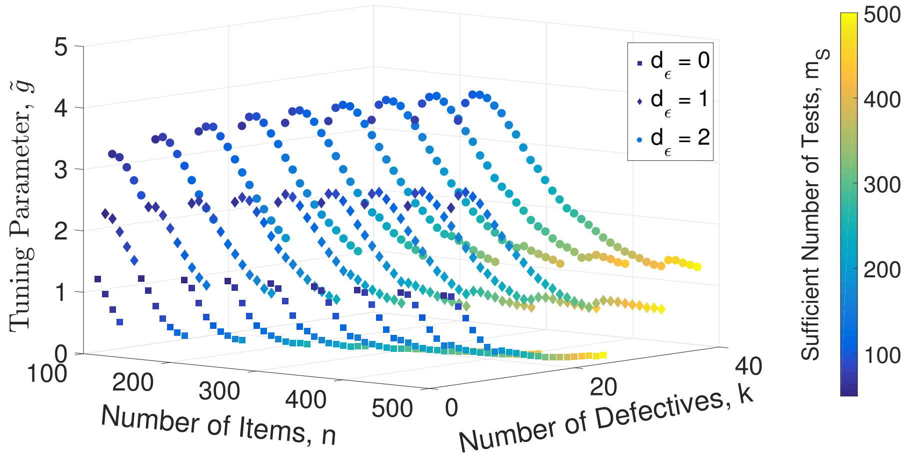


Figure 3.11: Solving for $\tilde{g}$ numerically by varying n from 100 to 500 in steps of 50 and suitable values of k where the DD algorithm succeeds with $\delta = 0.01$ and $d_\epsilon \in \{0, 1, 2\}$.

by varying the population size, n, in range $[100, 500]$ in steps of 50 with suitable values of k, $\delta = 0.01$ and $d_\epsilon \in \{0, 1, 2\}$. The color bar in Figure 3.11 shows the sufficient number of tests, m_S, obtained by solving the optimization problem. We observe that $\tilde{g}$ increases with d_ϵ since the values of g at which $\mathbb{P}(G = g)$ becomes negligible shifts to the right. Further, as k increases, $\tilde{g}$ shows as slight increase, achieving a peak value at $k \approx d_\epsilon + 4$ and then, tapering off as $k^{\alpha-1}e^{-\beta k}\beta^\alpha$ with $\alpha \approx 1.667$ and $\beta \in [1/5, 1/3]$. Finally, $\tilde{g}$ varies as $\log^2 n$ for a fixed k.

3.4.6 Testing Rate Surface and Sufficient Tests Contours

We now introduce the notions of the *testing rate surface* and *sufficient tests contours*, which allow better visualization of the trade-off between the error margin and the confidence parameter. These are illustrated in the left (testing rate surface) and right (sufficient tests contour) subplots of Figure 3.12, Figure 3.13 and Figure 3.14 for COMP-B, COMP-R, and DD algorithms, respectively. Specifically, the testing rate surface shows the sufficient testing rate, ρ_R, as a function of the error tolerance g_ϵ (or d_ϵ) and confidence $1 - \delta$. The

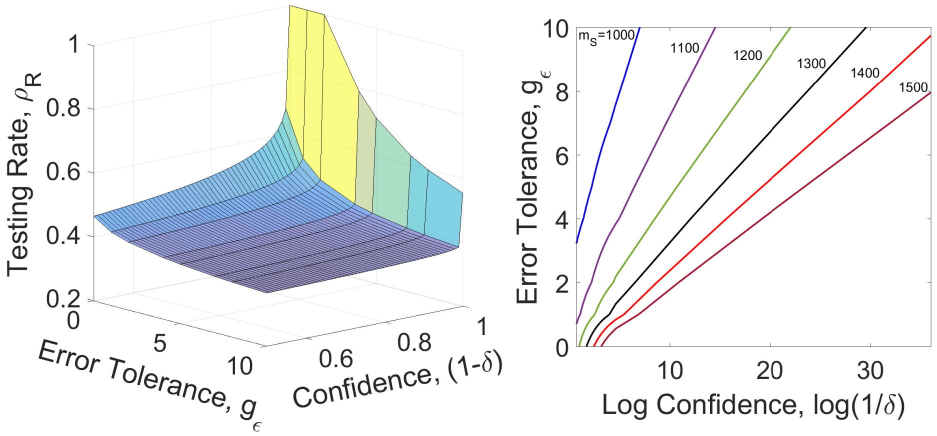


Figure 3.12: (Left) The sufficient testing rate *surface* and (Right) sufficient number of tests *contours* vs. the confidence parameter, $1 - \delta$, and error tolerance, g_ϵ with $n = 2500$, $k = 50$ and $p = 1/k$ for COMP-B.

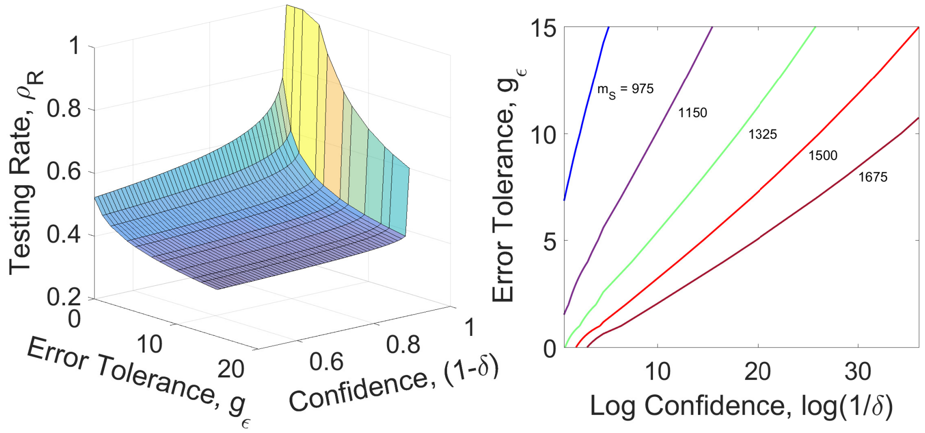


Figure 3.13: (Left) The sufficient testing rate *surface* and (Right) sufficient number of tests *contours* vs. the confidence parameter, $1 - \delta$, and error tolerance, g_ϵ with $n = 2500$, $k = 50$, $c = 1/2$ and $s = s^*$ for COMP-R.

testing rate contours are plotted over log confidence and mark the boundary over which a given number of tests are sufficient. For example, referring to Figure 3.12, all error tolerance and confidence values to the right and under the blue curve are achievable when 1,000 group tests are used. From the left subplots, we see that allowing for a nonzero error tolerance allows us to reach a high confidence level without significantly increasing the number of tests. On the other hand, if exact recovery is required, the number of tests

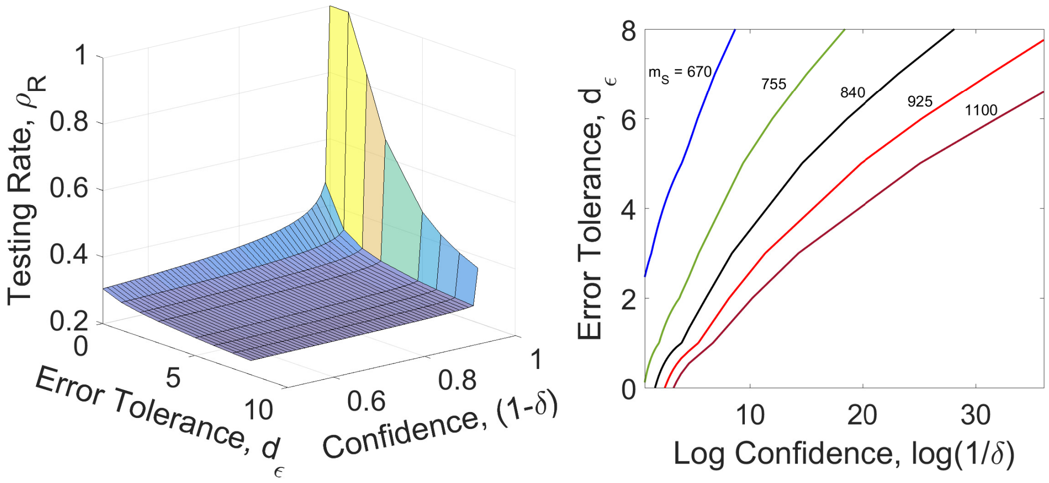


Figure 3.14: (Left) The sufficient testing rate *surface* and (Right) sufficient number of tests *contours* vs. the confidence parameter, $1 - \delta$, and error tolerance, d_ϵ with $n = 2500$, $k = 50$ and $p = 1/k$ for DD.

rapidly increases as the confidence approaches one. Similar observations on the trade-off between the error tolerance and confidence parameter can be made from the right subplots also.

Comparing the testing rate surfaces of COMP-B, COMP-R, and DD algorithms, we note that although the geometry of the surfaces is visually similar, the testing rate offered by the DD algorithm is better by a factor ≈ 1.5 compared to that of COMP-B and by a factor ≈ 1.6 compared to that of COMP-R when the confidence level is close to 1, across various error tolerances ϵ. In other words, at a given confidence level, allowing for a small number of missed defectives leads to a larger reduction the number of group tests compared to allowing for a small number of false positives. Thus, at high confidence (≈ 1.0) and low ϵ, the COMP-R algorithm requires the highest number of tests at 1675, followed closely by COMP-B at 1500 and then by a larger margin, by the DD algorithm at 1100.

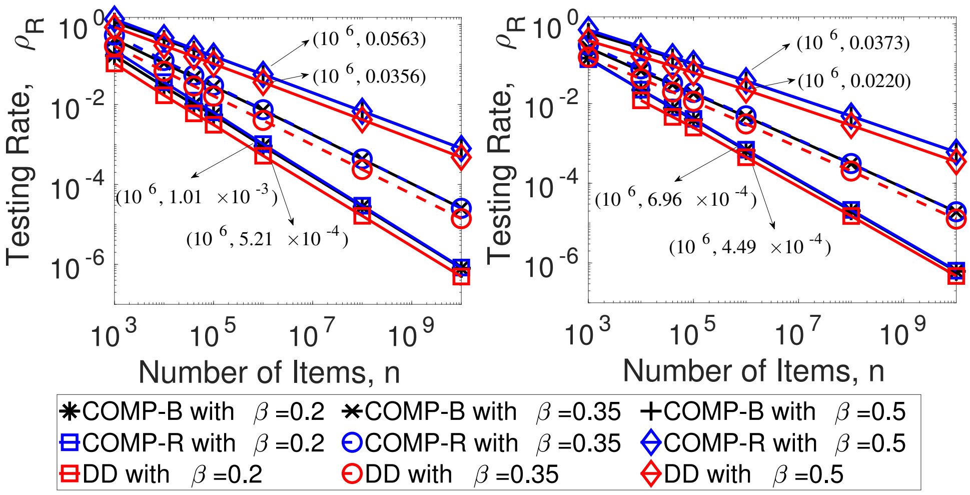


Figure 3.15: Sufficient testing rate vs. population size, n, across various inverse-sparsity parameter β with $k = 0.95n^{\beta}$, $s = s^*$, $c = 1/2$, $p = 1/k$ and $\delta = 10^{-3}$ for COMP-B, COMP-R and DD algorithm in exact recovery case in the left subplot and approximate recovery case ($g_\epsilon = d_\epsilon = 5$) in the right subplot.

3.4.7 Testing Rate vs. Population Size

In the sub-linear regime, we choose $k = \Theta(n^{\beta})$, where $\beta \in (0, 1)$ is called the *inverse-sparsity* parameter, because the number of defective items increases (i.e., the item vector $\mathbf{x}$ becomes less sparse) as β increases. In Figure 3.15, we compare the testing rate as n increases across various values of $\beta \in \{0.2, 0.35, 0.5\}$ at high confidence, i.e., when $\delta = 10^{-3}$. The left subplot in the figure shows the testing rate under exact recovery. The right subplot shows the behavior for approximate recovery, i.e., with $g_\epsilon = d_\epsilon = 5$.

From the left subplot of Figure 3.15, we see that, for exact recovery, a testing rate of ≈ 0.0563 and 1.01×10^{-3} for COMP (i.e., for both COMP-B and COMP-R) and a testing rate of ≈ 0.0356 and 5.21×10^{-4} for DD at $\beta = 0.5$ and 0.2, respectively, are sufficient when the population size is $n = 10^6$. Similarly, from the right subplot, we see that the testing rates relax to 0.0373 and 6.96×10^{-4} for COMP and to 0.022 and 4.49×10^{-4} for DD at $\beta = 0.5$ and 0.2, respectively, at the same population size when we allow for 5 errors. In summary, the testing rates relax by $\approx 33\%$ and 31% for COMP and by $\approx 38\%$ and 13% for DD given $n = 10^6$ when we allow for 5 errors at $\beta = 0.5$ and 0.2, respectively. The lower percentage change in the case of the DD algorithm at $\beta = 0.2$ can be attributed to the dependency of the sufficient number of tests on $\log \binom{k}{d_\epsilon+1}$ in DD vs. that on $\log \binom{n-k}{g_\epsilon+1}$ in COMP.

As β increases, the testing rates increases in all the three cases. More specifically, we have seen in Section 3.2.1 that $O(k \log n)$ tests are sufficient for exact recovery for large n. With $k = \Theta(n^\beta)$, we get the following: $\log \rho_R = O((\beta - 1)\log n + \log(\log n))$. Therefore, $\log \rho_R$ is approximately linearly decreasing with $\log n$ with slope $\beta - 1$ for large $\log n$, which matches with the exact bound shown in the plot. From the curves in the right subplot, we see that our observation on the slope holds true even for the approximate recovery case. Finally, the DD algorithm performs the best in terms of the sufficient number of tests required, across all β's. While this is known for the exact recovery case [4,7], we see from the right subplot that a similar observation holds for the approximate recovery case also.

3.5 Chapter Summary

In this chapter, we used the PAC analysis to derive a sufficiency bound on the number of tests needed for both exact and approximate defective set recovery for Boolean group testing algorithms, namely, COMP (under Bernoulli and near-constant row-weight designs denoted by COMP-B and COMP-R, respectively) and DD (under Bernoulli design). The PAC-learning-based bounds are a function of the number of items, n, the number of defective items, k, a confidence parameter, δ, and an approximation error tolerance, ϵ. This analysis enabled us to characterize a lower bound on the cumulative distribution of the approximation errors. We showed that the PAC-based bounds reduce to known sufficiency bounds in the case of exact recovery when we set $\epsilon = 0$.

Also, we related the number of false positives, g_ϵ (similarly, the number of false negatives, d_ϵ) to the approximation error probability ϵ used in the PAC formulation. In contrast to existing works, we optimized the design parameter, η, to get a tighter COMP-R bound. Further, the classical CCP results do not apply to approximate recovery. Therefore, we extended the analysis of the coupon collector problem to handle *collection of only a subset of coupons* and applied the new results to derive a sufficiency bound for COMP-R. Further, order-wise behavior of the PAC bounds for large n and k were derived.

Towards the end of the chapter, we presented numerical results illustrating the bounds vis-à-vis the desired confidence level for both exact and approximate recovery cases. We demonstrated the advantage of the PAC formulation by empirically illustrating its ability to quantify the sufficient number of group tests across various confidence levels and error tolerances. Lastly, we observed that the PAC-based analysis allows us to trade off the *accuracy* of defective set recovery with the *confidence* with which the decoded set meets that accuracy. A visualization of this trade-off in the form of a sufficient number of tests *contour/surface*, which shows its dependence on the approximation error tolerance and the probability of failure to meet the required error tolerance, was presented.

4 Recovery Algorithms for Covid-19 Group Testing

Chapter Highlights

We now change gears from the theoretical analysis discussed in the previous chapters and focus on developing novel recovery algorithms for group testing. In particular, we present novel recovery algorithms for detecting COrona VIrus Disease (Covid-19) infected samples using group testing. The detection of Covid-19-infected samples involves checking for the presence of a virus called severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) using the reverse transcriptase (quantitative) polymerase chain reaction (RT-qPCR) test procedure. We start this chapter with a brief overview of the RT-qPCR process, including the role of the *cycle threshold* (CT) and the *PCR efficiency factor*, q. Then, we describe a system model for a typical RT-qPCR test process and extend the model to a pooled testing setup.

We then recap two classical group testing algorithms: 1) Combinatorial Orthogonal Matching Pursuit (COMP) and 2) Definite Defectives (DD) as applied to the Covid-19 detection using group testing with *binary test outcomes*. COMP only makes FP errors, and DD only makes FN errors. In order to control both these errors, we present new, iterative algorithms: 1) Gradient Descent - CT (GD-CT) and 2) Iterative Mirrored Hard-Thresholding - CT (IMHT-CT) for detecting the infected samples and in addition, estimating their CT values using the *quantitative group test outcomes*. The convergence results for the GT-CT algorithm show that the iterates, i.e., the estimates of the individual sample CTs, converge in probability to a local minimizer. Numerical simulations comparing the normalized mean squared error (NMSE), false positive rate (FPR) and false negative rate (FNR) performances of GD-CT and IMHT-CT algorithms are presented. In terms of NMSE and FPR, the GD-CT algorithm outperforms the IMHT-CT algorithm. The performance is similar in terms of the FNR.

4.1 Introduction

COrona VIrus Disease (Covid-19), which originated in late 2019 as a local infection, turned into a pandemic of unprecedented levels. The disease spread rapidly and globally from the place of its origin. The mortality rate of Covid-19 is statistically low ($\sim 0.5 - 4\%$) [148]. However, the large caseload in small geographical regions destabilized the healthcare systems in many countries during the year 2020-2021 and, in certain cases, extended till 2022. The disease entered the endemic stage in 2023. In a significant number of cases (guesses vary widely $\sim 50 - 85\%$), the disease manifested itself with mild or no symptoms [149–153]. However, such individuals (silent spreaders) could still spread the disease [154, 155]. During large public gatherings, the presence of such individuals can and has in the past resulted in massive super-spreader events. Thus, it became a major concern for opening up campuses, office spaces, and other public amenities. Rapid testing, contact tracing, and isolation of the infected have been the only known effective way to control the spread of the disease [156].

Various vaccines have come out on the market in the past few years. However, the virus mutated rapidly, with each strain having a different combination of lethality and transmissibility (R_0) [157]. As a result, most parts of the world witnessed at least three waves of the pandemic, varying in severity. New strains and local waves have appeared even in May 2024.[1] Furthermore, with rapid globalization and faster methods to travel in place, a new, local infection of this kind could be a significant concern in the future. A few notable examples as provided by the World Health Organization (WHO) for the diseases

[1] FiLRT variant detected in Singapore: https://www.travelweekly-asia.com/Destination-Travel/FLiRT-alert-for-Singapore-as-Covid-cases-rise.

with epidemic potential where there are no or insufficient counter-measures include Zika, Nipa, Lassa fever, to name a few.[2] Primary prevention steps should be in place to curb such outbreaks. Therefore, testing protocols which are systematic, cost-effective, reliable, and repeatable that can aid effective contact tracing have not lost their importance [158]. In this chapter, we are interested in algorithms for detecting infected samples from a population using group testing. In particular, we focus on Covid-19 disease detection. However, as we shall see below, the testing method used to construct our system model is used in various other infectious disease detection. Therefore, our algorithms can be potentially applied in various scenarios. Among the various methods that have been developed to detect the Covid-19 virus in an individual [159], one of the most reliable testing methods is the RT-qPCR, which is described next.

4.1.1 The RT-qPCR Process

The RT-qPCR process [159, 160] is a type of nucleic acid amplification test. A biological (e.g., naso-oropharyngeal swab) sample is collected from an individual and viral RNA molecules, if any, are extracted via a pre-test preparation process. The RNA molecules are converted to complementary DNA (cDNA) molecules using the RT process. The PCR process, which is a sequence of exponential amplification cycles with heating and cooling phases, is conducted next. Identical copies of the target DNA are obtained in each cycle, roughly doubling the initial population. The rate at which viral loads replicate is often called the PCR (amplification) *efficiency factor,* denoted by q.

A *Taqman* probe is added, which contains fluorophores that emit light upon excitation.

[2] WHO Listed Diseases with Emergency R&D Priority: https://www.who.int/activities/prioritizing-diseases-for-research-and-development-in-emergency-contexts

The intensity of light emitted is proportional to the number of viral DNA strands present in the sample at any given time. The cycle at which the fluorescent light intensity exceeds a preset threshold, denoted by τ, is called the *cycle threshold* (CT) value and is the quantitative output of the RT-qPCR test.

The CT usually takes values between $15 - 35$ for positive Covid-19 tests [161]. It can also be noted that a higher initial viral load implies that the preset threshold is crossed in the earlier cycles in the PCR process, i.e., CT is low. Also, zero viral loads in the sample are depicted by CT $= \infty$, to say that the preset threshold is not crossed in a finite time.

4.1.2 Motivation

The major non-renewable components of the RT-qPCR test are the reagents like the Taqman probes, primers, and time-to-test itself. The overall RT-qPCR test procedure typically takes 3 to 8 hours [162]. A standard PCR plate can accommodate roughly 93 or 381 individual samples depending upon the plate layout, after reserving space for positive and negative controls (PC and NCs).[3] Scaling horizontally by procuring more testing kits is often not feasible due to the high procurement and the operational costs involved. Therefore, vertically scaling, i.e., making the testing process more efficient in effective time-taken per sample and resource usage is crucial in aiding large-scale testing reliably and repetitively (e.g., daily).

Vertical scaling of the test process can be accomplished by pool testing [3] (a.k.a. group testing). In recent decades, group testing has converged to the area of compressed sensing (CS), and is often called Boolean compressed sensing [163].

[3] PCR Plastics - PCR Plate Configurations: https://www.thermofisher.com/in/en/home/life-science/cloning/cloning-learning-center/invitrogen-school-of-molecular-biology/pcr-education/pcr-qpcr-plastics/pcr-qpcr-plastics-considerations.html

4.1.3 Related Work

We recall that the idea of pool testing, first proposed by Dorfman [1] in 1943 was used to screen prospective entrants into the US military for Syphilis. Pool testing has a long history, and has been applied in various settings, including in nucleic acid amplification tests like PCR (see [58] and the references therein) and also for Covid-19 detection [60].

As mentioned in Section 1.2, the Dorfman-style testing methods are called *adaptive* pool testing [3]. In contrast, in non-adaptive pool testing, all the required tests are performed in a single stage, followed by an application of a suitable decoding algorithm [4, 5, 164] to recover the individual sample status given the pooled test outcomes and a *pooling matrix*. Each RT-PCR test takes several hours to run, and at the same time, a standard RT-PCR plate can accommodate either 93 or 381 samples depending on the plate layout. An adaptive test, which is a multi-stage procedure, would require 2x or more time than a single-stage non-adaptive group test. Secondly, pooled sample preparation before each stage of multi-stage adaptive testing would expose the technician to bio-hazards for longer duration. Lastly, using finite size, deterministic non-adaptive pooling matrices is practical, especially with the advent of pooling robots [88]. Therefore, non-adaptive pooling methods offer time-advantage, are practitioner-friendly, and are safer for infectious disease testing than the adaptive testing methods. A binary pooling matrix specifies which individuals participate in which test.

The authors in [163] show that the estimation of the n length individual status vector is feasible with an arbitrarily small probability of error using m pooled tests as long as $m \geq O(k \log n)$, where k denotes the number of sick individuals. However, this is an asymptotic bound, i.e., it is valid as $n \to \infty$, with k growing sub-linearly with n [163].

Recently, testing of pooled samples was shown to be feasible for the detection of Covid-19 using RT-qPCR [59]. The authors in [61] optimize pool size and test protocol for Covid-19 detection via RT-qPCR tests and conclude that the limit of detection is $1-3$ RNA copies per μl. An algorithm for classifying each individual as having *{no, low, medium, high}* infection level was developed in [62]. However, in practice, one may be interested in recovering the actual viral load of the individuals, rather than coarse classification.

The performance of binary pool testing algorithms like COMP and its noisy version, Noisy-COMP (NCOMP) was studied in [63]. In the current chapter, we use COMP as a pre-processing step to filter out negative tests with CT $= \infty$ and reduce the problem dimension. The authors in [165] use the non-negative least absolute deviation (NN-LAD) algorithm for decoding in a non-adaptive pool testing setup. The authors in [64] and [66] consider improving test reliability and throughput via possibly non-binary pooling matrices. Although these papers use sparse recovery techniques to infer the individual viral loads, the system model in these papers is formulated as linear observations corrupted by additive noise, which does not match with well-accepted RT-qPCR models [166].

In [65], a weighted least-squares (LS) approach is used to solve the non-linear CS problem $\mathbf{y} = f(\mathbf{A}\mathbf{x})$, where $\mathbf{A}$ is the test matrix, $f(\cdot)$ is the non-linearity due to the amplification and interpolation operation relating the CT values, $\mathbf{y}$, to the sample viral loads, $\mathbf{x}$. For lab-experiments, they model the sensing matrix as $\mathbf{A} = \mathbf{P} \odot \mathbf{W}$, where $\mathbf{P}$ is a binary-valued participation matrix and $\mathbf{W}$ is a positive real-valued sample allocation matrix. The (i, j)th element of the allocation matrix determines the fraction of the jth sample participating in the ith test. We note that the sensing matrix is no longer binary-valued; instead, it has either 0 or positive real entries.

Despite the large body of work on group testing for Covid-19, the existing methods available in the literature are primarily Boolean group testing-based or compressed sensing-based approaches, which assume a linear system model. As we shall see in Section 4.5 and Section 5.4, we show that our recovery algorithms outperform both Boolean group testing and compressed sensing-based methods. Further, an algorithm inspired from the area of *non-linear compressed sensing techniques* (see Section 4.4.2) and applied to our non-linear RT-qPCR model performs worse than the proposed gradient descent-based algorithm (see Section 4.4.1). In particular, due to the exponential nature of the RT-qPCR process (see Section 4.2 for more details), linearized RT-qPCR model-based methods are inaccurate. Further, we show in Section 5.4.4 that the performance of our algorithms is superior to that of the state-of-the-art compressed sensing-based methods available at the time our paper on the topic was published (see Section 1.6), under a noisy measurement model and when the PCR efficiency factor is unknown (see Table 5.2 and Table 5.3). Also, our algorithms perform better than the Boolean group testing algorithms (see Section 4.5.2, Section 4.5.3, Section 5.4.2 and Section 5.4.3).

In addition to the binary and CS-based quantitative methods for group testing, there are approaches which do not fall strictly into either category, and therefore require a separate mention. For instance, the authors in [167, 168] propose to use tropical arithmetic and formulate an adaptive group testing protocols based on a *delay and match* principle. The delay-and-match principle uses a protocol where they add samples into the pool during the testing process, i.e., say after, Δ cycles, and use this information while decoding. Similarly, the authors in [169] present algorithms for the so-called tropical group testing. The authors in [170] present random block test designs with a doubly-disjunct property and deterministic constructions for tropical group testing.

More recently, the Covid-19 detection problem has been addressed by various other formulations which incorporate the community structure [119] or side-information collected from contact tracing [118]. Further, techniques like learned factor graphs which combine factor graphs with deep learning methods [121], an expectation maximization-based adaptive protocol [171], a belief propagation approach to handle the scenario when two types of defectives, say, signifying omicron and delta variants of Covid-19 [172], need to be detected, have been proposed. Lastly, a high-performance Bayesian method that uses a distributed computing framework for streaming/online analysis (Apache Spark) has been proposed to scale the disease surveillance [173, 174]. These recent works on Covid-19 recovery algorithms are significantly different from the focus of this thesis in Chapter 4 and Chapter 5.

4.1.4 Contributions

In this chapter, we mainly focus on estimating the viral loads and determining healthy/sick status of individuals using the pooled RT-qPCR outcomes and the deterministic, single-stage pooling matrix when the PCR efficiency factor, q, is known. The rest of the chapter is organized as below:

1. We describe the system model for the pooled RT-qPCR in Section 4.2.

2. We then restate two binary (test outcome) group testing algorithms, namely, combinatorial orthogonal matching pursuit (COMP) and definite defectives (DD) as applied to the CT-space pooled RT-qPCR system model in Section 4.3. Recall that COMP makes only FP errors whereas DD makes only FN errors. We use COMP in

a pre-processing step to reduce the dimensions of the system model thereby ensuring faster convergence of the proposed recovery algorithms using quantitative test outcomes.

3. We develop two novel algorithms, namely, a gradient descent (GD)-CT method and an iterative mirrored hard thresholding (IMHT)-CT method, which can recover the individual CT vector given the pooled CT vector and the pooling matrix in Section 4.4 The challenge lies in addressing the non-linear nature of the mathematical model of the RT-qPCR process. Due to this, the recovery problem departs from the standard sparse signal recovery problem [175, 176].

4. We present a theorem to show that the GD-CT algorithm converges to a local optimum in Section 4.4.1 and also empirically evaluate the performance of our algorithms in terms of the normalized mean-squared error (NMSE), false positive rate (FPR) and false negative rate (FNR) in Section 4.5.

One of the main takeaways from this chapter is that the use of quantitative measurements in a non-adaptive pool test setting results in significant cumulative cost savings. Also, the optimal achievable testing rate vs. prevalence rate compiled using various deterministic pooling matrix designs and recovery algorithms in the non-linear RT-qPCR model is crucial for reducing pooled testing ideas to practice.

4.2 System Model and Problem Statement

A system model inspired by the RT-qPCR mechanism [166] is described in this section. Denote the maximum number of cycles in the PCR process by $c_{\max}$. The cycle number is indexed by t and therefore, $t \in \{1, 2, \ldots, c_{\max}\}$. Also, denote the efficiency of the PCR reaction by q, and let x_0 denote the initial viral load concentration (e.g., DNA molecules per μl). For the sample to be considered positive, the viral load concentration should cross a preset threshold τ in $c \leq c_{\max}$ PCR cycles. Using [166, Equation (7)], the relationship governing the growth of the viral load concentration up to the cycle threshold can be written as:

$$\prod_{t=1}^{\lfloor c \rfloor} (1 + q_t)(1 + q_{\lfloor c \rfloor + 1})^{c - \lfloor c \rfloor} x_0 = \tau.$$

Note that, $c \in \mathbb{R}$ although the cycles are indexed by $\{1, 2, \ldots, c_{\max}\}$. This is because the PCR machine's software performs the interpolation implied by the above equation to output a real-valued CT.

By calibrating the operating protocol, in practice, one can ensure that the variation in q_ts across tests is negligible. Then, the model can be simplified by letting $q_t = q$ for all $t = 1, 2, \ldots, c_{\max}$. Thus, (4.1) can be simplified as

$$(1 + q)^c x_0 = \tau. \tag{4.1}$$

Further, without loss of generality, we have taken the proportionality constant to be unity [166, see Equation (4)], since one can appropriately scale the threshold τ. The value of q depends on various factors like probe-primer combination [177], dilution of the test solution, whether annealing equilibrium [166] method is used or not, etc. In most lab experiments, the value of q is observed to lie in $\mathcal{X} \triangleq [0.5, 1.0)$ over various probe-primer combinations [166, 177]. As mentioned in Section 4.1.1, when the viral load is zero, the threshold τ is not reached even after $c_{\max}$ cycles. This is indicated by setting $c = \infty$. In practice, depending on the PCR kit and sample preparation protocols, manufacturer's instructions etc., $c_{\max} = 40$ to 50 PCR cycles are conducted.

We now extend the model to pool testing based on RT-qPCR and account for noise. Here, n individuals participate in m pool tests, with $m \ll n$. The CT values observed from the m pool tests are collected in a vector as $\mathbf{c} \triangleq [c_1, c_2, \ldots, c_m]^T$, where c_i is the CT of the ith pool test. Similarly, the initial viral loads contributed by the n individuals to the pool tests can be written as $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$.[4] In non-adaptive pool testing, the pool tests are defined by the binary pooling matrix $\mathbf{A} \in \{0,1\}^{m\times n}$, where the (i,j)th element of $\mathbf{A}$ equals 1 if the jth individual participates in the ith test, and equals 0 otherwise. We can now extend (4.1) to the pool testing framework and write the model for the ith pool's CT measurement, denoted by c_i, as

$$\tau(1+q)^{-c_i} = (1+q)^{\epsilon_i}\mathbf{A}_i^T\mathbf{x}\ , \quad i = 1, 2, \ldots, m, \tag{4.2}$$

where $\mathbf{A}_i^T$ is the ith row of $\mathbf{A}$ and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is the CT measurement noise with unknown variance σ_ϵ^2.

In (4.2), the term $\mathbf{A}_i^T\mathbf{x}$ represents the effect of pooling. That is, the total initial viral load in the ith pool test is the sum of all the initial viral loads of the individual samples participating in the ith test, determined by the locations of 1s in the row $\mathbf{A}_i^T$. Also, note that the noise contribution appears as an exponent to the overall process efficiency factor, $(1+q)$. This is due to (4.1), where the CT depends on the initial viral load via an exponential term. Therefore, any additive noise observed during the measurement of the

[4] Note that, if the jth individual does not participate in the ith test, we may set $x_j = 0$ for that test. Also note that, in practice, the jth individual may contribute different initial viral loads to the different tests it participates in. However, in the detection regimes of interest, where reliable detection is possible (e.g., where the positive individuals contribute about 100 or more viral particles per μl to the test), these variations do not significantly affect the resulting CT values.

CT values shows a similar exponential dependency with the initial viral load. A similar model was used in [88, Equation (9)], where the authors expressed system model in terms of the ratio of the initial viral loads in the pooled samples and the viral load corresponding to the minimum observed CT among all pools. In contrast, the (intermediate) model in (4.2) relates the actual un-normalized initial viral loads of the individual samples with the observed pooled CT values. Further, as we shall see, our final model relates the observed pooled CT values to the individual sample CT values, which we want to estimate.

4.2.1 Problem Statement

The goal is to solve the inverse problem of inferring the vector of individual viral loads, $\mathbf{x}$, from (4.2) given the pooling matrix $\mathbf{A}$ and m pooled CTs, $c_1, c_2, \ldots, c_m$. We note that, the system model as shown in (4.2) is under-determined since $m \ll n$. When the prevalence rate is low, $\|\mathbf{x}\|_0 \ll n$, sparse signal recovery methods can potentially be used for solving the inverse problem at hand.[5] However, our measurement model in (4.2) is nonlinear and the noise is multiplicative, unlike the standard sparse signal recovery problem [175, 176]. Further, the values of τ and q are unknown. The range of values over which the viral loads typically vary is large, making the inverse problem numerically hard to solve. Suppose $x_0 = 1$ in (4.1) (i.e., 1 viral particle per μl) results in $c = 35$ in the noiseless case. Assuming an ideal PCR efficiency of $q = 1$, we obtain $\log_2 \tau = 35$. Thus, an observed $c = 15$ corresponds to a viral load $x_0 = 2^{20} \approx 10^6$. As a result, the range over which the viral load can potentially vary is $[1, 10^6]$. In contrast, the range over which the CT values vary is much smaller, i.e., $[15, 35]$.

[5] The expression $\|\mathbf{x}\|_0$ denotes the l_0 norm of a vector $\mathbf{x}$, i.e., the number of non-zero elements in $\mathbf{x}$ [178, Definition 2.1].

In order to solve the problem in a numerically stable manner, we transform the problem from *viral-load*-space into *CT*-space. To keep the notations distinct between the *pooled-sample CTs* and the *individual-sample CTs*, the individual-sample CTs are henceforth denoted by $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T \in \mathbb{R}^{n\times 1}$. From (4.1), the individual-sample CTs are related to the viral loads as

$$\mathbf{x} = \tau(1+q)^{-\mathbf{u}}, \tag{4.3}$$

where $(1+q)^{-\mathbf{u}} \triangleq [(1+q)^{-u_1}, (1+q)^{-u_2}, \ldots, (1+q)^{-u_n}]^T \in \mathbb{R}^{n\times 1}$. Hence, we get

$$c_i = -\frac{1}{\log(1+q)} \log\left(\mathbf{A}_i^T (1+q)^{-\mathbf{u}}\right) + \epsilon_i \ , \quad i = 1, 2, \ldots, m. \tag{4.4}$$

In matrix-vector form,

$$\begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = -\frac{1}{\log(1+q)} \begin{bmatrix} \log\left(\mathbf{A}_1^T (1+q)^{-\mathbf{u}}\right) \\ \vdots \\ \log\left(\mathbf{A}_m^T (1+q)^{-\mathbf{u}}\right) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix},$$

$$\text{or } \mathbf{c} = -\frac{1}{\log(1+q)} \log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right) + \boldsymbol{\epsilon}. \tag{4.5}$$

The goal now is to solve the inverse problem of inferring **u**, the vector of individual CTs, from (4.5), given the pooling matrix, **A**, and the vector of pooled CTs, **c**. We note that the parameter τ does not appear in the CT-space formulation of the system model.

In summary, the following points:

- Usage of binary pooling matrix and the fact that there are two kinds of pooled test outcomes: negative test outcomes have CT = ∞ while the positive test outcomes are non-negative finite real values, *and*

- Multiplicative and non-Gaussian nature of the noise term, $(1+q)^{\epsilon}$ as seen in (4.2) or additive nature of the noise but with a non-linear model in the log space as seen in (4.5),

make the problem challenging. To this end, we develop a set of novel, robust recovery algorithms to estimate the vector of individual CTs.

4.3 Recovery Algorithms: Binary-Valued Measurements

In this section, we recap two well-known binary pool testing algorithms: COMP and DD [5,7] applied to the CT system model after transforming the model to a binary system. Recall that there are a variety of binary pool testing algorithms like SSS, SCOMP, etc. in the literature [5, 7]. We restrict our comparisons to COMP and DD, because, in a noiseless setting, (1) COMP algorithm does not make a type-2 (or FN) error; (2) DD algorithm does not make a type-1 (or FP) error as mentioned in Section 3.1. As mentioned earlier in Section 4.1.3, COMP is used in the pre-processing stage of the CT algorithms (see Section 4.4 and Section 5.3).

4.3.1 Combinatorial Orthogonal Matching Pursuit (COMP)

The measured pool CT's are converted to binary values using

$$b_i = \begin{cases} 1, & c_i \neq \infty \\ 0, & c_i = \infty, \end{cases} \tag{4.6}$$

for $i = 1, 2, \ldots, m$. Let $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ denote the index set of samples. The samples appearing in only negative pool tests (i.e., $b_i = 0$) are declared as definite negatives, denoted by the set $\mathcal{DN}$. Then, the samples remaining in the set $\mathcal{N} \setminus \mathcal{DN}$ are declared as positive samples.

In the sequel, we first pre-process the CT values using the COMP algorithm to filter out the definite negatives. Then, in the system model, we remove the rows of $\mathbf{A}$ corresponding to the negative test outcomes, and the columns of $\mathbf{A}$ and the entries of $\mathbf{u}$ corresponding to the definite negatives. This, in turn, reduces the overall system dimension for further processing. In fact, at low prevalence rates, the reduced system can even be over-determined. The application of COMP also removes the negative tests with CTs equal to ∞ and is thus numerically better suited for optimization-based recovery algorithms. Now, in order to avoid additional notation, we reuse the notations defined earlier to define the model after the COMP stage. For example, n and m will respectively denote the number of samples and number of tests remaining, after the COMP stage.

4.3.2 Definite Defectives (DD)

The DD algorithm comprises two stages. In stage-1, similar to the COMP algorithm, the definite negatives and hence the set of *possible positives*, $\mathcal{PP} \triangleq \mathcal{N} \setminus \mathcal{DN}$ is determined. In stage-2, the items from $\mathcal{PP}$ which are sole participants in positive pool tests (i.e., $b_i = 1$) are declared as positive samples and the rest are declared as negative samples.

4.4 Recovery Algorithms: Known PCR Efficiency Factor

In this section, we describe two algorithms which use the pooled CTs, the efficiency factor, q and the pooling matrix, $\mathbf{A}$ as inputs and recover the individual CTs.

4.4.1 Gradient Descent (GD) - CT Algorithm

Recall that the additive measurement noise is modeled as a Gaussian, i.e., $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$, $i \in \{1, 2, \ldots, m\}$. Further, the vector of viral loads, $\mathbf{x}$, is sparse. An optimum solution to (4.5) can be obtained by a maximum aposteriori probability (MAP)-type formulation with a spike-and-slab prior [179]. This leads to a l_0-norm regularized optimization objective comprised of the sum of two terms. The first term arises from the least-squares or l_2-norm minimization of the error vector, and the second (regularization) term imposes sparsity. In particular, with (4.5) as the system model, the optimization problem we wish to solve can be written as follows:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \frac{1}{2} \|\boldsymbol{\epsilon}\|_2^2 + \bar{\kappa} \left\|\tau(1+q)^{-\mathbf{u}}\right\|_0 , \tag{4.7}$$

where $\bar{\kappa}$ is a regularization parameter and the expression $\|\boldsymbol{\epsilon}\|_2$ denotes the l_2 norm of a vector $\boldsymbol{\epsilon}$ [180].

We note that the ℓ_0 optimization problem in (4.7) is NP-hard (see Appendix C.3). A commonly adopted approach is to relax the above NP-hard problem into an ℓ_1 optimization problem [175, 181, 182]. Towards this end, we define $\tilde{\mathbf{x}} = (1+q)^{-\mathbf{u}}$ such that $\|\tilde{\mathbf{x}}\|_\infty \leq \tilde{\kappa}$, followed by computing the convex biconjugate of $f(\tilde{\mathbf{x}}) = \|\tilde{\mathbf{x}}\|_0$ to get $f^{**}(\tilde{\mathbf{x}}) = \frac{1}{\tilde{\kappa}}\|\tilde{\mathbf{x}}\|_1$ [183]. We then replace the ℓ_0 norm term in (4.7) with its ℓ_1 relaxation to get

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \frac{1}{2} \|\boldsymbol{\epsilon}\|_2^2 + \kappa \left\|(1+q)^{-\mathbf{u}}\right\|_1 , \tag{4.8}$$

where $\kappa \triangleq \bar{\kappa}|\tau|/\tilde{\kappa} \overset{\tau\geq 0}{=} \bar{\kappa}\tau/\tilde{\kappa}$ is the effective regularization parameter for the optimization problem and the expression $\|\mathbf{x}\|_1$ denotes the l_1 norm of a vector $\mathbf{x}$ [180]. We note that (4.8) is still a hard problem to solve due to its non-linear and non-convex nature. In the literature, the gradient descent (GD) is one of the attractive and practical algorithms available to solve a wide range of optimization problems [184]. It is especially useful when the gradient of the objective function exists and can be efficiently computed. The

Algorithm 1 Gradient Descent (GD) - CT

```
Input: $\mathbf{c}$, $\mathbf{A}$, $K$, $\kappa$, $\eta$ and $q$
Output: $\hat{\mathbf{u}}$
Initialize $\mathbf{u}_1$.
for $k \leftarrow 1$ to $K$ do
    Compute $\mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q)$ using (4.9) with $\mathbf{u} = \mathbf{u}_k$.
    Update
        $\mathbf{u}_{k+1} = \mathbf{u}_k - \eta \mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q).$    (4.10)
end for
Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{K+1}$.
```

optimization problem in (4.8) can be solved using the GD algorithm and is termed as the *GD-CT* algorithm due to the usage of CT values.

In order to derive the GD-CT algorithm, we start by computing the gradient of the objective function in (4.8) with respect to $\mathbf{u}$, denoted by $\mathbf{g}(\mathbf{u}; \mathbf{A}, \mathbf{c}, q)$. Denoting the (i, j)th element of $\mathbf{A}$ by A_{ij}, the gradient is given by

$$\mathbf{g}(\mathbf{u}; \mathbf{A}, \mathbf{c}, q) = \sum_{i=1}^{m} \left(\frac{\log\left(\mathbf{A}_i^T (1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i \right) \times \frac{-1}{\mathbf{A}_i^T (1+q)^{-\mathbf{u}}} \begin{bmatrix} \frac{A_{i1}}{(1+q)^{u_1}} \\ \vdots \\ \frac{A_{in}}{(1+q)^{u_n}} \end{bmatrix} - \kappa \left(\ln(1+q)\right)(1+q)^{-\mathbf{u}}. \tag{4.9}$$

The GD-CT procedure is presented in Algorithm 1, where η is the step size and K is the maximum number of iterations.

In Algorithm 1 and henceforth, we denote the vector $\mathbf{u}$ at kth iteration of the algorithm and the jth entry of $\mathbf{u}$ by $\mathbf{u}_k$ and u_j, respectively. Then, the (j, j)th diagonal element of the Hessian $\mathbf{H}(\mathbf{u})$ of the objective function is given by

$$H_{jj}(\mathbf{u}) = \sum_{i=1}^{m} \left[\frac{c_i A_{ij} \ln(1+q)(1+q)^{-u_j}}{\mathbf{A}_i^T (1+q)^{-\mathbf{u}}} - \frac{c_i A_{ij}^2 \ln(1+q)(1+q)^{-2u_j}}{(\mathbf{A}_i^T (1+q)^{-\mathbf{u}})^2} \right.$$

$$
\begin{aligned}
&-\frac{A_{ij}\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{(1+q)^{u_j}\;\mathbf{A}_i^T(1+q)^{-\mathbf{u}}}+\frac{A_{ij}^2(1+q)^{-2u_j}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2}\\
&\left.-\frac{A_{ij}^2\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})(1+q)^{-2u_j}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2}\right]-\kappa\ln^2(1+q)(1+q)^{-u_j}, \qquad (4.11)
\end{aligned}
$$

and the (j,j')th element for $j \neq j'$ is given by

$$
H_{jj'}(\mathbf{u})=\sum_{i=1}^{m}\frac{A_{ij}A_{ij'}(1+q)^{-u_j-u_{j'}}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2}\times\left[c_i+\frac{1}{\log(1+q)}-\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})\right]. \qquad (4.12)
$$

Theorem 4.1. *[185, 186] Given an open convex set, $\mathcal{S}$, let the objective function, $f : \mathcal{S} \mapsto \mathbb{R}$ be twice differentiable. Let η be the step size of the GD-CT algorithm such that $0 < \eta < 1/L$, where $\nabla_{\mathbf{u}} f$ is L-Lipschitz continuous, for $L < \infty$. The GD-CT update as given by (4.10), generates a sequence of iterates $\{\mathbf{u}_k\}$. If the $\lim_{k\to\infty}$ $\mathbf{u}_k$ exists, then $\mathbb{P}\left(\lim_{k\to\infty}\ \mathbf{u}_k = \mathbf{u}^*\right) = 1$, where $\mathbf{u}^*$ is a local minimizer.*

Theorem 4.1 says that the gradient descent algorithm never converges to a saddle point.[6] We now show that the objective function in (4.8) satisfies the conditions in Theorem 4.1, and, therefore, the GD-CT iterates converge to a local minimum.

The objective function in (4.8) is twice differentiable (see (4.11) and (4.12)) and is comprised of elementary operations. Hence, it is analytic and satisfies the *Łojasiewicz gradient inequality.*[7] Therefore, the limit of the iterates exists [185]. Further, showing that the gradient is L-Lipschitz is equivalent to bounding the spectral norm of the Hessian, $\|\|\mathbf{H}\|\|_2 = \sqrt{\lambda_{\max}(\mathbf{H}^H\mathbf{H})} \leq L < \infty$, where $\lambda_{\max}(\mathbf{H}^H\mathbf{H})$ is the largest eigenvalue of $\mathbf{H}^H\mathbf{H}$ [186]. Since $\mathbf{H}^H\mathbf{H} \succcurlyeq 0$, we have $\lambda_{\max}(\mathbf{H}^H\mathbf{H}) \leq \text{trace}(\mathbf{H}^H\mathbf{H}) = \|\mathbf{H}\|_F^2$.[8] Therefore, it suffices to show that the entries of the Hessian are upper bounded by finite constants.

[6] See Appendix C.4 for an intuition, reproduced from [185].

[7] See Appendix C.5 for the formal definition and a note.

[8] The expressions $\|\mathbf{A}\|_F^2$ and $\|\|\mathbf{A}\|\|_2$ denote the Frobenius norm and spectral norm of the matrix $\mathbf{A}$ and $\mathbf{A} \succcurlyeq 0$ means that the matrix $\mathbf{A}$ is positive semi-definite [180].

Recall that the COMP pre-processing removes the tests with pooled CT values $c_i = \infty$. Thus, at least one positive sample with $u_j < \infty$ appears in each test, and the pooled CT is bounded as $c_i \leq c_{\max}$. This also ensures that $\mathbf{A}_i^T(1+q)^{-\mathbf{u}} \neq 0$ in the denominator of (4.11) or (4.12). Let $\mathcal{S} = (m_u, M_u)$ such that, $0 < m_u < u_j < M_u$. The positive lower bound assumption is valid since at least one PCR cycle is conducted. Let $\tilde{\mathbf{x}} = (1+q)^{-\mathbf{u}}$, such that the entries, $\tilde{x}_j$, of $\tilde{\mathbf{x}}$ are bounded between $m_x < \tilde{x}_j < M_x < \infty$ for the positive individuals and $\tilde{x}_j = 0$ for the negative individuals, where m_x and M_x are constants depending only on M_u and m_u, respectively. Lastly, note that $\log(m_x \left\|\mathbf{A}_i^T\right\|_1) < \log(\mathbf{A}_i^T\tilde{\mathbf{x}}) < \log(M_x \left\|\mathbf{A}_i^T\right\|_1)$. Since $1 \leq \left\|\mathbf{A}_i^T\right\|_1 \leq n$, we can relax the bound further as $\log(m_x) < \log(\mathbf{A}_i^T\tilde{\mathbf{x}}) < \log(nM_x)$. Define $\gamma \triangleq \max(|\log(m_x)|, |\log(nM_x)|)$ and note that $|\log(\mathbf{A}_i^T\tilde{\mathbf{x}})| < \gamma$. Then, the entries of the Hessian can be bounded as

$$H_{jj'} \leq \begin{cases} C_H\left[1 + \gamma(n+1) + nc_{\max}\ln(1+q)\right], & j = j' \\ C_H\left[\gamma + c_{\max} + \frac{1}{\ln(1+q)}\right], & j \neq j', \end{cases}$$

with $C_H \triangleq m(M_x/m_x)^2$. Since $\|\mathbf{H}\|_F^2$ contains n diagonal and $n(n-1)$ off-diagonal entries of $\mathbf{H}$, the trace$(\mathbf{H}^H\mathbf{H})$, and hence the $|||\mathbf{H}(\mathbf{u})|||_2$ is finitely upper bounded, satisfying the last of the sufficient conditions for almost sure convergence of Algorithm 1 to a local minimizer using Theorem 4.1.

Remark: Instead of using a constant step-size η, we can use adaptive step-sizes. Algorithm 1 can be modified to allow step-size adaptation as follows: Replace *step 4* by

$$\eta_k = \text{AdaptStep}(\eta_{k-1})$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \eta_k \mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q),$$

where AdaptStep$(\cdot)$ implements the recipe to adapt the step size [187, 188]. In this case, the L-Lipschitz requirement on the gradient in Theorem 4.1 can be relaxed.

4.4.2 Iterative Mirrored Hard Thresholding (IMHT) - CT Algorithm

Motivated by the literature in non-linear compressed sensing [189], the system model in (4.5) can be seen as a non-linear transformation relating the pooled CTs vector, $\mathbf{c}$ with the individual CTs vector, $\mathbf{u}$, given by $\mathbf{c} = \Phi(\mathbf{u}) + \boldsymbol{\epsilon}$, where $\Phi(\mathbf{u}) \triangleq -\frac{\log\left(\mathbf{A}\left(\frac{1}{(1+q)^{\mathbf{u}}}\right)\right)}{\log(1+q)}$ is a multi-variable vector-valued non-linear function. A first-order Taylor series approximation of $\Phi(\mathbf{u})$ about a fixed point $\mathbf{u}^*$ can be written as $\Phi(\mathbf{u}) \approxeq \Phi(\mathbf{u}^*) + \mathbf{J}(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*)$ and is valid when $\|\mathbf{u} - \mathbf{u}^*\| \leq \delta'$ for a sufficiently small $\delta' > 0$. In the Taylor series expansion, $\mathbf{J}(\mathbf{u})$ is the $m \times n$ Jacobian matrix of $\Phi(\mathbf{u})$ computed at $\mathbf{u} = \mathbf{u}^*$. The (i, j)th element of $\mathbf{J}(\mathbf{u})$ is

$$\mathbf{J}_{ij}(\mathbf{u}) = \frac{\partial(\Phi_i(\mathbf{u}))}{\partial u_j} = \frac{A_{ij}}{(1+q)^{u_j}}\left(\frac{1}{\mathbf{A}_i^T(1+q)^{-\mathbf{u}}}\right), \tag{4.13}$$

where $\Phi_i(\mathbf{u})$ is the ith element of $\Phi(\mathbf{u})$.

We introduce the following additional notations before describing the algorithm. Let s denote the number of finite entries in the vector, $\mathbf{u}$ or equivalently, the sparsity of the viral-load vector, $\mathbf{x}$. Let $\boldsymbol{\Pi}_{u_{th},s}(\mathbf{u})$ denote the *mirrored* hard-thresholding operation, i.e., the operation that sets the $n - s$ largest values of $\mathbf{u}$ to u_{th}. With η denoting the step size and K denoting the maximum number of iterations, the *IMHT-CT* algorithm is described in Algorithm 2. Although Algorithm 2 takes the value of s as an input, we have observed that an overestimated value of s does not affect the simulation performance.

Algorithm 2 Iterative Mirrored Hard Thresholding (IMHT) - CT

```
Input: $\mathbf{c}$, $\mathbf{A}$, $s$, $K$, $\eta$ and $q$
Output: $\hat{\mathbf{u}}$
Initialize $\mathbf{u}_1$.
for $k \leftarrow 1$ to $K$ do
    Compute the entries of $\mathbf{J}(\mathbf{u}_k)$ using (4.13) for each $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$ with $\mathbf{u} = \mathbf{u}_k$.
    Update
        $\mathbf{u}_{k+1} = \mathbf{\Pi}_{u_{th},s}(\mathbf{u}_k - \eta \mathbf{J}(\mathbf{u}_k)\,(\mathbf{c} - \Phi(\mathbf{u}_k))).$   (4.14)
end for
Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{K+1}$.
```

4.5 Numerical Simulations

In this section, we present the simulation results showing the effectiveness of our algorithms for recovering the infected samples given the outcomes of group testing and the pooling matrix. We start by describing the experimental setup in the next subsection.

4.5.1 Experimental Setup

For generating the simulation data, we use $q = 0.95$ and the Kirkman matrix[9] with $m = 45$ rows and $n = 105$ columns as the pooling matrix, unless stated otherwise.[10] We choose $m = 45$ because a standard PCR plate can accommodate a maximum of 93 test samples at a time (See Section 4.1.2). Therefore, using $m = 45$, one can perform two pooled tests in parallel with a 93-capacity plate, making it practically useful in terms of reducing wastage due to empty wells. The indices of nonzero viral loads in $\mathbf{x} \in \mathbb{N}^{105\times 1}$ are picked uniformly at random and stored in an index set $\mathcal{NZI}$ with cardinality $|\mathcal{NZI}| = \lceil ns_x/100 \rceil$, where

[9] See Appendix C.1 for a note on Kirkman designs.

[10] We obtain Kirkman matrix of size 45×105 by selecting first 105 columns from the full sized 45×285 Kirkman matrix. Other options which can be used are to select columns uniformly randomly without replacement; or select/drop a column if it does not increase/decrease the mutual coherence, etc.

s_x (%) is the prevalence rate. The viral load in a negative sample is zero. The positive entries of the initial viral-load vector, $\mathbf{x}_j$, are drawn from the Poisson(λ_j) distribution where $\lambda_j \sim \text{Unif}(100, 10^6)$, for $j \in \mathcal{NZI}$. Finally, using the relationship in (4.2), the pooled CT vector, $\mathbf{c} \in \mathbb{R}_+^{45\times 1}$ is generated. In the distribution of the hyper-parameter λ_j, the lower and upper limits are the average least and highest viral load (or equivalently, cDNA/ RNA particles) in a positive sample.

We use CT $= 45$ as the decision threshold, i.e., samples with estimated CT > 45 are declared as negative. This parameter can be easily varied based on the actual number of PCR cycles conducted in the RT-qPCR machine. The maximum number of iterations is set as $K = 500$. The CT threshold value is set as $u_{th} = 100$ in Algorithm 2. Also, in Algorithm 2, the true value of s is assumed to be known in our simulations. In practice, s can be estimated using the prevalence rate curve. Finally, the regularization parameter, $\kappa = 10^{-3}$, and the step size $\eta = 0.01$ are chosen via cross-validation. We have noticed in our simulations that the value of κ can be set very close to 0 without significantly changing the performance. This is because, after the COMP stage, the problem is often over-determined. Therefore, the relative importance of the sparsity promoting term is low. Also, since the viral loads are non-negative, the l_1 regularization is not necessary to ensure a unique solution to (4.8) [190, 191].

All the results are averaged over 10,000 Monte-Carlo runs. We use the following metrics: average NMSE in the CT values recovered, the average FNR, and the average FPR, for the prevalence rate (sparsity) up to 10%. The NMSE is the mean squared error (MSE) between the estimate $\hat{\mathbf{u}}$ and the true vector $\mathbf{u}$ normalized by the mean ℓ_2 norm-squared of the true vector, i.e., NMSE $\triangleq \mathbb{E}\left[\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2\right] / \mathbb{E}\left[\|\mathbf{u}\|_2^2\right]$. Ignoring the role of the infinities

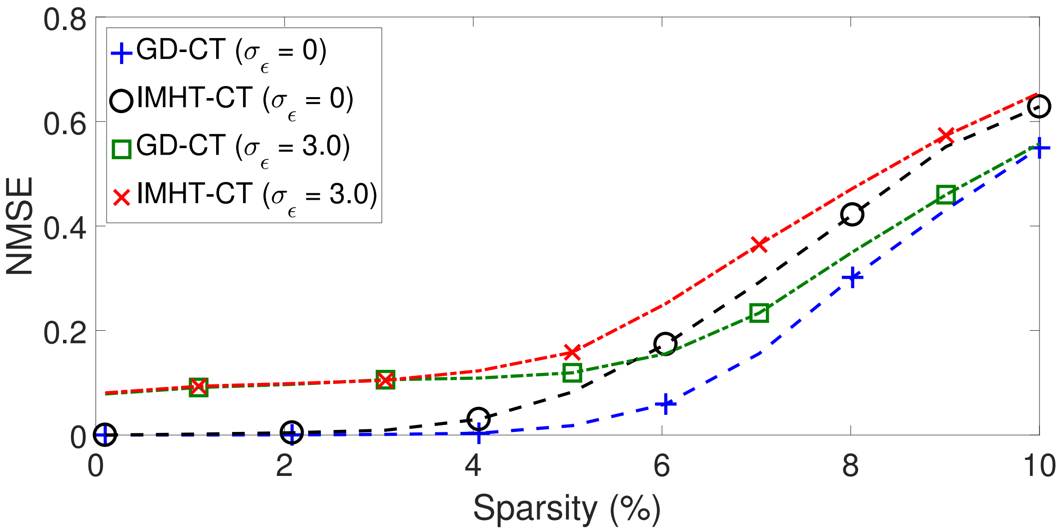


Figure 4.1: Comparison of the NMSE performance of GD-CT and IMHT-CT algorithms over different sparsity (%) levels at different noise standard deviations, σ_ϵ.

in the computation of ℓ_2 norm of the CT vector, we note that the average ℓ_2 norm of the CT vector increases as the number of defective items in the population increases, thereby proportionally scaling un-normalized error metrics like MSE. In order to remove this bias in the error performance when comparing across the sparsity levels, the NMSE is considered instead of the MSE.

The comparison is performed across two scenarios: firstly, under the various noise standard deviations, $\sigma_\epsilon \in \{0, 3.0\}$ [177] at $q = 0.95$ and secondly, under different efficiency factors, $q \in \{0.5, 0.95\}$ at $\sigma_\epsilon = 0$. The noise standard deviation $\sigma_\epsilon = 0$ indicates the noiseless model.

4.5.2 Algorithm Performance at Different Noise Levels

We illustrate the NMSE performance of the GD-CT and IMHT-CT algorithms in Figure 4.1 as the sparsity (%) is varied from 0.1% to 10% over two different values of the noise standard deviations, σ_ϵ. From Figure 4.1, we observe that the NMSE curves shift

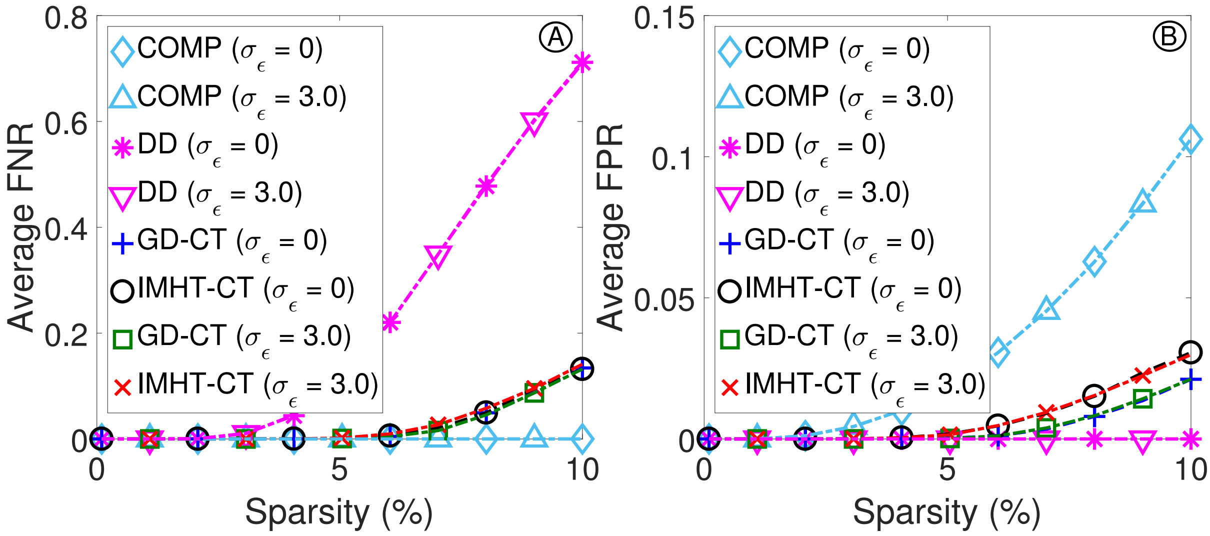


Figure 4.2: Comparison of the FNR and FPR performances of COMP, DD, GD-CT and IMHT-CT algorithms over different sparsity (%) levels at different noise standard deviations, σ_ϵ.

up when the noise $\sigma_\epsilon > 0$. The NMSE of GD-CT algorithm is lower than that of the IMHT-CT algorithm at higher sparsity levels. As the number of infected sample in the population increases, we observe that the NMSE increases at faster rate and the NMSE curves corresponding to the noiseless (i.e., $\sigma_\epsilon = 0$) and noisy (i.e., $\sigma_\epsilon = 3.0$) case come closer to each other. The decoding capability of the recovery algorithm depends on the test matrix design. Therefore, the decoding process fails irrespective of noise level and the recovery algorithm estimates get worse. Lastly, in addition to offering a better NMSE error, the NMSE curve of the GD-CT algorithm stays at its minimum value up to a sparsity of 4% as compared to IMHT-CT where the curve raising above the minimum value from 2% at $\sigma_\epsilon = 0$. Similar behavior is seen in the noisy case too making the GD-CT, a better algorithm, of the two.

The FNR and FPR performances of GD-CT and IMHT-CT algorithms are shown in Figure 4.2 at two noise standard deviations as the sparsity (%) is varied from 0.1% to 10%.

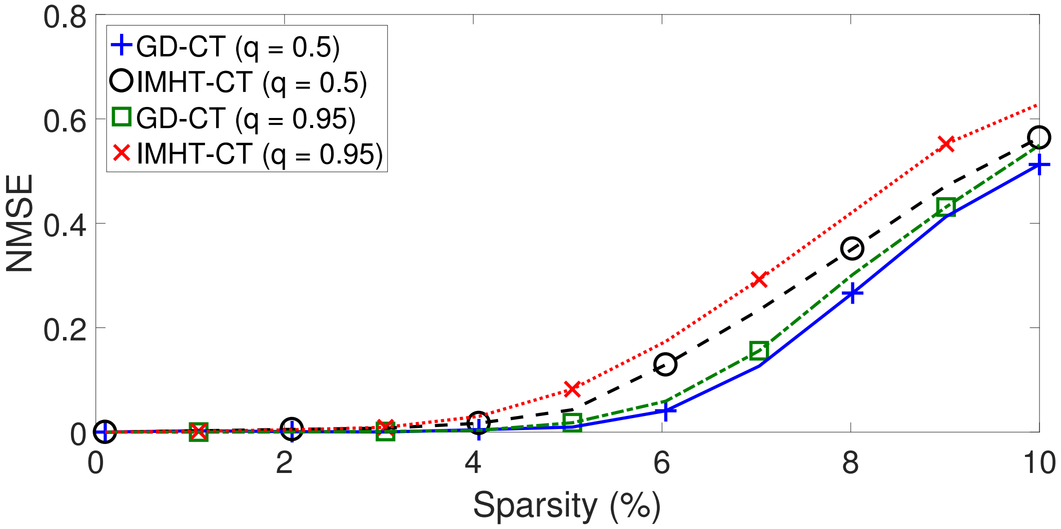


Figure 4.3: Comparison of the NMSE performance of GD-CT and IMHT-CT algorithms over different sparsity (%) levels at two PCR efficiency factors, q, used to generate the data.

From the subplot Ⓐ in Figure 4.2, we see that the FNR of both GD-CT and IMHT-CT algorithms are similar across σ_ϵ. In addition, COMP makes no FN errors irrespective of the noise level whereas DD makes FN errors higher than either GD-CT or IMHT-CT algorithms as sparsity levels increase. From subplot Ⓑ in Figure 4.2, we observe that the performance of the GD-CT algorithm is better than that of the IMHT-CT algorithm. Here, we observe further that DD makes no FP errors whereas COMP makes much higher FP error as compared to GD-CT and IMHT-CT algorithms. Further, we see that the FPR and FNR performances of both GD-CT and IMHT-CT algorithms are not affected by the noise levels. Lastly, even though the NMSE performance of the algorithms degrade as the infected sample percentage in the population increases, the FNR and FPR remain stable and agnostic of the noise level in the system. In order to degrade the FNR and FPR performance, the noise levels have to be set to much higher values.

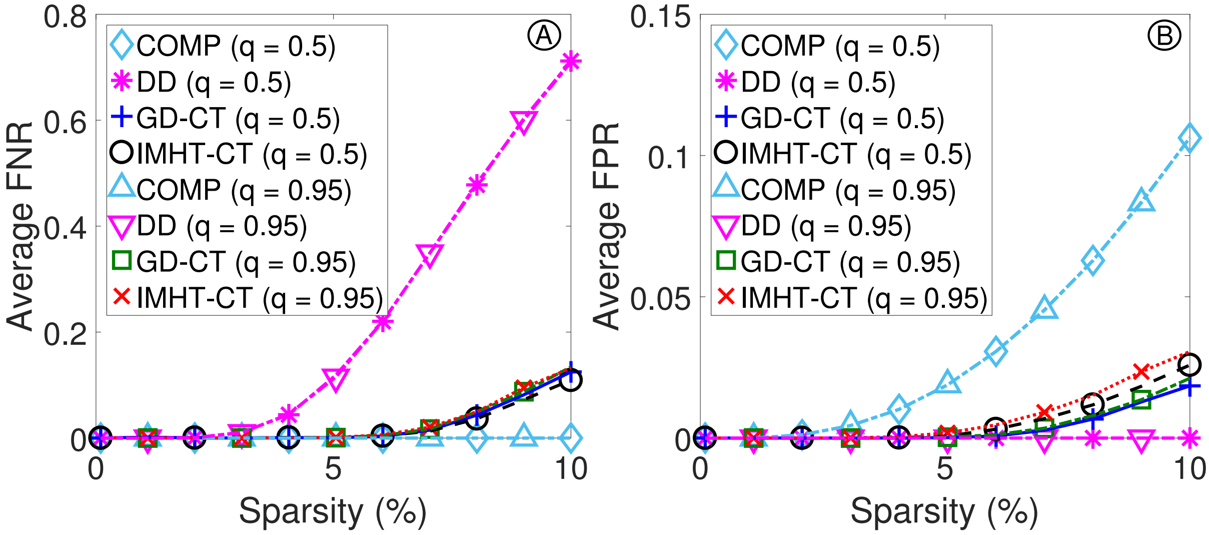


Figure 4.4: Comparison of the FNR and FPR performances of COMP, DD, GD-CT and IMHT-CT algorithms over different sparsity (%) levels at different PCR efficiency factors, q, used to generate the data.

4.5.3 Algorithm Performance at Different PCR Efficiency Factors

In this subsection, we discuss the performance of the GD-CT and IMHT-CT algorithms across the PCR efficiency factors. The NMSE performance of the GD-CT and IMHT-CT algorithms are shown in Figure 4.3 as the sparsity (%) is varied from 0.1% to 10% over two different values of the PCR efficiency factor, q. It can be observed that the NMSE performance of GD-CT across efficiency factors is better than that of the IMHT-CT algorithm. The relative variation in the performance of algorithms across the two efficiency factors (q) is $\approx$ 20% for the IMHT-CT algorithm and $\approx$ 9% for the GD-CT algorithm at sparsity level of 10%. Thus, the performance of the GD-CT algorithm is more stable with respect to variations in the efficiency factor.

The FNR and FPR performances of GD-CT and IMHT-CT algorithms are shown in Figure 4.4 at two PCR efficiency factors as the sparsity (%) is varied from 0.1% to 10%. From the subplot Ⓐ in Figure 4.4, we see that the FNR of both GD-CT and IMHT-CT algorithms are similar across the two values of q. In addition, as seen before in Section 4.5.2, COMP makes no FN errors whereas DD makes FN errors higher than either GD-CT or IMHT-CT algorithms as sparsity levels increase irrespective of the value of q used to generate the data. From subplot Ⓑ in Figure 4.4, we observe that the performance of the GD-CT algorithm is slightly better than that of the IMHT-CT algorithm. Here, we observe further that DD makes no FP errors whereas COMP makes much higher FP error as compared to GD-CT and IMHT-CT algorithms. Further, we see that the FNR and FPR performance of GD-CT is not affected by the value of q used to generate the data. However, the performance of IMHT-CT algorithm shows a slight variation at higher sparsity (%) across the two values of q used in the experiment.

4.6 Chapter Summary

In this chapter, we developed novel recovery algorithms for detecting Covid-19-infected samples using pooled RT-qPCR. The binary group testing algorithms like COMP and DD were restated with the Covid-19 system model. COMP makes only FP errors, whereas DD makes only FN errors. Therefore, COMP can be used in the pre-processing stage to reduce the system dimension. In addition, two novel, iterative quantitative group testing algorithms, namely, GD-CT and IMHT-CT, were presented. The GD-CT iterates converge to a local minimum of the cost function in (4.8).

The update step in (4.14) of the IMHT-CT algorithm executes a projection step, namely, the mirrored-hard thresholding operation, $\mathbf{\Pi}_{u_{th},s}(\mathbf{u})$. Compared to the GD-CT algorithm, where iterates are computed optimally using the direction of the negative gradient, the projection step in the IMHT-CT algorithm does not guarantee that the projected iterate exactly satisfies the optimality condition. Therefore, there is a trade-off between ensuring that the iterates are feasible (as in IMHT-CT) vs. the iterates satisfying the optimality condition (as in GD-CT). Our numerical simulations revealed that the FPR and NMSE performance of the GD-CT algorithm is better than that of the IMHT-CT algorithm, whereas the FNR performances are similar. Further, the GD-CT algorithm performs better in terms of NMSE as compared to the IMHT-CT algorithm. Overall, in balance, the GD-CT algorithm outperforms the IMHT-CT algorithm.

The GD-CT and IMHT-CT algorithms take the PCR efficiency factor, q, as an input. The value of q depends on various factors like the probe-primers used and ambient temperature, to name a few. Therefore, it is essential to jointly estimate both the individual sample CTs and the PCR efficiency factor. We address this issue in the next chapter.

5 Robust Recovery Algorithms for Covid-19 Under Uncertainty

Chapter Highlights

This chapter overcomes a shortcoming of the algorithms presented in the previous chapters. That is, we focus on recovering the CT values of the individuals from the pooled-CT values under uncertainty. The value of the PCR efficiency factor, q, dictates the pace of the reaction, i.e., the rate at which the viral load in the sample replicates. In practice, q is unknown and, therefore, is set to a nominal value, e.g., 0.95. In theory, an ideal value for the PCR efficiency factor is unity. However, the value of the PCR efficiency factor varies in the range $[0.5, 1.0)$ and depends on factors like the probe-primer combination used and the ambient temperature. Due to the highly non-linear nature of the RT-qPCR system model where q appears as the base of an exponential term, a small change in q results in a wide variation in the CT value. Therefore, algorithms that can recover the infected sample CT values when q is unknown are essential. In this chapter, we develop two novel, iterative algorithms: 1) Block Coordinate Descent - CT (BCD-CT) and 2) Alternating Direction Method of Multipliers - CT (ADMM-CT), which can jointly estimate both the individual sample CTs and the PCR efficiency factor. The individual sample CTs can then be used to determine the infection status of the samples. Our algorithm performances are robust, i.e., a similar performance is observed across the range of values q can take.

We start this chapter with a recap of the related work and the system model presented in the previous chapter. Next, we describe the BCD-CT and ADMM-CT algorithms, followed by a discussion on the convergence to a local optimum. Numerical simulations are presented comparing the NMSE, FPR and FNR performances of BCD-CT and ADMM-CT. In terms of NMSE and FPR, the ADMM-CT outperforms BCD-CT. The performance is similar in terms of the FNR. Further, the performance is compared with that of binary group testing algorithms like COMP and DD, as well as CS-based algorithms. Lastly, we present empirical results related to the Covid-19 pandemic testing numbers to show when group testing is useful, followed by a discussion on the impact of test designs on the testing rates obtainable for different prevalence rates.

5.1 Introduction

The recent Covid-19 pandemic has increased the interest in novel group testing algorithms [88, 165]. In particular, recovery algorithms that can estimate the infection status of an individual sample from the quantitative pooled test outcome and pooling matrix have gained much attention (see Chapter 4). As discussed in Section 4.1.2, systematic, cost-effective, reliable, and repeatable testing protocols that form the backbone of a standard contact tracing procedure are essential. Towards this end, binary- and quantitative-outcome group testing algorithms were discussed in Section 4.3 and Section 4.4, respectively. The GD-CT and IMHT-CT algorithms presented in Section 4.4 require the knowledge of certain model parameters like the PCR efficiency factor.

The PCR efficiency factor, denoted by q, determines the *pace* of the chemical reaction and typically lies in the range $[0.5, 1.0)$. Rarely, in controlled conditions, q can exceed 1.0 when the environment is conducive for faster-than-normal growth of the microbe under study. In the context of Covid-19, the value of q is set to a nominal value of 0.95 [88]. However, the actual value of q depends on various factors like the probe-primers used, concentration and clustering of the virus particles in the initial mixture, and the ambient temperature, to name a few. Recall from (4.5) that q appears as a base of an exponent in the non-linear RT-qPCR model. Further, the pooled-CT value is obtained by summing up these exponential terms as dictated by the (row-) entries of the pooling matrix. Therefore, a small change in the value of q results in wide variation in the CT value. As we shall see in Section 5.4.4, setting an incorrect value of q results in the degradation of the performance of the recovery algorithms (see Table 5.3). Therefore, it is essential to design recovery algorithms that are robust to the uncertainty in the value of q. Therefore, we develop novel recovery algorithms to jointly estimate q and the individual-sample CTs (i.e., the infection status), given the pooled testing outcomes and the pooling matrix. In particular, we develop two iterative, joint-estimation algorithms: 1) Block Coordinate Descent - CT (BCD-CT) and 2) Alternating Direction Method of Multiplier - CT (ADMM-CT). As mentioned in Chapter 4, the suffix CT denotes the fact that the algorithm operates in the CT space.

5.1.1 Related Work

We refer the readers to Section 4.1.3 for a more detailed discussion of the related work. In this sub-section, we summarize the salient points relevant for this chapter. The authors in [88] convert the observed pooled-sample CT values into corresponding viral loads. Such conversion requires the knowledge of the PCR efficiency factor ($q = 0.95$ is used). The CS-based algorithms proposed in [165] implicitly use the knowledge of q to convert the system model into a form where standard CS-based algorithms can be applied. We show that the mismatch between the actual value of q and its value assumed by the decoding algorithm can result in severe performance degradation. Therefore, developing recovery algorithms that are robust to the uncertainty in the knowledge of q is essential.

5.1.2 Contributions

In this chapter, we focus on estimating the viral loads and determining healthy/sick status of individuals using the pooled RT-qPCR outcomes and the deterministic, single-stage pooling matrix when the PCR efficiency factor is unknown. The rest of the chapter is organized as below:

1. We recall the system model for the RT-qPCR in the CT-space in Section 5.2.

2. We develop two novel algorithms: 1) ADMM-CT and 2) BCD-CT, which can jointly recover the individual sample CT vector and estimate the unknown efficiency factor in Section 5.3. The sub-problem of estimating the CTs is performed using GD-CT (see Algorithm 1) or IMHT-CT (see Algorithm 2). The sub-problem involving q is solved using the projected gradient descent (PGD) method (see Section 5.3.1.1). Our recovery algorithms are robust to noise and varying machine parameters.

3. We empirically evaluate the performance of our algorithms in Section 5.4 and show that they outperform related algorithms in the literature under practical settings (e.g., unknown machine-specific parameters, CT measurement noise, etc.) in terms of the NMSE and the sparsity level up to which the algorithms guarantee zero recovery errors.

4. The advantage of using quantitative measurements in non-adaptive pool testing in terms of the testing rate and hence the cost is presented in Section 5.5.1 using publicly available data on the number of tests conducted. The results illustrate that using quantitative measurements results in significant cumulative cost savings.

5. We compile the best rates achievable for a given prevalence rate using deterministic testing matrices like Kirkman[1] [192] and Euler[2] [193] designs in Section 5.5.2. The prevalence rate estimate (e.g., the previous day's value) can be used to decide the optimum (testing) rate design for each day.

One of the main takeaways from this chapter is that our algorithms recover the unnormalized CT values without the knowledge of the machine-specific parameters. This

[1] See Appendix C.1 for a note on Kirkman designs.

[2] See Appendix C.2 for a note on Euler designs.

makes the output of our algorithms similar to that of individual testing.

5.2 System Model Summary and Focus of the Chapter

We recall the system model of the pooled RT-qPCR presented in Section 4.2, where there are n individuals participating in m pool tests, with $m \ll n$ given by

$$\begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = -\frac{1}{\log(1+q)} \begin{bmatrix} \log\left(\mathbf{A}_1^T(1+q)^{-\mathbf{u}}\right) \\ \vdots \\ \log\left(\mathbf{A}_m^T(1+q)^{-\mathbf{u}}\right) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix},$$
$$\text{or } \mathbf{c} = -\frac{1}{\log(1+q)} \log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right) + \boldsymbol{\epsilon}, \tag{5.1}$$

where $q \in \mathcal{X} \triangleq [0.5, 1.0)$ denotes the efficiency of the PCR reaction, $c_i \in \mathbb{R}$ for $i \in \{1, 2, \ldots, m\}$ denote the CT values observed in the ith pool test, $\mathbf{A} \in \{0,1\}^{m \times n}$ denote the pooling matrix, where the (i,j)th element of $\mathbf{A}$ equals 1 if the jth individual participates in the ith test, and equals 0 otherwise. Further, $\mathbf{A}_i^T$ is the ith row of $\mathbf{A}$, $u_j \in \mathbb{R}$ denotes the individual-sample CT value and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is the CT measurement noise with unknown variance σ_ϵ^2 in (5.1). Equivalently, the multiplicative noise model is written as

$$\tau(1+q)^{-c_i} = (1+q)^{\epsilon_i}\mathbf{A}_i^T\mathbf{x} \ , \quad i = 1, 2, \ldots, m. \tag{5.2}$$

The focus of this chapter is to design algorithms to solve the inverse problem of inferring the vector of individual viral loads, $\mathbf{u}$, from (5.2) given the pooling matrix $\mathbf{A}$ and the vector of pooled CTs, $\mathbf{c}$ when q is unknown.

In summary,

- Usage of binary pooling matrix and the fact that there are two kinds of pooled test outcomes: negative test outcomes have CT $= \infty$ while the positive test outcomes are non-negative finite real values, *and*

- Multiplicative and non-Gaussian nature of the noise term, $(1+q)^{\epsilon}$ as seen in (5.2) or additive nature of the noise but with a non-linear model in the log space as seen in (4.5),

makes the problem different from standard models considered in sparse signal recovery problems [175, 176]. To this end, we develop a set of novel, robust recovery algorithms to estimate the vector of individual CTs.

5.3 Recovery Algorithms: Unknown Efficiency Factor

In this subsection, we develop two algorithms whose basic constituents are Algorithm 1 (GD-CT) or Algorithm 2 (IMHT-CT) described in Chapter 4, and are capable of jointly estimating q and $\mathbf{u}$.

5.3.1 Block Coordinate Descent (BCD) - CT Algorithm

When q is unknown, we modify the overall optimization problem from (4.8) as

$$\hat{\mathbf{u}}, \hat{q} = \arg\min_{\mathbf{u},q} \frac{1}{2}\|\boldsymbol{\epsilon}\|_2^2 + \kappa \left\|(1+q)^{-\mathbf{u}}\right\|_1 \quad \text{s.t. } q \in \mathcal{X}, \tag{5.3}$$

where $\mathcal{X}$ is the convex box constraint set used to restrict the values that q can take. Since q is a scalar, the box constraint is an interval on the real line. The block coordinate descent (BCD) algorithm is well suited for problems where the coordinates or variables of optimization show *block-commonality* [194]. From (4.9), it is clear that the multiplicative factor is common across the entries of the gradient. Therefore, the maximum advantage is obtained in terms of the processing efficiency when the *block-component*s are defined as

Algorithm 3 Block Coordinate Descent (BCD) - CT

```
Input: $\mathbf{c}$, $\mathbf{A}$, $M$, $K$, $L$, $\kappa$, $\eta$ and $\beta$
Output: $\hat{\mathbf{u}}$, $\hat{q}$
Initialize $\mathbf{u}_1$ and $q_1$.
for $k \leftarrow 1$ to $M$ do
    Update
        $\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \frac{1}{2} \left\| \frac{\log\left(\mathbf{A}(1+q_k)^{-\mathbf{u}}\right)}{\log(1+q_k)} + \mathbf{c} \right\|_2^2 + \kappa \left\| (1+q_k)^{-\mathbf{u}} \right\|_1 \quad (5.4)$
        $q_{k+1} = \arg\min_{q \in \mathcal{X}} \frac{1}{2} \left\| \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}_{k+1}}\right)}{\log(1+q)} + \mathbf{c} \right\|_2^2 + \kappa \left\| (1+q)^{-\mathbf{u}_{k+1}} \right\|_1 \quad (5.5)$
end for
Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{M+1}$ and set the estimate of $q$, $\hat{q} = q_{M+1}$.
```

$\boldsymbol{\alpha} = [\mathbf{u}; q]$, where $\boldsymbol{\alpha}$ denotes the overall parameter vector. The vector $\mathbf{u}$ is considered as one block, and the scalar q is considered as the another block. The BCD algorithm for estimating the values of $\mathbf{u}$ and q in a cyclic fashion is called as the *BCD-CT* algorithm. Denoting the maximum number of iterations by M, the BCD-CT algorithm is described in Algorithm 3.

In Algorithm 3 (BCD-CT), the solution to (5.4) in each iteration can be obtained using either Algorithm 1 (GD-CT) or Algorithm 2 (IMHT-CT). When the information on whether GD-CT or IMHT-CT algorithms is used is important, we use either BCD-CT-G or BCD-CT-I, respectively.

5.3.1.1 Project Gradient Descent Algorithm for Estimating q

The projected gradient descent (PGD) algorithm is used to solve sub-problem (5.5) involving the parameter q in each of the BCD iterations. Let L denote the maximum number of iterations in the PGD algorithm, $\Pi_{\mathcal{X}}(w)$ denote the projection of the point $w \in \mathbb{R}$ into the set $\mathcal{X} \subseteq \mathbb{R}$, β be the step size and $g(q = q_{k,l}; \mathbf{u} = \mathbf{u}_k)$ denote the gradient of objective function in (5.5) with respect to q evaluated at $q = q_{k,l}$ and $\mathbf{u} = \mathbf{u}_k$. It is easy to show that $g(q;\mathbf{u})$ can be computed using

$$\begin{aligned}
g(q;\mathbf{u}) &= \frac{\partial}{\partial q} \left\| \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + \mathbf{c} \right\|_2^2 \\
&= \frac{\partial}{\partial q} \sum_{i=1}^{m} \left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i \right)^2 \\
&= \sum_{i=1}^{m} 2 \left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i \right) \frac{\partial}{\partial q} \left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} \right) \\
&= \sum_{i=1}^{m} \left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i \right) \\
&\qquad \times \left[-\frac{1}{\log(1+q)\mathbf{A}_i^T(1+q)^{-\mathbf{u}}} \sum_{j=1}^{n} \frac{A_{ij}u_j}{(1+q)^{u_j+1}} - \frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{(1+q)(\log(1+q))^2} \right] \\
&\quad - \sum_{j=1}^{n} \frac{\kappa u_j}{(1+q)^{u_j+1}}.
\end{aligned} \tag{5.6}$$

The PGD algorithm is described in Algorithm 4. The second derivative, $H(q)$, of the cost function in (5.3) w.r.t. q is

$$\begin{aligned}
H(q) = \sum_{i=1}^{m} & \left[\frac{1}{\log(1+q)\ \mathbf{A}_i^T(1+q)^{-\mathbf{u}}} \sum_{j=1}^{n} \frac{A_{ij}u_j}{(1+q)^{u_j+1}} + \frac{\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{(1+q)(\log(1+q))^2} \right]^2 \\
& + \left(c_i + \frac{\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{\log(1+q)} \right) \\
& \times \left[\frac{\sum_{j=1}^{n} A_{ij}u_j(u_j+1)(1+q)^{-(u_j+2)}}{\log(1+q)\ \mathbf{A}_i^T(1+q)^{-\mathbf{u}}} + \frac{\left(\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})\right)^2 (\log(1+q)+2)}{(1+q)^2(\log(1+q))^3} \right. \\
& \left. + \frac{\left(\sum_{j=1}^{n} A_{ij}u_j(1+q)^{-(u_j+1)}\right)(\log(1+q)+1)}{(1+q)(\log(1+q))^2\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)} - \frac{\left(\sum_{j=1}^{n} A_{ij}u_j(1+q)^{-(u_j+1)}\right)^2}{\log(1+q)\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)^2} \right]
\end{aligned}$$

$$+\sum_{j=1}^{n}\frac{\kappa u_j(u_j+1)}{(1+q)^{(u_j+2)}}. \tag{5.7}$$

Let $q \in \mathcal{X} \triangleq [m_q, M_q),\ m_q < M_q < \infty$. In a positive test where amplification occurs in each PCR cycle, $m_q > 0$. When $u_j < \infty$, $H(q)$ is upper bounded as

$$\begin{aligned} H(q) \leq L_1' \triangleq m &\left[\frac{nM_u}{\log(1+m_q)(1+M_q)^{-M_u}(1+m_q)^{m_u+1}} + \frac{\gamma'}{(1+m_q)(\log(1+m_q))^2}\right]^2 \\ &+ m\left(c_{\max} + \frac{\gamma'}{\log(1+m_q)}\right) \\ &\times\left[\frac{n(M_u{+}1)^2(1{+}m_q)^{-(m_u+2)}}{\log(1+m_q)(1+M_q)^{-M_u}} + \frac{(\gamma')^2(\log(1{+}M_q){+}2)}{(1+m_q)^2(\log(1+m_q))^3}\right. \\ &\left.+\frac{nM_u(1+m_q)^{-(m_u+1)}(\log(1+M_q)+1)}{(1+m_q)(\log(1+m_q))^2(1+M_q)^{-M_u}}\right] + \frac{n\kappa M_u(M_u+1)}{(1+m_q)^{(m_u+2)}}, \end{aligned} \tag{5.8}$$

where $\gamma' \triangleq \max\left(|\log(1+M_q)^{-M_u}|, |\log(n(1+m_q)^{-m_u})|\right)$ and when $u_j = \infty$, as

$$\begin{aligned} H(q) \leq L_2' \triangleq m &\left[\frac{\gamma'}{(1+m_q)(\log(1+m_q))^2}\right]^2 \\ &+ m\left(c_{\max} + \frac{\gamma'}{\log(1+m_q)}\right)\left[\frac{(\gamma')^2\left(\log(1+M_q)+2\right)}{(1+m_q)^2(\log(1+m_q))^3}\right]. \end{aligned} \tag{5.9}$$

In summary, $H(q) \leq L' \triangleq \max(L_1', L_2') < \infty$. Let the step size, β, be chosen such that $0 < \beta < 1/L'$. Using Theorem 4.1 for the sequence of scalar iterates $\{q_k\}$, in the context of gradient descent update step in (5.10), we can conclude that the iterates converge to a local minimizer, q^*.

5.3.2 Alternating Direction Method of Multipliers (ADMM) - CT Algorithm

The alternating direction method of multiplier (ADMM) procedure is another well-known parallel or distributed optimization framework [195]. The essence of the ADMM recipe lies in using the advantages of the *dual formulations* and the *augmented Lagrangian* [195].

Algorithm 4 Projected Gradient Descent (PGD)

```
Input: $\mathbf{c}$, $\mathbf{A}$, $\mathbf{u}_k$, $q_k$, $L$ and $\beta$
Output: $q_{k+1}$
Initialize $q_{k,1} = q_k$.
for $l \leftarrow 1$ to $L$ do
    Compute $g(q_{k,l}; \mathbf{u}_k)$ using (5.6).
    Update
        $q_{k,l+1} = \Pi_{\mathcal{X}}\left(q_{k,l} - \beta g(q_{k,l}; \mathbf{u}_k)\right).$    (5.10)
end for
Set $q_{k+1} = q_{k,L+1}$.
```

A modification to (5.3) on these lines yields the required ADMM optimization problem, given by

$$\hat{\mathbf{u}}, \hat{\mathbf{w}}, \hat{q} = \underset{\mathbf{u},\mathbf{w},q}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{\epsilon}\right\|_2^2 + \kappa\left\|(1+q)^{-\mathbf{w}}\right\|_1 + \frac{\rho}{2}\left\|\mathbf{u} - \mathbf{w} + \boldsymbol{\mu}\right\|_2^2 \ \text{ s.t. } q \in \mathcal{X}, \tag{5.11}$$

where ρ is the penalty parameter, $\boldsymbol{\mu}$ denotes the dual variable and $\mathbf{w}$ is the auxiliary variable. The dual variable is the result of the re-parameterization which converts the standard ADMM problem into its *scaled dual form* [195]. The advantage of the dual formulation, along with the auxiliary variable, is that it allows for the main optimization problem to be decomposed into multiple, simpler sub-problems which are computationally efficient to solve. For example, from (5.11), at least 3 sub-problems involving $\mathbf{u}$, $\mathbf{w}$ and q can be observed. However, dual methods may have slow convergence rates. A penalty term, $\frac{\rho}{2}\left\|\mathbf{u} - \mathbf{w} + \boldsymbol{\mu}\right\|_2^2$, is added to convert the primal problem into a strongly convex objective to overcome this issue. Lastly, from (5.11), we note that $\boldsymbol{\mu} \rightarrow \mathbf{0}$ and $\hat{\mathbf{w}} = \hat{\mathbf{u}}$ on convergence.

Let N denote the maximum number of iterations. The ADMM algorithm for estimating $\mathbf{u}$ and q given $\mathbf{c}$ and $\mathbf{A}$, called *ADMM-CT* algorithm is described in Algorithm 5.

Algorithm 5 Alternating Direction Method of Multipliers (ADMM) - CT

Input: $\mathbf{c}$, $\mathbf{A}$, N, K, L, κ, η and β

Output: $\hat{\mathbf{u}}$, $\hat{q}$

```
1: Initialize $\mathbf{u}_1$, $\mathbf{w}_1$, $\boldsymbol{\mu}_1$ and $q_1$.
2: for $k \leftarrow 1$ to $N$ do
3:     Update
```

$$\begin{aligned}
\mathbf{u}_{k+1} &= \arg\min_{\mathbf{u}} \frac{1}{2}\left\|\frac{\log\left(\mathbf{A}(1+q_k)^{-\mathbf{u}}\right)}{\log(1+q_k)} + \mathbf{c}\right\|_2^2 \\
&\quad + \frac{\rho}{2}||\mathbf{u} - \mathbf{w}_k + \boldsymbol{\mu}_k||_2^2 \\
q_{k+1} &= \arg\min_{q\in\mathcal{X}} \frac{1}{2}\left\|\frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}_{k+1}}\right)}{\log(1+q)} + \mathbf{c}\right\|_2^2 \\
&\quad + \kappa\left\|(1+q)^{-\mathbf{w}_k}\right\|_1 \\
\mathbf{w}_{k+1} &= \arg\min_{\mathbf{w}} \kappa\left\|(1+q_k)^{-\mathbf{w}}\right\|_1 + \frac{\rho}{2}||\mathbf{u}_k - \mathbf{w} + \boldsymbol{\mu}_k||_2^2 \qquad (5.12) \\
\boldsymbol{\mu}_{k+1} &= \boldsymbol{\mu}_k + (\mathbf{u}_{k+1} - \mathbf{w}_{k+1}).
\end{aligned}$$

```
4: end for
5: Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{N+1}$ and set the estimate of $q$, $\hat{q} = q_{N+1}$.
```

In Algorithm 5, we note that the sub-problem in (5.12) is convex in $\mathbf{w}$. Hence, it can be solved using any convex optimization package (e.g., CVX [196]). As stated earlier, the sub-problem involving optimization over $\mathbf{u}$ can be solved using GD-CT (Algorithm 1) with a term $\rho(\mathbf{u} - \mathbf{w} + \boldsymbol{\mu})$ added to the gradient in (4.9). Similarly, the sub-problem of estimating q_{k+1} can be solved using Algorithm 4.

5.3.3 Comparison Between ADMM-CT and BCD-CT Estimates

We illustrate the advantage of the ADMM-CT algorithm over the BCD-CT algorithm (with GD-CT algorithm for optimizing $\mathbf{u}$) in terms of the *quality* of the final solution in Figure 5.1. For the current discussion, the quality of the final solution is measured as the *gap* between the estimates of the individual CTs for the positive and negative samples.

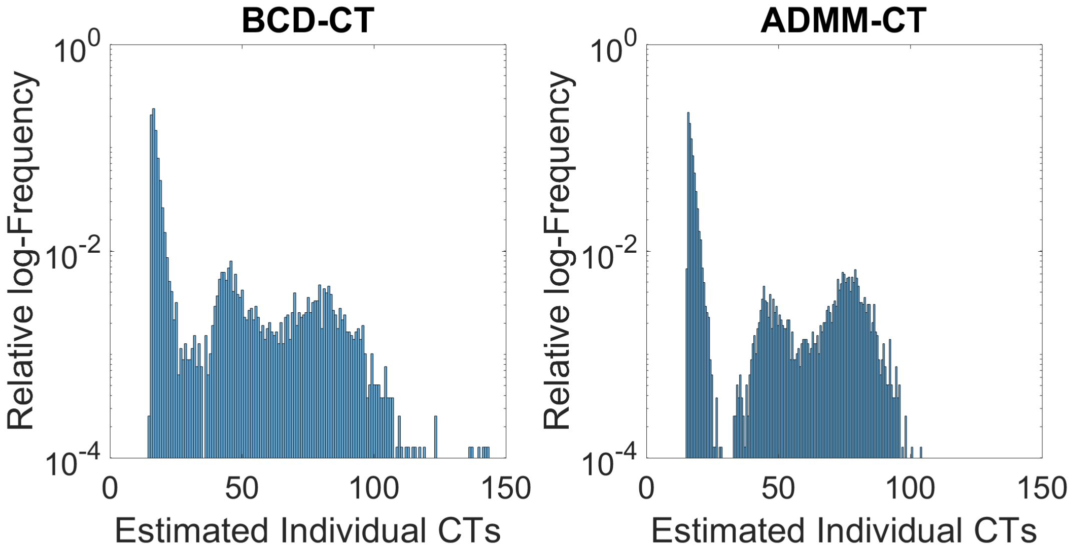


Figure 5.1: Comparison of ADMM-CT and BCD-CT algorithms in terms of the distribution of the estimated individual CTs.

As mentioned in Section 4.1.1, the individual CTs for a positive sample is finite and is between $15 - 35$. However, for a negative sample, the individual CT is ∞. Therefore, from the perspective of the recovery algorithms, a larger margin between the estimate of the individual CTs for positive and negative samples is desirable. In addition, a larger margin translates to the robustness of the algorithm to the choice of the decision thresholds. From Figure 5.1, we see that there is a well-defined and more prominent margin between the individual CT estimates of the positive and negative samples obtained using the ADMM-CT algorithm compared to the estimates obtained from the BCD-CT algorithm.

5.3.4 Convergence to Local Minima

In the optimization problems described above, due to the presence of log and exponential-to-q terms, the objective function is non-convex in $\mathbf{u}$. To overcome the problem of convergence to the local minima, the BCD-CT and ADMM-CT algorithms are run multiple

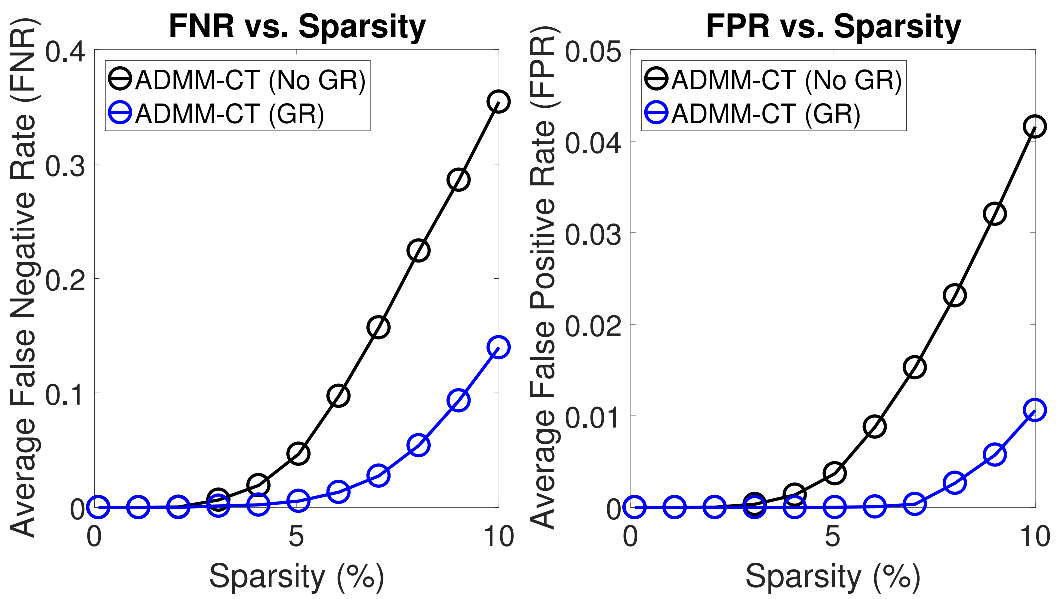


Figure 5.2: Comparison of ADMM-CT algorithm with and without Gaussian randomization to overcome local optima.

times with *Gaussian randomization* (GR) i.e., the algorithm is run with different initial conditions and the estimates with the lowest objective function value are chosen as the final estimates of q and $\mathbf{u}$ [197]. The term *Gaussian* in GR refers to the fact that the initial samples are drawn from a Gaussian distribution with certain mean and variance. However, our simulations show that the performance of our recovery algorithms is insensitive to the specific distribution used for randomization (see Section 5.4 for more details). The FPR and FNR performance improvement obtained from GR in the ADMM-CT algorithm is shown in Figure 5.2.

We note that the complexity of the algorithms is not a major concern, as the RT-qPCR test itself takes several hours to run. All of the recovery algorithms presented here run in a few seconds on a computer, and therefore their relative complexity is not important for this application.

Table 5.1: Recap of the Parameters Used in the Experimental Setup from Chapter 4

Parameter Name	Parameter Value
Kirkman Pooling Matrix Size, $m \times n$	45×105
CT Decision Threshold	45
Maximum GD-CT/IMHT-CT Iterations, K	500
Threshold on CT in IMHT-CT, u_{th}	100
Regularization Parameter, κ	10^{-3}
GD-CT/IMHT-CT Step Size, η	0.01

5.4 Numerical Simulations

In this section, we present the simulation results showing the effectiveness of our algorithms for recovering the infected samples given the outcomes of group testing and the pooling matrix when the PCR efficiency factor is unknown. The comparison is performed across two scenarios: firstly, under the various noise standard deviations, $\sigma_\epsilon \in \{0, 3.0\}$ [177] at $q = 0.95$ and secondly, under different efficiency factors, $q \in \{0.5, 0.95\}$ at $\sigma_\epsilon = 0$. In addition, the algorithms are run with Gaussian randomization, as mentioned earlier. Lastly, the performance of the developed algorithms in compared with standard CS-based methods, Algorithm 1 (GD-CT) and binary group testing methods from the literature [4,5,88,165]. We describe the experimental setup in the next subsection.

5.4.1 Experimental Setup

The experimental setup used to generate the results is similar to that used in Chapter 4. We recap the parameter setting for easy reference in Table 5.1.

The indices of nonzero viral loads in $\mathbf{x} \in \mathbb{N}^{105\times1}$ are picked uniformly at random and stored in an index set $\mathcal{NZI}$ with cardinality $|\mathcal{NZI}| = \lceil ns_x/100 \rceil$, where s_x (%) is the prevalence rate. The viral load in a negative sample is zero. The positive entries of the initial viral-load vector, $[\mathbf{x}]_j$, are drawn from the Poisson(λ_j) distribution where $\lambda_j \sim$ Unif$(100, 10^6)$, for $j \in \mathcal{NZI}$. Finally, using the relationship in (5.2), the pooled CT vector, $\mathbf{c} \in \mathbb{R}_+^{45\times 1}$ is generated. In the distribution of the hyper-parameter λ_j, the lower and upper limits are the average least and highest viral load (or equivalently, cDNA/ RNA particles) in a positive sample.

In the initialization step of the algorithms, the starting points $\mathbf{u}_1$ and $\mathbf{w}_1$ are drawn i.i.d. from a Laplace distribution with mean 75 and shape parameter 10 and $q_1 \sim$ Unif$(\mathcal{X})$. As mentioned in Section 4.5.1, the distribution function and the associated parameters used do not affect the results much. For example, choosing Normal$(50, 10)$, Laplace$(50, 10)$, Unif$(10, 100)$ etc. give similar results. Finally, the ADMM-CT penalty parameter, $\rho = 0.01$, the regularization parameter, $\kappa = 10^{-3}$, and the step size in the PGD algorithm is set as 0.01 each are chosen via cross-validation.

All the results are averaged over $10,000$ Monte-Carlo runs. We compare the performance using the average NMSE in the CT values recovered, the average FNR, and the average FPR metrics for the prevalence rate (sparsity) up to 10%.

5.4.2 Algorithm Performance at Different Noise Levels

We illustrate the NMSE performance of the BCD-CT-G, BCD-CT-I and ADMM-CT algorithms in Figure 5.3 as the sparsity (%) is varied from 0.1% to 10% over two different values of the noise standard deviations, σ_ϵ. The NMSE performance of the ADMM-CT algorithm is the best, followed by that of BCD-CT-G and BCD-CT-I algorithms. Although ADMM-CT shows a nonzero NMSE at low sparsity levels, there is no effect on the FPR and FNR performances, as we shall see later. Further, from Figure 5.3, the NMSE curves shift up as the noise level increases. As the number of infected sample in the

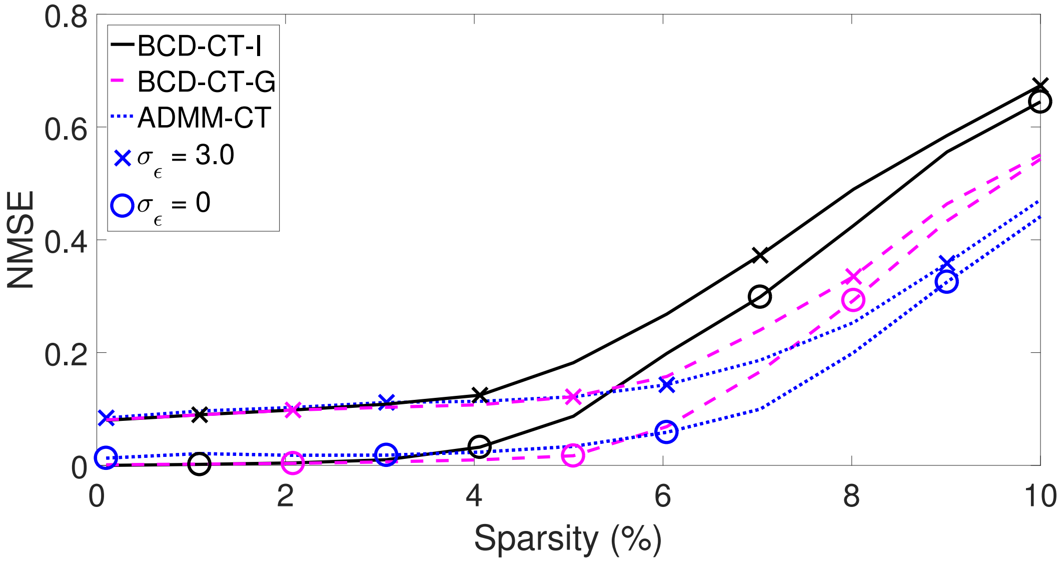


Figure 5.3: Comparison of the average NMSE across ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms over different sparsity (%) and noise levels.

population increases, we observe that the NMSE increases at faster rate and the NMSE curves corresponding to the noiseless (i.e., $\sigma_\epsilon = 0$) and noisy (i.e., $\sigma_\epsilon = 3.0$) case come closer to each other. Similar observation was made in Section 4.5.2 where this effect was attributed to the fact that the decoding capability of the recovery algorithm depends on the test matrix design. Therefore, the decoding process fails irrespective of noise level and the recovery algorithm estimates get worse.

The FNR and FPR performances of the binary group testing algorithms, namely, COMP and DD along with the BCD-CT-G, BCD-CT-I and ADMM-CT algorithms are shown in Figure 5.4 at two noise standard deviations as the sparsity (%) is varied from 0.1% to 10%. We can observe from the subplot Ⓐ in Figure 5.4 that the FNR performance of ADMM-CT, BCD-CT-I, and BCD-CT-G are similar. The FPR of the ADMM-CT algorithm is the best, followed by BCD-CT-G and BCD-CT-I algorithms, as seen in the subplot Ⓑ across the noise levels. Further, the FPR performance of the BCD-CT-G is

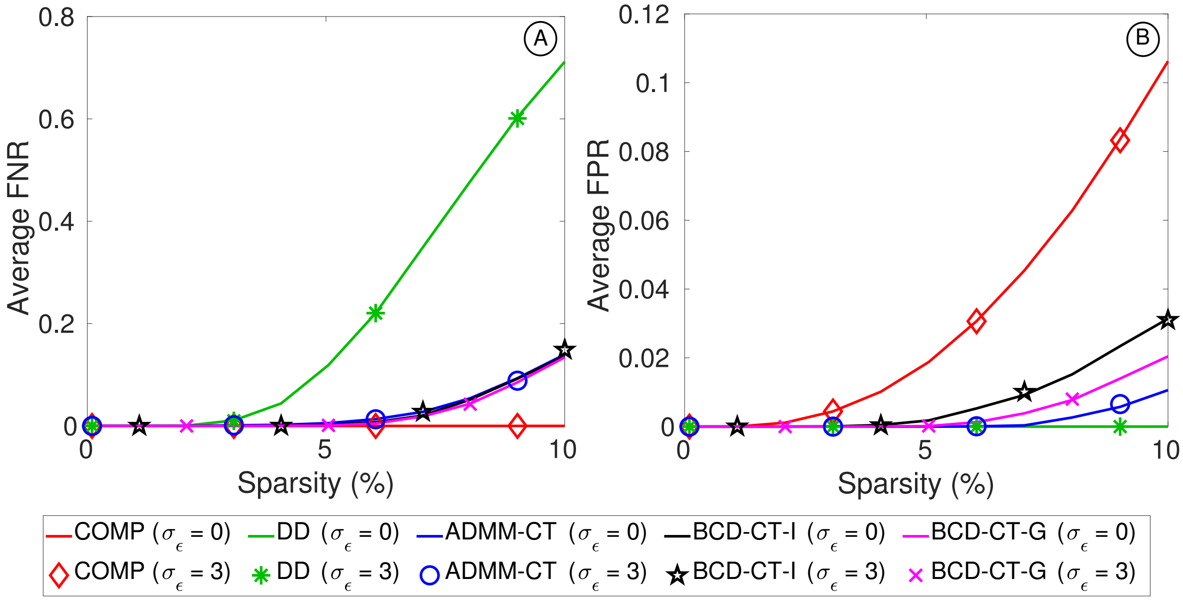


Figure 5.4: Comparison of the average FNR and average FPR across COMP, DD, ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms over different sparsity (%) at different noise levels.

better than that of the BCD-CT-I algorithm reflecting the similar behavior which was observed between the GD-CT and IMHT-CT algorithms in Figure 4.2. The ADMM-CT approach is able to match the FNR of COMP while simultaneously matching the FPR of the DD algorithms. Although COMP and DD optimize for FNR and FPR, respectively, the corresponding FPR and FNR performances are poor. Thus, this highlights the advantage of quantitative measurement-based algorithms over binary pool testing algorithms like COMP and DD. Also, pooled CT-based algorithms can estimate the individual CTs, which could be independently valuable in practice.

5.4.3 Algorithm Performance at Different PCR Efficiency Factors

In this subsection, we discuss the performance of the developed algorithms across the PCR efficiency factors. The NMSE performance of the BCD-CT-G, BCD-CT-I and

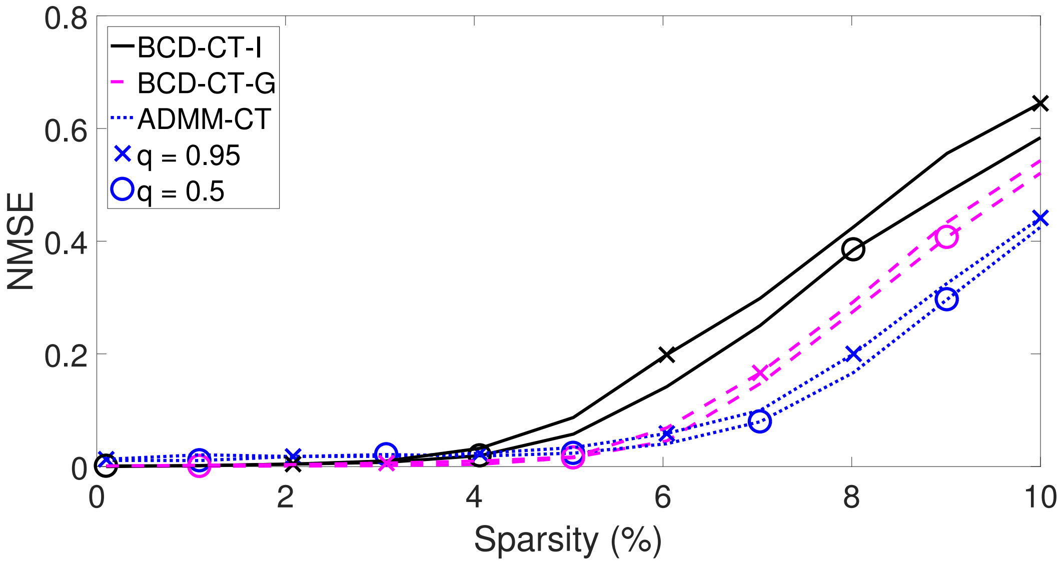


Figure 5.5: Comparison of the average NMSE across ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms for different sparsity (%) and efficiency factors.

ADMM-CT algorithms are shown in Figure 5.5 as the sparsity (%) is varied from 0.1% to 10% over two different values of the PCR efficiency factor, q. From Figure 5.5, the NMSE performance of the ADMM-CT algorithm is the best, followed by that of BCD-CT-G and BCD-CT-I algorithms. In addition, we see that the performance at $q = 0.5$ is slightly better with relatively notable difference for the BCD-CT-I algorithm. Thus, in line with our previous observations from Figure 4.3, the IMHT based algorithms exhibit a wider variation in performance across different values of q compared to the GD based algorithms, even when q is unknown and is estimated from the observed CT values.

We define the mean squared error between q and its estimate, $\hat{q}$, as $\text{MSE}_q \triangleq \mathbb{E}[|q - \hat{q}|^2]$. When the sparsity $< 5\%$, we observed that $\text{MSE}_q \approx 0.052, 0.101$ and 0.26 for the $\sigma_\epsilon = 0, 1.0$ and 3.0, respectively. Also, while small errors in the estimate of q do not significantly affect the algorithm's FPR or FNR, larger errors cause the algorithm to make FP or FN errors at lower sparsity levels (see Table 5.3).

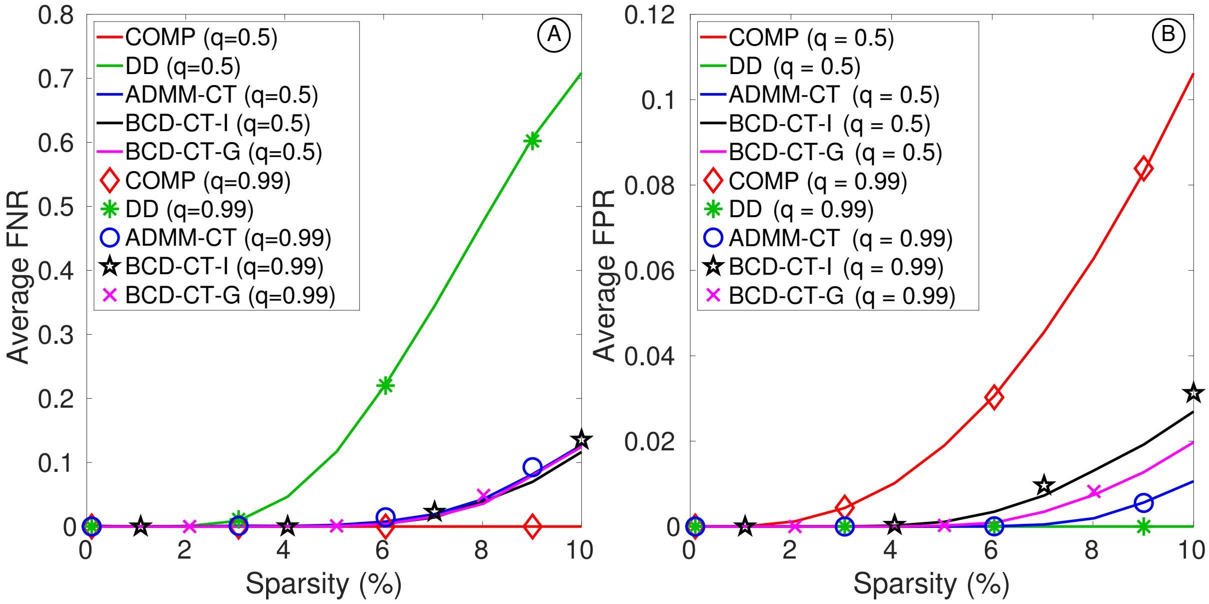


Figure 5.6: Comparison of the average FNR and average FPR across COMP, DD, ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms over different sparsity (%) at different efficiency factors.

We compare the FNR and FPR performances of algorithms in Figure 5.6 across various efficiency factors. We include the performance of binary measurement-based algorithms: COMP and DD. We can observe from the subplot Ⓐ in Figure 5.6 that the FNR of ADMM-CT, BCD-CT-I, and BCD-CT-G are similar. The FPR of the ADMM-CT algorithm is the best, followed by BCD-CT-G and BCD-CT-I algorithms, as seen in the subplot Ⓑ across the PCR efficiency factors. Finally, we observe the benefit of using quantitative measurement-based algorithms over binary measurements across different PCR efficiency factors too.

5.4.4 Comparison with CS-based Methods: Robustness to Uncertainty

In this final subsection, we compare the performance of the ADMM-CT algorithm with the existing CS-based algorithms in the literature [88,165]. These previous studies directly apply CS-based recovery techniques (with or without an initial COMP stage) to the observation model in (5.2). Certain CS-based algorithms like Linear Absolute Shrinkage and Selection Operator (LASSO) are known to yield the optimal solution even in the non-linear regime, under certain conditions [198, 199]. In particular, the authors in [198] show that when the entries of the pooling matrix are drawn i.i.d. from $\sim \mathcal{N}(0,1)$, the performance of LASSO with non-linear measurements characterized by a non-linear mapping $g(\cdot)$ is asymptotically the same as if the measurements were of the linear form $c_i = \mu \mathbf{A}_i^T \mathbf{x} + \sigma \epsilon_i$, where $\mu \triangleq \mathbb{E}[\bar{\gamma} g(\bar{\gamma})]$, $\sigma \triangleq \mathbb{E}[(g(\bar{\gamma}) - \mu\bar{\gamma})^2]$ and $\bar{\gamma} \sim \mathcal{N}(0,1)$. Similarly, the authors in [199] derive upper and lower bounds for the MSE under a Poisson measurement model and when the entries of the pooling matrix satisfy certain boundedness conditions. The simulation results in this sub-section shows that our approach is better under the non-asymptotic regime with a deterministic binary pooling matrix. Note that, in order to use (5.2) directly with CS-based algorithms, one needs to assume a nominal value of q to compute the left hand side of the measurement equation of the observed CT values [88]. We also mention that one can use cross-validation to choose the value of q, instead of assuming a nominal value.

We demonstrate the robustness of the ADMM-CT algorithm in the following two aspects. The first one is the robustness to the CT measurement noise. The second aspect is the robustness to the unknown efficiency factor, q. In addition, the performance of the best quantitative group testing algorithm from Chapter 4, i.e., GD-CT is also shown. We denote the efficiency factor used for the CT-to-viral-load conversion by q_{conv}, which could be different from the actual q of the PCR process. We consider the minimum sparsity

Table 5.2: Comparison of k_e Across Recovery Algorithms at Various Noise Variances

Noise σ_ϵ	0 (Noiseless)	0.1	0.5	1.0
NN-LASSO	3	1	1	1
NN-LS	3	1	1	1
NN-LAD	4	2	1	1
NN-OMP	4	1	1	1
SBL	8	4	2	1
COMP	2	2	2	2
DD	2	2	2	2
GD-CT	4	4	4	4
ADMM-CT	4	4	4	4

percentage (denoted by k_e) at which the recovery algorithm makes at least one FP or FN out of $10,000$ experiments as our metric for comparison. Table 5.2 summarizes the values of k_e for each of the recovery algorithms considered, at different noise (σ_ϵ) levels.

In CS-based algorithms, a threshold is fixed, below which the estimates are set to 0. We fix the threshold to balance out the FP and FN errors. Further, the hyper-parameters (regularization constants etc.) required by the CS based algorithms are chosen via cross-validation [88, S.III]. We compare against the non-negative LASSO (NN-LASSO), NN least squares (NN-LS), NN least absolute deviation (NN-LAD), NN orthogonal matching pursuit (NN-OMP) and sparse Bayesian learning (SBL) algorithms proposed for RT-PCR based pool testing for Covid-19 in the literature [88, 165]. In addition, we include the performance of the binary group testing algorithms like COMP and DD. From Table 5.2, we note that as the noise level increases, k_e obtained by CS-based algorithms degrades. In particular, at practical noise levels [177], ADMM-CT has a clear advantage over the CS-based algorithms, even though SBL outperforms ADMM-CT in the noiseless case.

On the other hand, COMP and DD show similar performance across the noise variances under consideration, but perform worse than our algorithms. Recall from (4.6) that the binary pooled test outcome is set to unity if a finite pooled-CT value is observed in that pooled test and is set to zero otherwise. Note that an additive noise with variance can not make a finite pooled-CT value infinity or vice-versa. Thus, the precise pooled-CT values, noisy or otherwise, is lost in the binary test outcomes. Therefore, the performances of COMP and DD are unaffected by the noise levels.

Finally, the performance of the GD-CT algorithm is similar to that of the ADMM-CT algorithm across the noise levels. Recall that GD-CT solves the sub-problem involving the estimation of $\mathbf{u}$ in Algorithm 5 and uses the value of q as the input. As we shall see below, the advantage of ADMM-CT over the GD-CT algorithm arises in more practical case when q is unknown.

Next, we summarize the effect of mismatch between the q and q_{conv} in Table 5.3. Similar to the noise performance analysis, we use the metric k_e to demonstrate the robustness of different algorithms. For this purpose, we fix the $q_{\text{conv}} = 0.95$ as in [88] and vary the actual $q \in \{0.99, 0.95, 0.9, 0.8, 0.65, 0.5\}$ of the RT-qPCR process [166, 177]. From Table 5.3, we note that as the mismatch between q and q_{conv} increases, the performance of the CS-based algorithms and the GD-CT algorithm degrades. The ADMM-CT algorithm estimates both $\mathbf{u}$ and q, and hence, the parameter q_{conv} is not used by it. The performance of the binary group testing algorithms, namely, COMP and DD remains same across the range of q using which the data is generated. This is due to the fact that the pooled binary test outcomes are obtained using (4.6) which do not depend on the value of q. However, the performance of COMP and DD in terms of k_e is worse than that of our algorithms.

Table 5.3: Comparison of k_e Across Recovery Algorithms at Various Process Efficiency Factors, q, at Noise $\sigma_\epsilon = 0.1$ and $q_{\text{conv}} = 0.95$

q	0.99	0.95	0.9	0.8	0.65	0.5
NN-LASSO	1	1	1	1	1	1
NN-LS	1	1	1	1	1	1
NN-LAD	1	2	1	1	1	1
NN-OMP	1	1	1	1	1	1
SBL	4	4	4	3	3	3
COMP	2	2	2	2	2	2
DD	2	2	2	2	2	2
GD-CT	3	4	3	2	2	2
ADMM-CT	4	4	4	4	4	4

Although SBL is robust to mismatch in q, the ADMM-CT still has an advantage over the existing algorithms. We also observed that the NMSE in the CT values returned by SBL are of the order 0.5, while that for ADMM-CT are of the order $0.01 - 0.04$ at $q = 0.99$ and $k_e = 4$. Thus, the SBL algorithm performs poorer than ADMM-CT in terms of estimating the individual CT values. Similar observations hold for GD-CT, where its NMSE is of the order 0.38 at $q = 0.5$ and $k_e = 4$. The ADMM-CT algorithm is robust across the values of PCR efficiency factors observed in practice, and outputs the individual CTs with low NMSE.

In summary, our algorithms are robust to the CT measurement noise levels observed in practical RT-qPCR. Also, they do not require knowledge of the machine-specific parameters, and the performance is similar across the spectrum of PCR (amplification) efficiency factors seen in practice.

5.5 Pool Testing Case Study: Covid-19

The discussion in this section presents empirical results from the case study. The discussion aims at bridging the gap between the theoretical results, their implications etc. and the needs of the practitioners like lab technicians, committee and government bodies which takes decisions on the state/ province/ country's testing protocols and strategies, workforce health-monitoring and screening committees in any public or private organizations, to name a few. In particular, we address the following two questions:

1. Is group testing (or pool testing) useful in practice? in Section 5.5.1.

2. How do one read the theoretical results and use them in practice? in Section 5.5.2.

5.5.1 Advantage of Pool Testing: An Empirical Evaluation Using Covid-19 Data

We start by showing the advantage of using pool testing from the publicly available history of Covid-19 test numbers. Although we focus on non-adaptive pool testing with quantitative measurements in this work, in this experiment, we include one adaptive pool testing and two non-adaptive settings: a binary model and our quantitative model. We consider data from six Indian states: Karnataka, Kerala, Tamil Nadu, Maharashtra, Delhi, and Uttar Pradesh. The data used are primarily sourced by the Indian Council of Medical Research (ICMR), New Delhi, India,[3] and aggregated by a third-party website: https://www.covid19india.org/. As stated in Section 4.1, the symptomatic percentage is set equal to 20%. The number of primary contacts per symptomatic individual are considered to be 4. These estimates can be further refined using population density

[3] ICMR: https://www.icmr.gov.in/

information, data collected from contact tracing applications, etc. Further, the primary contacts of the symptomatic individuals are assumed to be more likely (by a factor of 4) to have the disease. In our analysis, following the standard protocols and triage processes, pool-testing is applied only on asymptomatic individuals and non-primary contacts; symptomatic individuals and primary contacts are tested individually. Dorfman adaptive testing with optimum pool size requires $2\sqrt{(n'k')}$ tests, where n' represents the number of non-symptomatic and non-primary contacts tested, and k' denotes the number of positive cases who are non-symptomatic and non-primary contacts [7]. In addition, we compute the counting bound, which is a lower bound on the number of tests under a binary testing model, as $\mathrm{CB} = k' \log_2(\frac{n'}{k'})$ [7]. From [7], the DD algorithm with near-constant column weight pooling matrix design requires roughly $\frac{\mathrm{CB}}{0.45}$ tests for identifying all the sick individuals.

Finally, the total number of tests required by the non-adaptive pool-testing method using deterministic matrix designs like Kirkman and Euler with quantitative measurements is computed empirically, as follows. A set of pooling matrix designs are compiled in $\mathcal{M} = \{\mathrm{Euler} - (15 \times 25), \ldots, \mathrm{Kirkman} - (45 \times 285), \ldots, \mathrm{Kirkman} - (93 \times 1240), \mathrm{Euler} - (361 \times 6859), \ldots\}$. We consider 26 Kirkman and 26 Euler-based design matrices. Thus, the cardinality of the set is $|\mathcal{M}| = 52$. The set is further extended as follows: Denote the *testing rate* of a pooling matrix of size $R \times C$ as $\xi = R/C$ where, C denotes the number of individuals tested, and R denotes the number of tests. Smaller column-truncated matrices are constructed by dropping the last few columns [88] to obtain matrices with rates $[\xi_{\text{round}} : 0.1 : 0.9]$ where ξ_{round} is the value obtained by rounding up 10ξ to the next integer and then dividing by 10. That is, if $\xi = 0.33$, $\xi_{\text{round}} = 0.4$. In this way, we obtain a total of

474 deterministic test matrices.

For each pooling matrix in the set $\mathcal{M}$, a 1000 Monte Carlo run experiment determines the maximum prevalence rate post which our ADMM-CT recovery algorithm makes either a FP or a FN error. This maximum prevalence rate is denoted by k_e for the given pooling matrix and is added into a look-up table. The non-symptomatic and non-primary contact prevalence rate is computed using the parameters described above. The best rate matrix design is the matrix with the lowest rate whose k_e exceeds the given prevalence rate. The rate thus obtained multiplied by n' gives the number of tests needed if non-adaptive pool-testing with our recovery algorithms and the optimum test matrix from the set $\mathcal{M}$ is used.

We then add the individual tests conducted on symptomatic and on primary contacts to obtain the total number of tests required by each of the three methods: Dorfman with optimal pool size (*Optimized Dorfman*), DD with the near-constant column weight design (*Achievable, DD*) and our approach (the ADMM-CT algorithm). The comparison of the cumulative number of tests from April 2020 till July 2021 is shown in Figure 5.7 for all the 6 Indian states, along with the prevalence rate trend over the same duration.

From Figure 5.7, the cumulative number of tests required by using non-adaptive pool testing with quantitative measurements and the ADMM-CT algorithm for recovery (blue curve in Figure 5.7) is lower than that obtained by using non-adaptive pool testing with binary measurements (black curve), and this is further better than the adaptive testing numbers (green curve). Finally, pool testing methods have significant advantage compared to individual testing (red curve) when the prevalence rates are low (also see Figure 5.9 and the associated discussion.) The total number of tests saved translates to (resource)

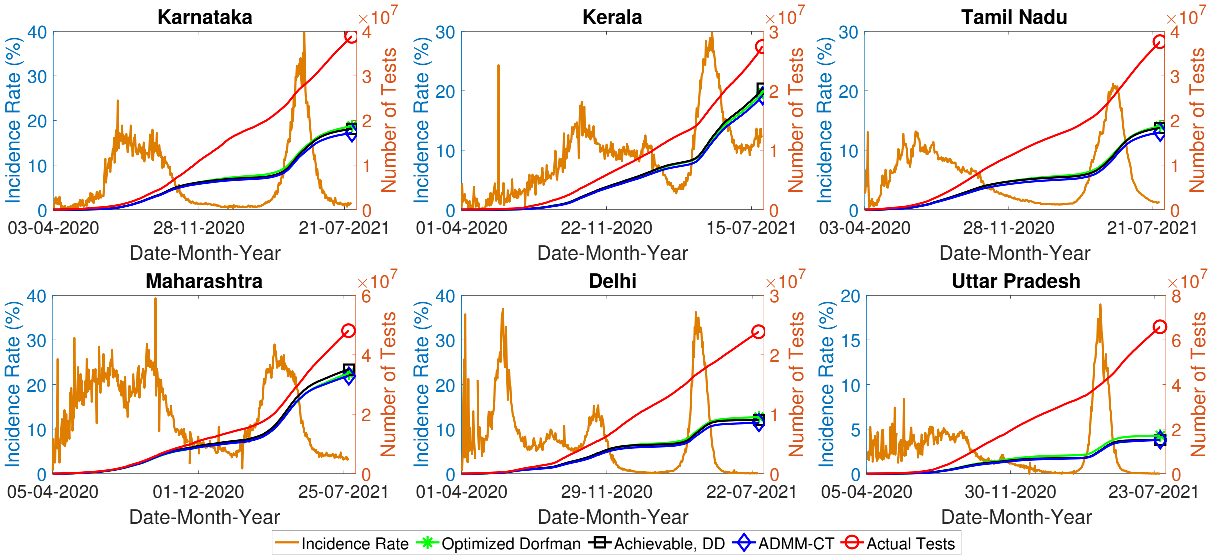


Figure 5.7: The prevalence rate trend and the comparison of the cumulative number of tests required by *Optimized Dorfman*, *Achievable, DD* and *ADMM-CT* algorithms for 6 Indian states along with actual cumulative tests conducted from April 2020 till July 2021.

cost and time saving. For instance, using the numbers for the state of Karnataka, the cumulative number of tests saved over the past 1.5 years, if a pool testing method (e.g., the ADMM-CT approach) is used compared to the individual testing is $\sim$ 21.6 million tests. Using the nominal cost per RT-qPCR test as $\approx$ 14 USD, we obtain a cost-saving of $\sim$ 302.46 million USD. Therefore, pool testing methods have a significant advantage as compared to individual testing. Further, our ADMM-CT approach requires $\sim$ 1.55 million, and $\sim$ 0.93 million fewer tests than the optimized Dorfman and the achievable tests required by DD approaches, respectively. Under the above-mentioned RT-qPCR cost estimate, the cost savings obtained by using the ADMM-CT approach instead of optimized Dorfman and the tests required by DD are $\sim$ 21.78 million and $\sim$ 13.05 million USD, respectively. Thus, our approach is better in terms of the cost savings among the other pool testing methods considered.

5.5.2 A Practical Pool Test Protocol

Based on the previous discussion about the benefits of non-adaptive group testing with quantitative measurements, in this sub-section, we address the following aspects:

1. We demonstrate that using the local prevalence rates to design the tests is useful in practice, by using the actual testing numbers and positivity rates from 6 different states of India during different stages of the pandemic. The advantage of the using local prevalence rates was also discussed in [200].

2. We empirically characterize the testing rates achievable using deterministic pooling matrices for different prevalence rates. That is, we provide an insight into the question of which deterministic pooling matrix should be chosen to guarantee a near-zero errors at each prevalence rate.

3. We also collect empirical results on the prevalence rates at which adaptive, non-adaptive binary measurements-based, and non-adaptive quantitative measurements-based group testing perform the best, in terms of the testing rates achieved.

The authors in [200] give a lower bound on the number of tests required given the heterogeneity profile (i.e., prevalence rate, risk profile, contact maps etc.) of the local population. In particular, they focus on the two-stage group testing algorithms like Bernoulli sampling, Dorfman, constant tests per sample, the doubly constant algorithm etc., with random pooling matrices. Our work complements this approach, since we focus on single-stage group testing with deterministic pooling matrices.

To this end, first, we empirically show that using the *local* (e.g., state-wise) prevalence rate to select the pooling matrix is advantageous over using the *global* prevalence rate

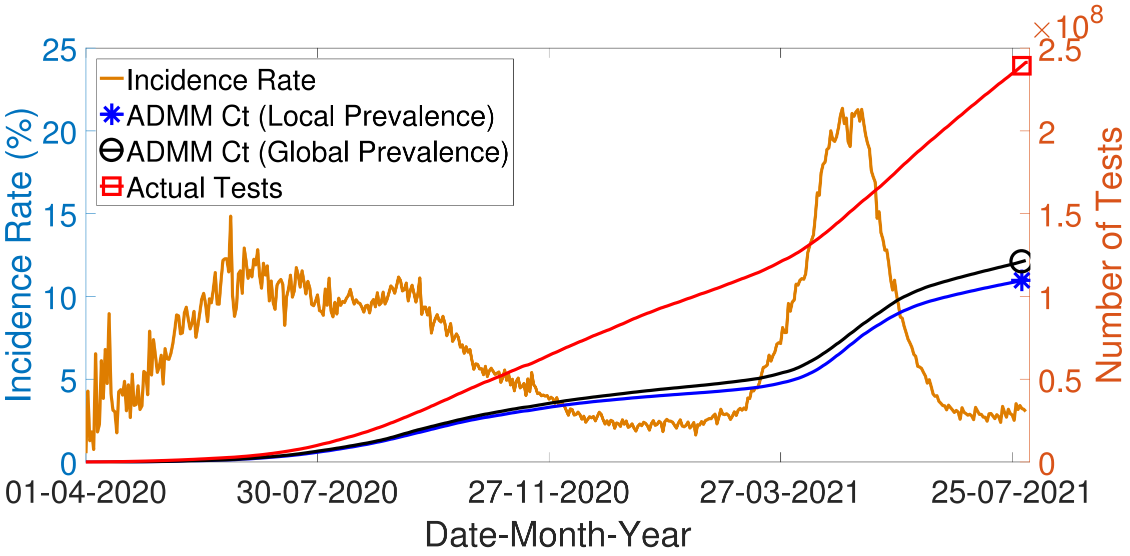


Figure 5.8: Comparison of the cumulative number of tests required by our approach using the local prevalence rates versus using the aggregated numbers across 6 states to estimate a global prevalence rate, along with actual cumulative tests conducted and the variation of aggregated positivity rate over time from April 2020 till July 2021.

using actual testing data. The total number of tests conducted across the 6 states are aggregated to obtain the total number of individual tests conducted and the total number of tests required if our algorithm is used. Also, using the raw aggregated data from these 6 states, a *global* prevalence rate is computed. Finally, the number of tests required by our method using the global prevalence rate information is also computed.

The comparison of the cumulative number of tests required under individual testing and when the local/global prevalence rates are used is shown, along with the global prevalence rate from April 2020 till July 2021 in Figure 5.8. It can be observed from Figure 5.8 that there is a clear advantage of using local prevalence rate information. Usage of the local prevalence rate to design the pool test saves $\sim$ 11.44 million tests and hence, a cost-saving of $\sim$ 160.21 million USD, compared to the global prevalence rate based design.

Next, we illustrate how choosing the test matrix based on the prevalence rate helps.

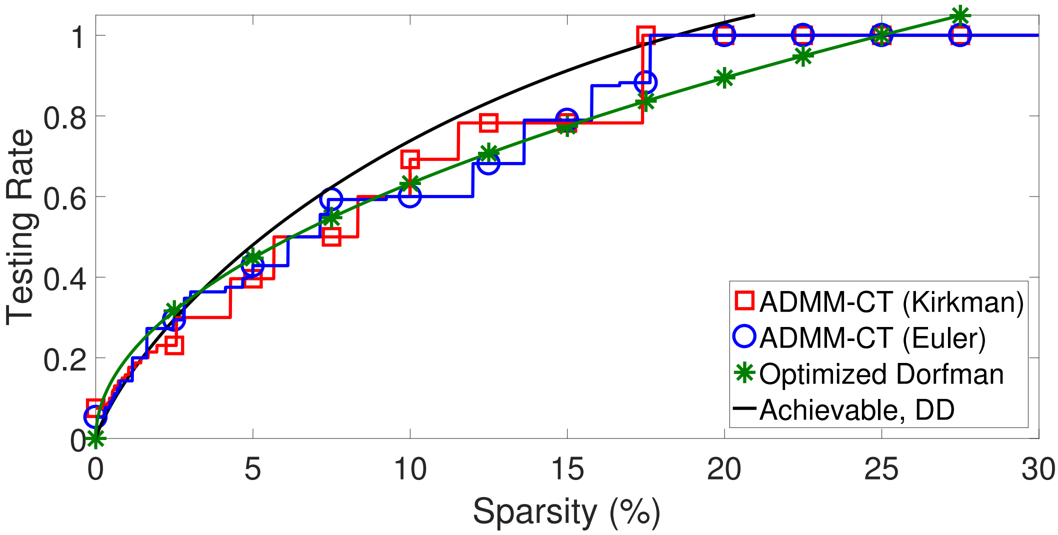


Figure 5.9: Comparison of two-stage Dorfman testing rate with single-stage rates and optimum choice of deterministic matrices: Kirkman and Euler types available across the prevalence rates, i.e., sparsity levels (%).

Figure 5.9 shows the testing rate achieved by selecting the best member of the Euler and Kirkman family of matrices at each prevalence rate. The figure also shows the testing rate achieved by the Dorfman method with the optimum pool size for each prevalence rate and the rate achieved by the DD algorithm, computed approximately from the counting bound as mentioned earlier.

From Figure 5.9, we can make several interesting observations. First, pool testing is beneficial compared to individual testing when the prevalence rate is $< 25\%$. Second, non-adaptive pool testing methods have an advantage over the adaptive Dorfman style testing when the prevalence rate is $< 13.5\%$. Third, the quantitative (or CT) measurement-based method has an advantage over DD, a binary model-based approach, when the prevalence rate $\sim 1.6 - 17.5\%$. Fourth, the adaptive Dorfman and the quantitative measurement methods provide similar testing rates when the prevalence rate $\sim 0.25 - 1.6\%$. Further from Figure 5.9, we note that the testing rates shown in the blue and red curves, corresponding to Euler and Kirkman designs, respectively, overlap at some prevalence rates. Thus, it indicates the existence of multiple non-adaptive deterministic pooling strategies.

5.6 Chapter Summary

In this chapter, we developed two iterative algorithms: 1) ADMM-CT and 2) BCD-CT for jointly estimating the PCR efficiency factor and the individual-sample CTs. Our algorithm's performance was compared across different noise levels and the PCR efficiency factors. Numerical simulations showed that the ADMM-CT algorithm performs better than the BCD-CT variants, namely BCD-CT-G and BCD-CT-I. Among the BCD-CT variants, the BCD-CT-G performs better than the BCD-CT-I algorithm.

In Algorithm 3, the BCD iterations are not tied together since each block is optimized by independently solving the sub-problems in a cyclic manner (see (5.4) and (5.5)). In contrast, the iterates are better coupled in the ADMM-CT algorithm. In Algorithm 5, 1) an auxiliary variable, $\mathbf{w}_k$, similar in behavior to the individual-sample CTs, $\mathbf{u}_k$, is used; 2) the sub-problem that solves for $\mathbf{u}_k$ uses a previous value of q_k, whereas, the sub-problem which solves for $\mathbf{w}_k$ uses the updated $q_k \leftarrow q_{k+1}$; and 3) the auxiliary variables $\mathbf{w}_k$ and $\mathbf{u}_k$ are tied together using a dual variable, $\boldsymbol{\mu}_k$, in the ADMM-CT algorithm. Due to these reasons, the ADMM-CT outperforms the BCD-CT algorithm.

The ADMM-CT algorithm was shown to be robust to uncertainty in the PCR efficiency factor, i.e., usage of an incorrect conversion factor in the model does not affect the performance of the ADMM-CT algorithm. In contrast, the performance of the CS-based methods degrades as the uncertainty in q increases, showing the necessity and effectiveness of our method. Further, the robustness of our algorithm, as the measurement-noise level is increased, is illustrated. The CS-based algorithms' performance degrades rapidly as the noise level increases. In contrast, our algorithms are not significantly affected by the noise level.

We presented empirical results related to the Covid-19 pandemic testing numbers to show when group testing is useful. In summary, the quantitative measurement-based approaches are more cost-effective than the binary outcome-based methods. We then demonstrated that using the local prevalence rates to design the tests is helpful in practice by using the actual testing numbers and positivity rates from 6 different states of India during different stages of the Covid-19 pandemic. We empirically characterized the testing rates achievable using deterministic pooling matrices for different prevalence rates. That is, we provided insight into which deterministic pooling matrix should be chosen to guarantee a near-zero error at each prevalence rate. Finally, we presented empirical results on the prevalence rates at which adaptive group testing, non-adaptive binary measurements-based group testing, and non-adaptive quantitative measurements-based group testing method gives the best testing rate.

6 | Conclusions and Future Work

Chapter Highlights

In this chapter, we summarize the thesis, including the key contributions and observations made along the way. We have addressed two aspects of group testing: 1) theoretical analysis of group testing algorithms and 2) development of novel iterative recovery algorithms with an application to Covid-19 detection. We first set the stage for the theoretical analysis by viewing the non-adaptive group testing problem as a function learning problem. In particular, this connection enabled us to apply PAC analysis to derive novel sufficiency bounds on the number of tests for well-known Boolean non-adaptive group testing algorithms. The derived bounds are a function of the number of items and defective items along with two additional parameters, namely, 1) the approximation error tolerance probability and 2) the confidence level. We showed that the derived bounds are tight and provided deeper insights into their behavior through order-wise analysis. Thus, this thesis provides a common framework for analyzing non-adaptive group testing algorithms accommodating both exact and approximate recovery *and* simultaneously accounts for the randomness in the test matrix. Next, we developed novel iterative recovery algorithms for Covid-19 detection using the pooled RT-qPCR model. Our algorithms overcome the performance degradation of the existing algorithms when the PCR model efficiency parameter is unknown. Further, we demonstrated the resource and cost savings achievable using quantitative measurements in a non-adaptive pool test setting.

This thesis has, on the whole, focused on studying theoretical aspects of group testing and applying group testing to pooled RT-qPCR-aided Covid-19 disease detection. *Through our theoretical analysis*, our primary contribution is developing a PAC-based framework for deriving sufficiency bounds on the number of tests as a function of both the confidence level and the approximation error tolerance. *On the application of group testing*, we motivated the necessity of using quantitative measurements in non-adaptive group testing for decoding the Covid-19 infected samples. We then developed two iterative algorithms for decoding the set of infected samples using the pooled RT-qPCR outcomes. The challenges that arise from the RT-qPCR model include non-linearity/multiplicative noise (which we addressed by reformulating the problem in the log domain, making the noise additive but the model non-linear), and the presence of infinities in the feasible set. Further, we extended the decoding algorithms to function even when the PCR efficiency factor is unknown. The effectiveness of the proposed algorithms is demonstrated in comparison with the Boolean group testing and CS-based algorithms.

Below, we present a chapter-wise summary of our key findings.

6.1 Summary of the Thesis

In Chapter 2, we set the stage for theoretical analysis of non-adaptive group testing algorithms using a PAC learning-based framework. In particular, the PAC learning view of the non-adaptive group testing problem was introduced in Section 2.2, where Lemma 2.1 establishes the equivalence of notions of exact recovery under the group testing and the PAC learning framework.

Next, we applied PAC analysis for deriving sufficiency bounds on the number of tests for well-known Boolean group testing algorithms, namely, COMP (under Bernoulli and near-constant row-weight test designs, denoted by COMP-B and COMP-R, respectively) and DD (under Bernoulli design), in Chapter 3. The derived PAC bounds are a function of the confidence parameter, δ, and the approximation error probability tolerance, ϵ, in addition to the number of items, n, and the number of defective items, k. In Section 3.2 and Section 3.3, we derived the PAC bounds for the COMP algorithm (see Theorem 3.1 and Theorem 3.2) and for the DD algorithm (see Theorem 3.3), respectively.

In the case of COMP-B and DD algorithms, our bound matches with the bound from the literature when we set $\epsilon = 0$. On the other hand, our COMP-R bound is tighter compared to the one in the literature at $\epsilon = 0$. The order-wise analysis shows that the bound is $\propto \log(C_d/\delta)$ and $\propto \log(1/\epsilon) + 1/\epsilon$ for large n and k, where $C_d = 1$ for COMP-B and DD and $C_d = 2$ for the COMP-R algorithm. We empirically showed that our bounds are tight, and allowing a small number of error allows one to obtain a significantly higher confidence for a given number of tests. The visualization methods, namely, *testing rate surface* and *sufficient tests contour* provide insights into the behavior of the bounds across various values of δ and ϵ. Finally, we observed that the *log-testing rate*, $\log \rho_R$, in the sub-linear regime, i.e., $k = \Theta(n^\beta)$ for $\beta \in (0, 1)$, is approximately linearly decreasing with $\log n$ with slope $\beta - 1$ for large n.

We then switched gears to discuss an application of group testing in Chapter 4 and Chapter 5 for Covid-19 detection using pooled RT-qPCR. A brief introduction to the RT-qPCR process can be found in Section 4.1.1 and the RT-qPCR system model in Section 4.2. We discussed COMP and DD in the context of Covid-19 detection in Section 4.3 and presented our GD-CT algorithm in Section 4.4.1 (see Algorithm 1) and IMHT-CT algorithm in Section 4.4.2 (see Algorithm 2). Numerical simulations in Section 4.5 showed that the GD-CT algorithm, in balance, performs the best under various settings.

In Chapter 5, we developed the BCD-CT algorithm in Section 5.3.1 and the ADMM-CT algorithm in Section 5.3.2. Numerical simulations in Section 5.4 showed that the ADMM-CT algorithm, in balance, performs the best under various settings. The numerical experiments were performed with a deterministic pooling matrix design (for motivation, see Section 4.1.2). However, the algorithms are applicable to any pooling matrix design.

In Section 5.5, we empirically showed that the non-adaptive protocol with quantitative measurements is beneficial in terms of cost savings when the prevalence rate is between $1.6-17.5\%$ and that local prevalence rate-aided test design choice offers even higher benefits. Also, we empirically characterized the testing rates achievable for different prevalence rates and group testing protocols.

In summary, the key take-home messages from this thesis are as follows:

1. The PAC formulation of the group testing problem serves as a universal framework to analyze various group testing algorithms. Unlike the classical PAC model, our formulation accommodates exact and approximate recovery settings. Further, the data distribution can be chosen based on the hypothesis space containing the target function. Our PAC analysis also enables characterizing a lower bound on the cumulative distribution of the approximation errors.

2. The PAC formulation of the group testing problem can be used to derive novel sufficiency bounds on the number of tests for non-adaptive group testing algorithms. The new sufficiency bounds account for both the randomness of the test matrix and the approximation error probability. The derived bounds are tight and agree with

the existing bounds in the exact recovery scenario.

3. Our work on Covid-19 detection using pooled RT-qPCR led to the development of novel quantitative measurements-based group testing algorithms. We empirically demonstrated the advantage of non-adaptive group testing algorithms using quantitative measurements, compared to the CS-based and binary group-testing algorithms, under different noise levels and when the PCR efficiency factor is unknown.

4. Although group testing has gained a renewed interest due to the recent Covid-19 pandemic, the theoretical progress often does not reach the intended audience, i.e., the practitioners. The visualization aids like *testing rate surface* and *sufficient tests contours* along with our case study on when group testing is beneficial, advantages of different protocols, and, in particular, considerations on how to design an optimum test protocol serves as a starting point in bridging this gap.

6.2 Future Work

As with most research, there is always room for further studies and improvement. We catalog a few promising directions.

1. Extension of the PAC framework presented in Chapter 2 to noisy models can pave the way for deriving sufficiency bounds for noisy group testing algorithms that account for both approximation error tolerance and confidence requirements.

2. A useful extension to Chapter 3 could involve analysis of group testing algorithms under different test matrices like (near-) constant column-weight and doubly-regular design, other group testing algorithms like the SSS and LiPo decoder.

3. The PAC formulation presented in this work differs from the classical PAC analysis since the data distribution can be chosen based on the hypothesis space containing the target function. This aspect could lead to novel results on PAC-learnability.

4. Continuing on the lines of the case study conducted in Chapter 5 along with the compilation of the testing rates for different testing protocols, it would be interesting to design different decision logic (e.g., list decoding-based, clustering-based, etc.) and study its effect on the testing rate in greater depth.

5. Finally, studying the sensitivity of the PAC bounds to variation in the design parameters (e.g., Bernoulli parameter, Row-weight parameter etc.) using domain adaptation techniques [201] could be an exciting direction to further harness the power of the PAC formulation of the group testing problem.

A | Appendix to Chapter 2

For the definitions of the notations used in the below proof, the readers can refer to Chapter 2.

A.1 Proof of Lemma 2.1

Proof. We first show that if $\hat{\mathcal{K}} = \mathcal{K}$, then $\hat{x}(\mathbf{a}) = x(\mathbf{a})$ with probability one. Note that there is a one-to-one mapping between any k-literal OR-ing function $\hat{x}$ and a corresponding k-sized set $\hat{\mathcal{K}}$ containing the elements participating in the OR-ing function represented by $\hat{x}$. Hence, as long as the distribution $\mathcal{D}$ is such that $\mathbb{P}_{\mathcal{D}}(a_j = 1) \in (0, 1),\ j \in [n]$ and a_js are independent,

$$\hat{\mathcal{K}} = \mathcal{K} \quad \implies \quad \mathbb{P}_{\mathbf{a}\sim\mathcal{D}}\left(\hat{x}(\mathbf{a}) = x(\mathbf{a})\right) = 1. \tag{A.1}$$

We show the converse by contrapositive. To this end, it suffices to show that if $\hat{\mathcal{K}} \neq \mathcal{K}$, then $\exists$ $\mathbf{a}$ that occurs with nonzero probability when $\mathbf{a} \sim \mathcal{D}$, such that $\hat{x}(\mathbf{a}) \neq x(\mathbf{a})$. In other words, we need to show that

$$\hat{\mathcal{K}} \neq \mathcal{K} \implies e(\hat{x}(\cdot), x(\cdot)) = \mathbb{P}_{\mathbf{a}\sim\mathcal{D}}\left(\hat{x}(\mathbf{a}) \neq x(\mathbf{a})\right) > 0. \tag{A.2}$$

Starting with $\hat{\mathcal{K}} \neq \mathcal{K}$, we argue that the above claim holds under the following covering cases:

1. When $\hat{\mathcal{K}} \setminus \mathcal{K} \neq \emptyset$, consider any $\mathbf{a} \sim \mathcal{D}$ such that
$$a_j = \begin{cases} 1, j \in \hat{\mathcal{K}} \setminus \mathcal{K} \\ 0, j \notin \hat{\mathcal{K}} \setminus \mathcal{K}. \end{cases}$$
Then, for such an $\mathbf{a}$, which occurs with nonzero probability when $\mathbf{a} \sim \mathcal{D}$, we have $\hat{x}(\mathbf{a}) = 1$ whereas $x(\mathbf{a}) = 0$.

2. When $\mathcal{K} \setminus \hat{\mathcal{K}} \neq \emptyset$, consider any $\mathbf{a} \sim \mathcal{D}$ such that
$$a_j = \begin{cases} 1, j \in \mathcal{K} \setminus \hat{\mathcal{K}} \\ 0, j \notin \mathcal{K} \setminus \hat{\mathcal{K}}. \end{cases}$$
Then, for such an $\mathbf{a}$, which occurs with nonzero probability when $\mathbf{a} \sim \mathcal{D}$, we have $\hat{x}(\mathbf{a}) = 0$ whereas $x(\mathbf{a}) = 1$.

Noting that the distribution $\mathcal{D}$ obeys $\mathbb{P}_{\mathcal{D}}(a_j = 1) \in (0,1),\ j \in [n]$ and that the a_js are independent, we get $\mathbb{P}_{\mathbf{a}\sim\mathcal{D}}(\hat{x}(\mathbf{a}) \neq x(\mathbf{a})) > 0$ as required in (A.2) in both the cases, thereby proving the converse part. $\blacksquare$

B | Appendix to Chapter 3

For the definitions of the notations used in the below proofs, the readers can refer to Chapter 2 and Chapter 3.

B.1 Proof of Lemma 3.1

Proof. Recall that the probability with which an item can participate in a test is p. A non-defective item will remain hidden in a group test under two mutually exclusive conditions: 1) the test outcome is positive or 2) the test outcome is negative but the item does not participate in the test. It then follows that the probability that g items will be hidden, denoted by $\mathbb{P}_g^h$, can be written as

$$\begin{aligned}\mathbb{P}_g^h &= \mathbb{P}[\text{positive test}] + \mathbb{P}[\text{negative test and } g \text{ items excluded}] \\ &= (1 - (1-p)^k) + (1-p)^{g+k}.\end{aligned}$$

Since the tests are independent, the probability that g items will remain hidden in m tests is given by

$$\mathbb{P}_g^h(m) = (1 - (1-p)^k + (1-p)^{g+k})^m, \tag{B.1}$$

as required. ■

B.2 Proof of Theorem 3.1

Proof. If we allow at most g_ϵ non-defective items to remain hidden, an error occurs if some subset of $g_\epsilon+1$ non-defective items remains hidden. Using the union bound

$$\mathbb{P}(e(\hat{x},x)>\epsilon)=\mathbb{P}(G>g_\epsilon)\leq\binom{n-k}{g_\epsilon+1}\mathbb{P}^h_{g_\epsilon+1}(m), \tag{B.2}$$

where the first equality holds since COMP algorithm only makes false positive errors. Using Lemma 3.1 in (B.2), we get

$$\mathbb{P}(e(\hat{x},x)>\epsilon)\leq\binom{n-k}{g_\epsilon+1}\left(1-(1-p)^k+(1-p)^{g_\epsilon+1+k}\right)^m. \tag{B.3}$$

Using $\mathbb{P}(e(\hat{x},x)>\epsilon)=\mathbb{P}(G>g_\epsilon)\leq\delta$ in (B.3) and rearranging the terms, we get the desired result. ■

B.3 Proof of Lemma 3.2

Proof. (a) Let T denote the number of draws needed to collect $w-g$ distinct coupons, and let T_i denote the number of draws needed to collect the ith coupon after $i-1$ coupons have been collected. Then, $T_1=1$ and $T=\sum_{i=1}^{w-g}T_i$. Now, the probability of collecting a new coupon in a single draw after $i-1$ coupons have been collected is $p_i=\frac{w-(i-1)}{w}$, so that T_is are independent Geometrically distributed random variables with expectation $1/p_i$. Therefore, the expected stopping time is

$$\begin{aligned}\mathbb{E}[T]&=\sum_{i=1}^{w-g}\mathbb{E}[T_i]\\&=\sum_{i=1}^{w-g}\frac{1}{p_i}\end{aligned}$$

$$
\begin{aligned}
&= \frac{w}{w} + \frac{w}{w-1} + \cdots + \frac{w}{w-(w-g-1)} \\
&= w\left[\left(\frac{1}{w} + \frac{1}{w-1} + \cdots + \frac{1}{g+1} + \frac{1}{g} + \cdots + \frac{1}{1}\right) - \left(\frac{1}{g} + \cdots + \frac{1}{1}\right)\right] \\
&= w\left[H_w - H_g\right],
\end{aligned}
$$

where H_w denotes wth Harmonic number as defined earlier. Using a well known asymptotic approximation for $H_w \approx \log w + \gamma$, where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant as mentioned in Section 3.2.2,[1] we get $\mathbb{E}[T] \approx w\left[\log w + \gamma - H_g\right]$, the required result.

(b) Let Z_i^r denote the event that the ith coupon was not picked in the first r trials (draws). Then,

$$
\begin{aligned}
\mathbb{P}(Z_i^r) &= \left(1 - \frac{1}{w}\right)^r \\
&\leq e^{-r/w}.
\end{aligned}
$$

Let Z_{i_1,i_2}^r denote the event that the i_1 and i_2th coupons, $i_1 \neq i_2$, were not picked in the first r trials. Then,

$$
\begin{aligned}
\mathbb{P}(Z_{i_1,i_2}^r) &= \left(1 - \frac{1}{w}\right)^r \cdot \left(1 - \frac{1}{w-1}\right)^r \\
&\leq e^{-r/w} \cdot e^{-r/(w-1)}.
\end{aligned}
$$

Continuing in this manner, the probability that $i_1, i_2, \ldots i_g$th coupons were not picked in the first r trials is given by

$$
\begin{aligned}
\mathbb{P}(Z_{\{i_l\}_{l=1}^g}^r) &= \prod_{i=0}^{g-1} \left(1 - \frac{1}{w-i}\right)^r \\
&\leq \prod_{i=0}^{g-1} e^{-r/(w-i)}
\end{aligned}
$$

[1] For e.g., for $w = 500$, the error in the approximation is about 0.015%.

$$= e^{-r\sum_{i=0}^{g-1} 1/(w-i)}$$

$$= e^{-r[H_w - H_{w-g}]}.$$

When we want to collect only $w-g$ coupons, the event of interest occurs when $g+1$ coupons are missed in first $r = \chi w\,[\log w + \gamma - H_g]$ trials. Note that,

$$\begin{aligned}
\mathbb{P}(Z^r_{\{i_l\}_{l=1}^{g+1}}) &\leq e^{-\chi w[\log w+\gamma-H_g]\left[H_w - H_{w-(g+1)}\right]} \\
&\leq e^{-\chi[\log w+\gamma-H_g](g+1)} \\
&= w^{-(g+1)\chi} e^{(g+1)\chi[H_g-\gamma]}, \qquad \text{(B.4)}
\end{aligned}$$

where we use the fact that $H_w - H_{w-(g+1)} = \sum_{i=0}^{g} 1/(w-i) \geq (g+1)/w$ in the penultimate step.

Since any $g+1$ coupons out of the w coupons can be missed, taking the union bound over $\binom{w}{g+1}$ sets, and using the inequalities $\binom{w}{q} \leq w^q/q!$, $q \leq \sqrt{w}$ and $q! \geq q^q/e^{q-1}$, $q \geq 1$ with $q = g+1$ to simplify the upper bound, we obtain

$$\begin{aligned}
\mathbb{P}(T > \chi\mathbb{E}[T]) &= \mathbb{P}\left(\bigcup_{i=1}^{\binom{w}{g+1}} Z^r_{\{i_l\}_{l=1}^{g+1}}\right) \\
&\leq \binom{w}{g+1}\mathbb{P}\left(Z^r_{\{i_l\}_{l=1}^{g+1}}\right) \\
&\leq \frac{w^{(g+1)}}{(g+1)!}\mathbb{P}(Z^r_{\{i_l\}_{l=1}^{g}}) \\
&\leq w^{(g+1)(-\chi+1)}\frac{e^{(g+1)\chi[H_g-\gamma]+g}}{(g+1)^{(g+1)}}, \qquad \text{(B.5)}
\end{aligned}$$

as required. ■

B.4 Proof of Theorem 3.2

Proof. In order to obtain the tail bound on m for s-length test vector designs, we start by modifying (3.11) as follows. Right-hand side is modified to $\chi(n-k)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]$. This is because all the distinct non-defectives excluding g_ϵ have been *collected* if $\chi(n-k)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]$ total non-defective items have been collected. From Lemma 3.2(b), the probability that the stopping time is more than χ times the expected stopping time is at most $(n-k)^{(g_\epsilon+1)(-\chi+1)}\frac{e^{(g_\epsilon+1)\chi[H_{g_\epsilon}-\gamma]+g_\epsilon}}{(g_\epsilon+1)^{(g_\epsilon+1)}}$.

The left-hand side of (3.11) is multiplied with $(1-\eta)$, where η is a design parameter to be specified by the Chernoff bound. Then, the probability that the actual number of items in the negative tests is smaller than $(1-\eta)$ times the expected number, $ms((n-k)/n)^s$ is at most [6]

$$\exp\left(-\eta^2 m\left(\frac{n-k}{n}\right)^s\right). \tag{B.6}$$

Taking the union bound over the above two low probability events, we have that

$$(1-\eta)ms\left(\frac{n-k}{n}\right)^s \geq \chi(n-k)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right] \tag{B.7}$$

does not hold with probability

$$\exp\left(-\eta^2 m\left(\frac{n-k}{n}\right)^s\right)+(n-k)^{(g_\epsilon+1)(-\chi+1)}\frac{e^{(g_\epsilon+1)\chi[H_{g_\epsilon}-\gamma]+g_\epsilon}}{(g_\epsilon+1)^{(g_\epsilon+1)}}. \tag{B.8}$$

From (B.7), we have

$$m \geq \frac{\chi(n-k)}{(1-\eta)s\left(\frac{n-k}{n}\right)^s}\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]. \tag{B.9}$$

Substituting (B.9) in (B.8), P_e is upper bounded by

$$\begin{aligned} P_e &\leq e^{-\eta^2 m\left(\frac{n-k}{n}\right)^s} + \frac{e^{(g_\epsilon+1)(-\chi+1)\log(n-k)+(g_\epsilon+1)\chi[H_{g_\epsilon}-\gamma]+g_\epsilon}}{e^{(g_\epsilon+1)\log(g_\epsilon+1)}} \\ &\leq \exp\left(\frac{-\eta^2\chi(n-k)}{(1-\eta)s}\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]\right) + e^{-\chi(g_\epsilon+1)[\log(n-k)+\gamma-H_{g_\epsilon}]+(g_\epsilon+1)\log\left(\frac{n-k}{g_\epsilon+1}\right)+g_\epsilon} \end{aligned} \tag{B.10}$$

Since $P_e = P(G > g_\epsilon) \leq \delta \in (0,1)$ and each term in the RHS of (B.10) is greater than 0, we choose a design parameter $c \in (0,1)$ such that

(a) $\exp\left(\frac{-\eta^2\chi(n-k)}{(1-\eta)s}\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]\right) \leq (1-c)\delta$, and

(b) $e^{-\chi(g_\epsilon+1)[\log(n-k)+\gamma-H_{g_\epsilon}]+(g_\epsilon+1)\log\left(\frac{n-k}{g_\epsilon+1}\right)+g_\epsilon} \leq c\delta$.

Simplifying (b), we get

$$\chi \geq \frac{\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1} + \frac{g_\epsilon}{g_\epsilon+1} + \log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}{\log(n-k)+\gamma-H_{g_\epsilon}}. \tag{B.11}$$

Substituting (B.11) in (a), we get

$$\begin{aligned} \frac{\eta^2}{1-\eta}\left(\frac{n-k}{s}\right)\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1} + \frac{g_\epsilon}{g_\epsilon+1} + \log\left(\frac{n-k}{g_\epsilon+1}\right)\right] &\geq \log\left(\frac{1}{(1-c)\delta}\right) \\ \frac{\eta^2}{1-\eta} \geq C \triangleq \frac{\log\left(\frac{1}{(1-c)\delta}\right)}{\left(\frac{n-k}{s}\right)\left[\frac{\log\left(\frac{1}{c\delta}\right)}{g_\epsilon+1}+\frac{g_\epsilon}{g_\epsilon+1}+\log\left(\frac{n-k}{g_\epsilon+1}\right)\right]}&. \end{aligned} \tag{B.12}$$

Noting that $C > 0$, the bound on η is obtained by solving the inequality $\eta^2 + C\eta - C \geq 0$ subject to $\eta \in (0,1)$ to get

$$\eta \geq \frac{-C+\sqrt{C^2+4C}}{2}. \tag{B.13}$$

Finally, (B.9), with (B.11) and (B.13), gives the required bound for the sufficient number of tests. ■

B.5 Proof of Corollary 3.1

Proof. We follow on similar lines as [6] and simplify (3.13) at $s = s^*$, to get

$$\begin{aligned} m &\geq \frac{\chi}{1-\eta} \frac{n-k}{\left(\frac{n-k}{n}\right)^{s^*}} \log\left(\frac{n}{n-k}\right)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right] \\ &\geq \frac{\chi}{1-\eta}\left(\frac{n}{n-k}\right)^{s^*}\left(k-\frac{k^2}{2(n-k)}\right)\left[\log(n-k)+\gamma-H_{g_\epsilon}\right] \qquad \text{(B.14)} \\ &\geq \frac{\chi k}{1-\eta}\left(\frac{n}{n-k}\right)^{s^*}\left[\log(n-k)+\gamma-H_{g_\epsilon}\right]. \qquad \text{(B.15)} \end{aligned}$$

where (B.14) is obtained by using $\log(1+x) \geq x - x^2/2$ with $x = k/(n-k)$ and (B.15) holds because increasing the RHS of (B.14) can only make the error probability in (B.8) smaller. ■

B.6 Proof of Lemma 3.3

Proof. Part (a) follows because the marginals are binomial, so it only remains to show parts (b) and (c).

(b) A non-defective item will be in the PDS if it does not participate in any of the negative tests. If there are B_- negative tests, the probability that any given item among the $n-k$ non-defective items will be in the PDS is $(1-p)^{B_-}$. Since the tests are independent, denoting the number of hidden non-defectives by the random variable G, it is easy to see that

$$G|B_- \sim \text{Bin}(n-k, (1-p)^{B_-}). \qquad \text{(B.16)}$$

Using $\mathbb{E}[G|B_- = r] = (n-k)(1-p)^r$, $0 \le r \le m$, we get

$$\begin{aligned}
\bar{g} =& \sum_{g=0}^{n-k} g\mathbb{P}(G=g) \\
=& \sum_{g=0}^{n-k} g \sum_{r=0}^{m} \mathbb{P}(B_- = r)\mathbb{P}(G=g|B_- = r) \\
=& \sum_{r=0}^{m} \mathbb{P}(B_- = r) \sum_{g=0}^{n-k} g\mathbb{P}(G=g|B_- = r) \\
=& \sum_{r=0}^{m} \mathbb{P}(B_- = r)\mathbb{E}[G|B_- = r] \\
=&(n-k) \sum_{r=0}^{m} \binom{m}{r} (q_-(1-p))^r (1-q_-)^{m-r} \\
=&(n-k)(1-p(1-p)^k)^m,
\end{aligned}$$

where the last step follows by using part (a) of this lemma and the binomial expansion for $(a+b)^m$ along with $q_- = (1-p)^k$.

(c) $\mathbb{P}(\cap_{i=1}^{d}\{L_i = 0\}|G = g)$ denotes the probability that the set output by the DD algorithm misses d of the defective items, conditioned on g hidden non-defective items being present in the PDS. In a single test, a defective item will *not* be missed, i.e., it will be identified as a definite defective, if it is the sole item among the PDS participating in that test; this occurs with probability $p(1-p)^{k-1}(1-p)^g$. Since the d items being classified as definite defectives are mutually exclusive events, the probability that none of the d defective items are correctly classified as a definite defective in a single test is $1 - dp(1-p)^{k-1}(1-p)^g$. Finally, this event should happen across all the m independently drawn tests. Thus, we have that

$$\mathbb{P}(\cap_{i=1}^{d}\{L_i = 0\}|G = g) = (1 - dp(1-p)^{k-1+g})^m,$$

which completes the proof. ■

B.7 Proof of Theorem 3.3

Proof. The error event will occur if *more than* d_ϵ defective items remain unidentified, i.e., $\cap_{i=1}^{d_\epsilon+1}\{L_i = 0\}$ occurs. Using Lemma 3.3(c) and the union bound, we get

$$\mathbb{P}(e(\hat{x}, x) > \epsilon | G = g) \leq \binom{k}{d_\epsilon + 1}(1 - (d_\epsilon + 1)p(1-p)^{k-1+g})^m, \tag{B.17}$$

and hence

$$\mathbb{P}(e(\hat{x}, x) > \epsilon) \leq \sum_{g=0}^{n-k} \binom{k}{d_\epsilon + 1}(1-(d_\epsilon + 1)p(1-p)^{k-1+g})^m \mathbb{P}(G = g). \tag{B.18}$$

Now, in order to characterize $\mathbb{P}(G = g)$, note that since the tests are drawn independently, $B_- \sim \text{Bin}(m, q_-)$, where $q_- = (1-p)^k$. Using this along with the fact that $G|B_- \sim \text{Bin}(n-k, (1-p)^{B_-})$ (see (B.16)), we get

$$\begin{aligned}
\mathbb{P}(G = g) &= \sum_{b=0}^{m} \mathbb{P}(G = g | B_- = b)\mathbb{P}(B_- = b) \\
&= \sum_{b=0}^{m} \left[\binom{n-k}{g}(1-p)^{bg}(1-(1-p)^b)^{n-k-g} \times \binom{m}{b}(1-p)^{kb}(1-(1-p)^k)^{m-b}\right] \\
&= \binom{n-k}{g} \sum_{b=1}^{m} \left[\binom{m}{b}(1-p)^{b(g+k)} \times (1-(1-p)^b)^{n-k-g}(1-(1-p)^k)^{m-b}\right].
\end{aligned} \tag{B.19}$$

Note that, in (B.19), we sum the terms from $b = 1$ since $1 - (1-p)^b = 0$ when $b = 0$. It is easy to see that $m\binom{n-k}{g}\binom{m}{m/2}(1-p)^{(g+k)}$ is a loose upper bound on $\mathbb{P}(G = g)$. Clearly, for any given m and large enough g, $\mathbb{P}(G = g)$ becomes negligible. Using this to replace g by

$\bar{g} + \tilde{g}$ for some non-negative $\tilde{g}$ (with $\bar{g}$ as given by of Lemma 3.3(b)), we get

$$\begin{aligned}
\mathbb{P}(e(\hat{x}, x) > \epsilon) &\leq \binom{k}{d_\epsilon+1} \sum_{g=0}^{n-k} (1-(d_\epsilon+1)p(1-p)^{k-1+g})^m \mathbb{P}(G = g) && \text{(B.20)} \\
&= \binom{k}{d_\epsilon+1} \Bigg\{ \sum_{g=0}^{\bar{g}+\tilde{g}} (1-(d_\epsilon+1)p(1-p)^{k-1+g})^m \mathbb{P}(G=g) \\
&\qquad + \sum_{g=\bar{g}+\tilde{g}+1}^{n-k} (1-(d_\epsilon+1)p(1-p)^{k-1+g})^m \mathbb{P}(G=g) \Bigg\} && \text{(B.21)} \\
&\leq \binom{k}{d_\epsilon+1} (1-(d_\epsilon+1)p(1-p)^{k-1+\bar{g}+\tilde{g}})^m \sum_{g=0}^{\bar{g}+\tilde{g}} \mathbb{P}(G=g) && \text{(B.22)} \\
&\leq \binom{k}{d_\epsilon+1} (1-(d_\epsilon+1)p(1-p)^{k-1+\bar{g}+\tilde{g}})^m, && \text{(B.23)}
\end{aligned}$$

where $\tilde{g} \geq 0$ is a tuning parameter chosen such that the inequality in (B.22) holds. ■

C Appendix to Chapter 4 and Chapter 5

For the definitions of the notations used in this appendix, the readers can refer to Chapter 4.

C.1 Kirkman Matrix Designs

Consider $m = 15$ schoolgirls who walk out three abreast for seven days in succession. Is it possible to arrange them daily, so that no two girls walk abreast twice? This problem is called *Kirkman's schoolgirl problem*. Reverend T. P. Kirkman posed this question in 1850 and wondered about the existence of the solution [192, Example 2.76]. The answer to this problem is known as the *Kirkman Triple System* of order m, KTS(m), or more generally as a Steiner triple system of order m, or STS(m). An STS(m) consists of $\binom{m}{2}/3$ m-length Boolean column vectors such that each member vector has exactly three 1s and the dot product of any two vectors is ≤ 1. The KTS(m) satisfies these conditions and in addition, possess *resolvability* property i.e., the member vectors can be arranged such that the sum of vectors from i to $i+m/3-1$ equals $\mathbf{1} \in \mathbb{R}^m$ for every $i \equiv 1$ modulo $m/3$. This property of KTS ensures that any l such group of vectors can be chosen from KTS to form an $m \times n$ *Kirkman matrix*, $n > m$ with $n = lm/3,\ 3 < l \leq (m-1)/2$, while keeping the number of 1s in each row fixed.

In addition, KTS and STS are solutions for other problems like social golfer problem [192]. For smaller m, the construction of Kirkman matrices can be done via greedy methods. For more information about Kirkman-based designs, see [88, 192]. A Kirkman matrix can be used for exact recovery of up to 3 positive samples [88]. From our simulations, we observe that the usage of quantitative measurements enable us to do better.

C.2 Euler Matrix Designs

The Euler matrix designs considered in this paper are based on *generalized Euler squares* (GES) [193]. In particular, we consider the following construction: let p be a prime; let $n \geq 1$ and $k \leq p^n$ denote the number of tests that an item participates in. If R is the number of items allowed in each test and d is a theoretical bound on the identifiable number of defective items, i.e., the d-disjunctness property is satisfied, then it is possible to construct a binary matrix of dimension $p^n k \times p^{n(r+1)}$, with $R = p^{nr}$ and $d = \lfloor \frac{k-1}{r} \rfloor + 1$, for $r \in \{1, 2\}$ and coherence at most r/k [193].

C.3 NP-Hardness of (4.7)

In this sub-subsection, we show the NP-hardness of (4.7). We note that solving the optimization problem with ℓ_0 norm as given in (4.7) entails enumeration of all the candidate solutions and evaluating the cost function to find the solution.

We present a formal proof of NP-hardness below. First, we transform the cost function in (4.7) as

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \; \frac{1}{2} \|\boldsymbol{\epsilon}\|_2^2 + \psi \left\|(1+q)^{-\mathbf{u}}\right\|_0 . \tag{C.1}$$

The above uses the fact that $\tau > 0$ and $\psi \triangleq \bar{\kappa}\tau$. Using (4.5), solving (C.1) is equivalent to solving

$$\min_{\mathbf{u}} \ \left\|(1+q)^{-\mathbf{u}}\right\|_0 \ \text{s.t.} \ \left\|\mathbf{c} + \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right)}{\log(1+q)}\right\|_2 \leq \nu. \tag{C.2}$$

We define $\nu' \triangleq \nu \log(1+q)$, $\bar{\mathbf{x}} \triangleq (1+q)^{-\mathbf{u}}$ and $\bar{\mathbf{y}} \triangleq (1+q)^{-\mathbf{c}}$ and rewrite (C.2) to get

$$\min_{\bar{\mathbf{x}}} \ \|\bar{\mathbf{x}}\|_0 \ \text{s.t.} \ \|-\log(\bar{\mathbf{y}}) + \log(\mathbf{A}\bar{\mathbf{x}})\|_2 \leq \nu'. \tag{C.3}$$

It suffices to show the NP-hardness of (C.3) for, say, $\nu' = 0$ and the Kirkman pooling matrix. Therefore, we get

$$\min_{\bar{\mathbf{x}}} \ \|\bar{\mathbf{x}}\|_0 \ \text{s.t.} \ \bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{x}}. \tag{C.4}$$

We are now ready to use the proof steps similar to [178, Theorem 2.17 in Section 2.3] in the noiseless case. The overall idea is to transform a known NP-hard problem (e.g., exact cover by 3-set problem) to the problem in (C.4) in polynomial time. To this end, we start by taking $\bar{\mathbf{y}} = [1, 1, \ldots, 1]^T$. Then, using the constraint in (C.4), we get $\|\mathbf{A}\bar{\mathbf{x}}\|_0 = m$. From Appendix C.1, we use $n = lm/3, \ 3 < l \leq (m-1)/2$ to note that $n \leq m(m-1)/6 < \binom{m}{3}$, for $m \geq 3$. Therefore, the pooling matrix construction can be done in polynomial time. Further, Kirkman matrices emerge from KTS which are the solutions to the Kirkman's schoolgirls problem. Hence, the column sums of Kirkman matrices are equal to 3, implying that $\|\mathbf{A}\bar{\mathbf{x}}\|_0 \leq 3\,\|\bar{\mathbf{x}}\|_0$.

In conclusion, we get $\|\bar{\mathbf{x}}\|_0 \geq m/3$. We can run the ℓ_0 normalization problem for the two cases: $\|\bar{\mathbf{x}}\|_0 = m/3$, and $\|\bar{\mathbf{x}}\|_0 > m/3$, to conclude that solving the ℓ_0 minimization problem enables one to solve the exact cover by 3-sets problem [178]. Therefore, the problem in (C.4) and hence, the original problem in (4.7) is NP-hard in general.

C.4 Convergence of Gradient Descent [185]

Given a function of $\mathbf{x} \in \mathbb{R}^n$ denoted by $f(\mathbf{x}) \in \mathbb{R}$, recall that the kth gradient descent step is given by $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)$, where η denotes the step size. Define a gradient map $g(\mathbf{x}) \triangleq \mathbf{x} - \eta \nabla f(\mathbf{x})$. A point $\mathbf{x}^*$ is a critical point of a function f if it is a fixed point of the gradient map $g(\mathbf{x}^*) = \mathbf{x}^*$, or equivalently $\nabla f(\mathbf{x}^*) = 0$. A critical point is a local minimum if there is a neighborhood U around $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in U$, and a local maximum if $f(\mathbf{x}^*) \geq f(\mathbf{x})$. Further, a critical point is a saddle point if for all neighborhoods U around $\mathbf{x}^*$, there are $\mathbf{x}, \mathbf{y} \in U$ such that $f(\mathbf{x}) \leq f(\mathbf{x}^*) \leq f(\mathbf{y})$. A critical point can be either a local minimum, a local maximum, or a saddle point.

Define a global stable set $W^s(\mathbf{x}^*)$ of a critical point $\mathbf{x}^*$ as a set of initial conditions (points) of gradient descent that converge to $\mathbf{x}^*$, i.e., $W^s(\mathbf{x}^*) = \{\mathbf{x} : \lim_k g^k(\mathbf{x}) = \mathbf{x}^*\}$, where $g^k(\mathbf{x})$ denotes a k-fold composition of the gradient map $g(\mathbf{x})$. Similarly, given a neighborhood U around $\mathbf{x}^*$, a local stable set is defined as $W^s_{\text{loc}}(\mathbf{x}^*) \triangleq \{\mathbf{x} : \lim_k g^k(\mathbf{x}) = \mathbf{x}^*, \mathbf{x} \in U\}$.

We start by noting that the gradient descent step is designed to move in the negative direction of the gradient. This eliminates the possibility of reaching a local maximum unless our initial condition, $\mathbf{x}_0$, is a local maximum. However, for an f with a countable number of local maxima, there is a zero probability of choosing such an initial condition, when it is uniformly distributed over the domain of f.

The argument that the gradient descent never converges to a saddle point, and, hence, only to a local minimum, is as follows. The local stable set $W^s_{\text{loc}}(\mathbf{x}^*)$ can be approximated by the span of the Eigenvectors corresponding to positive Eigenvalues of the Hessian $\mathbf{H}$ of f. Whenever there is a negative Eigenvalue, since $\mathbf{x}_0$ is uniformly randomly distributed in a neighborhood U around $\mathbf{x}^*$, by Taylor's theorem, we conclude that the probability of initializing in the span of the Eigenvectors corresponding to the positive Eigenvalues is zero. That is, $W^s_{\text{loc}}(\mathbf{x}^*)$ is of measure zero. Further, if there is a convergence to a critical point, for some sufficiently large k, any initial condition enters a local stable set. In other words, the global stable set is given by $\cup_{k=0}^{\infty} g^{-k}(W^s_{\text{loc}}(\mathbf{x}^*))$. Since $W^s_{\text{loc}}(\mathbf{x}^*)$ is of measure zero, the global stable set $W^s(\mathbf{x}^*)$ is also of measure zero. In conclusion, gradient descent never converges to a saddle point and instead to a local minimum, almost surely, if $0 < \eta < 1/L$, where ∇f is L-Lipschitz.

C.5 Łojasiewicz Gradient Inequality

Given a real analytical function, f, the Łojasiewicz gradient inequality gives an upper bound for the difference between $f(\mathbf{x})$ and $f(\mathbf{x}^*)$, where $\mathbf{x}$ is any point in the open neighborhood of a critical point, $\mathbf{x}^*$, of f. More formally, a critical point $\mathbf{x}^*$ of function, f satisfies the Łojasiewicz gradient inequality if there exists a neighborhood $\mathcal{V}$, $0 \leq a < 1$, and m, $\epsilon > 0$ such that $\|\nabla f(\mathbf{x})\| \geq m|f(\mathbf{x}) - f(\mathbf{x}^*)|^a$ $\forall$ $\mathbf{x}$ in $\{\mathbf{x} \in \mathcal{V} : f(\mathbf{x}^*) < f(\mathbf{x}) < f(\mathbf{x}^*) + \epsilon\}$ [185]. This gradient inequality is useful in proving the global linear convergence of gradient descent-based algorithms [185, 202].

Bibliography

[1] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Stat.*, vol. 14, no. 4, pp. 436–440, 1943. [Cited on pages: 2, 4, 12, and 83.]

[2] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, "Group testing and sparse signal recovery," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, 2008, pp. 1059–1063. [Cited on pages: 2, 29, and 31.]

[3] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, 2nd ed. World Scientific, 1999. [Cited on pages: 4, 82, and 83.]

[4] M. Aldridge, L. Baldassini, and O. Johnson, "Group Testing Algorithms: Bounds and Simulations," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3671–3687, 2014. [Cited on pages: xvii, 4, 15, 24, 30, 35, 38, 39, 42, 43, 45, 53, 58, 59, 60, 61, 63, 64, 65, 77, 83, and 120.]

[5] 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 *Proc. Allerton Conf. on Commun., Control and Comput.*, 2011, pp. 1832–1839. [Cited on pages: 4, 24, 34, 43, 53, 66, 83, 92, and 120.]

[6] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing:

Explicit bounds and novel algorithms," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 3019–3035, 2014. [Cited on pages: xvii, 4, 15, 24, 30, 35, 38, 41, 42, 43, 44, 45, 48, 49, 50, 53, 54, 64, 65, 151, and 153.]

[7] M. Aldridge, O. Johnson, and J. Scarlett, *Group Testing: An Information Theory Perspective*. Now Foundations and Trends, 2019. [Cited on pages: 4, 5, 15, 23, 25, 26, 27, 29, 30, 39, 43, 44, 47, 77, 92, and 131.]

[8] T. Berger, N. Mehravari, D. Towsley, and J. Wolf, "Random multiple-access communication and group testing," *IEEE Trans. Commun.*, vol. 32, no. 7, pp. 769–779, 1984. [Cited on page: 7.]

[9] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 185–191, 1985. [Cited on page: 7.]

[10] N. Seshadri and P. Srikantakumar, "On group testing protocols for binary-feedback multiaccess channel," *IEEE Trans. Commun.*, vol. 33, no. 6, pp. 574–577, 1985. [Cited on page: 7.]

[11] N. Garg and S. Mohan, "Group testing protocol with capture for random access communication," *IEEE Trans. Commun.*, vol. 35, no. 8, pp. 849–854, 1987. [Cited on page: 7.]

[12] D. Kurtz and M. Sidi, "Multiple access algorithms via group testing for heterogeneous population of users," *IEEE Trans. Commun.*, vol. 36, no. 12, pp. 1316–1323, 1988. [Cited on page: 7.]

[13] F. Hwang and Y. Yao, "A generalization of the monotonicity theorem in group

testing with applications to random multiaccess channels," *IEEE Transactions on Information Theory*, vol. 35, no. 3, pp. 698–700, 1989. [Cited on page: 7.]

[14] H. A. Inan, S. Ahn, P. Kairouz, and A. Ozgur, "A group testing approach to random access for short-packet communication," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 96–100. [Cited on page: 7.]

[15] J. Robin and E. Erkip, "Access delay constrained activity detection in massive random access," in *Proc. SPAWC*, 2021, pp. 191–195. [Cited on page: 8.]

[16] ——, "Active user identification in fast fading massive random access channels," in *Proc. IEEE Inf. Theory Workshop*, 04 2023, pp. 232–237. [Cited on page: 8.]

[17] H. A. Inan, P. Kairouz, and A. Ozgur, "Energy-limited massive random access via noisy group testing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 1101–1105. [Cited on page: 8.]

[18] Y. Zheng, N. Pitsianis, and D. Brady, "Nonadaptive group testing based fiber sensor deployment for multiperson tracking," *IEEE Sensors J.*, vol. 6, no. 2, pp. 490–494, 2006. [Cited on page: 8.]

[19] A. G. Amat, G. Liva, E. Paolini, and C. Stefanovic, "Coded slotted Aloha with stopping set resolution: A group testing approach," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, 2022, pp. 677–681. [Cited on page: 8.]

[20] G. Vershinin, A. Cohen, and O. Gurewitz, "Order-optimal multiple-access channel for massive MIMO via group testing decoding," *IEEE Trans. Commun.*, vol. 72, no. 7, pp. 3890–3904, 2024. [Cited on page: 8.]

[21] A. Sharma and C. R. Murthy, "Group testing-based spectrum hole search for cognitive radios," *IEEE Trans. Veh. Tech.*, vol. 63, no. 8, pp. 3794–3805, 2014. [Cited on pages: 8 and 23.]

[22] P. Saengudomlert and S. Buddhacharya, "Spectrum sensing through group testing for multicarrier transmissions based on wavelet packet modulation," in *Int. Conf. on Electr. Eng./Electron. Comput. Telecommun. and Inf. Technol. (ECTI-CON)*, 2019, pp. 21–24. [Cited on page: 9.]

[23] S. Zheng and X. Yang, "Wideband spectrum sensing based on group testing utilizing polyphase filter banks," in *Int. Conf. on Digit. Signal Process.*, 2014, pp. 592–596. [Cited on page: 9.]

[24] Y. Xuan, Y. Shen, N. P. Nguyen, and M. T. Thai, "A trigger identification service for defending reactive jammers in WSN," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 793–806, 2012. [Cited on page: 9.]

[25] T. Tošić and P. Frossard, "Distributed group testing detection in sensor networks," in *Proc. ICASSP*, 2012, pp. 3097–3100. [Cited on page: 9.]

[26] C. Lo, M. Liu, J. P. Lynch, and A. C. Gilbert, "Efficient sensor fault detection using combinatorial group testing," in *2013 IEEE Int. Conf. on Distrib. Comput. in Sensor Syst.*, 2013, pp. 199–206. [Cited on page: 9.]

[27] Y.-W. Hong and A. Scaglione, "Group testing for sensor networks: The value of asking the right questions," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, vol. 2, 2004, pp. 1297–1301 Vol.2. [Cited on page: 9.]

[28] ——, "Generalized group testing for retrieving distributed information," in *Proc. ICASSP*, vol. 3, 2005, pp. 681–684. [Cited on page: 9.]

[29] Y.-W. P. Hong and A. Scaglione, "Group testing for binary Markov sources: Data-driven group queries for cooperative sensor networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3538–3551, 2008. [Cited on page: 9.]

[30] H. Wang, X. Liu, Y. Xie, and H. Zeng, "The scalable group testing of invalid signatures based on Latin square in wireless sensors networks," in *Int. Conf. on Intell. Comput. and Signal Process. (ICSP)*, 2021, pp. 1153–1158. [Cited on page: 9.]

[31] M. Sobel and P. A. Groll, "Group testing to eliminate efficiently all defectives in a binomial sample," *Bell System Technical Journal*, vol. 38, no. 5, pp. 1179–1252, 1959. [Cited on pages: 10, 14, and 27.]

[32] C. Chen and F. Hwang, "Detecting and locating electrical shorts using group testing," *IEEE Trans. Circuits Syst.*, vol. 36, no. 8, pp. 1113–1116, 1989. [Cited on page: 10.]

[33] A. Kahng and S. Reda, "Combinatorial group testing methods for the BIST diagnosis problem," in *2004 Asia and South Pacific Design Automat. Conf. (ASP-DAC)*, 2004, pp. 113–116. [Cited on page: 10.]

[34] R. S. Oreifej, C. A. Sharma, and R. F. DeMara, "Expediting GA-based evolution using group testing techniques for reconfigurable hardware," in *IEEE Int. Conf. on Reconfig. Comput. and FPGA's (ReConFig)*, 2006, pp. 1–8. [Cited on page: 10.]

[35] T. Madej, "An application of group testing to the file comparison problem," in *Proc. Int. Conf. on Distrib. Comput. Syst.*, 1989, pp. 237–243. [Cited on page: 10.]

[36] T. Kitagawa, M. Hagiwara, K. Nuida, H. Watanabe, and H. Imai, "A group testing based deterministic tracing algorithm for a short random fingerprint code," in *Proc. Int. Symp. on Inf. Theory and its Appl. (ISITA)*, 2008, pp. 1–5. [Cited on page: 10.]

[37] T. Laarhoven, "Efficient probabilistic group testing based on traitor tracing," in *Proc. Allerton Conf. on Commun., Control and Comput.*, 2013, pp. 1458–1465. [Cited on page: 10.]

[38] P. Meerwald and T. Furon, "Group testing meets traitor tracing," in *Proc. ICASSP*, 2011, pp. 4204–4207. [Cited on page: 10.]

[39] B. Škorić, "Tally-based simple decoders for traitor tracing and group testing," *IEEE Trans. Inf. Forensic. and Secur.*, vol. 10, no. 6, pp. 1221–1233, 2015. [Cited on page: 10.]

[40] W. Tsai, Y. Chen, R. Paul, N. Liao, and H. Huang, "Cooperative and group testing in verification of dynamic composite web services," in *IEEE Annu. Comput. Softw. and Appl. Conf. (COMPSAC)*, vol. 2, 2004, pp. 170–173. [Cited on page: 10.]

[41] X. Bai and R. S. Kenett, "Risk-based adaptive group testing of semantic web services," in *IEEE Int. Comput. Softw. and Appl. Conf.*, vol. 2, 2009, pp. 485–490. [Cited on page: 10.]

[42] S. Mishra and C. Kumar, "Improving the trust of end users in enterprise SOA using combinatorial group testing methods," in *12th Int. Conf. on Digit. Inf. Manag. (ICDIM)*, 2017, pp. 144–149. [Cited on page: 10.]

[43] S. Hirose and J. Shikata, “Aggregate message authentication code capable of non-adaptive group-testing,” *IEEE Access*, vol. 8, pp. 216 116–216 126, 2020. [Cited on page: 11.]

[44] ——, “Group-testing aggregate entity authentication,” in *Proc. IEEE Inf. Theory Workshop*, 2023, pp. 227–231. [Cited on page: 11.]

[45] Y. Xuan, I. Shin, M. T. Thai, and T. Znati, “Detecting application denial-of-service attacks: A group-testing-based approach,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 8, pp. 1203–1216, 2010. [Cited on page: 11.]

[46] V. K. Gurani, P. Shynu, and C. L. Chowdhary, “Monitoring application for DoS attacks using group-testing,” in *Int. Conf. on Electron. and Commun. Syst. (ICECS)*, 2014, pp. 1–4. [Cited on page: 11.]

[47] M. T. Thai, Y. Xuan, I. Shin, and T. Znati, “On detection of malicious users using group testing techniques,” in *Proc. Int. Conf. on Distrib. Comput. Syst.*, 2008, pp. 206–213. [Cited on page: 11.]

[48] T. V. Bui, C. N. Huynh, and T. D. Nguyen, “Early detection for networking anomalies using non-adaptive group testing,” in *2013 Int. Conf. on ICT Converg. (ICTC)*, 2013, pp. 984–987. [Cited on page: 11.]

[49] H. N. Chinh, N. D. Thuc, and T. Hanh, “Early detection and limitation hot-IPs using non-adaptive group testing and dynamic firewall rules,” in *2014 Int. Conf. on Comput., Manag. and Telecommun. (ComManTel)*, 2014, pp. 286–290. [Cited on page: 11.]

[50] C. Wang, Q. Zhao, and C.-N. Chuah, "Group testing under sum observations for heavy hitter detection," in *Proc. ITA*, 2015, pp. 149–153. [Cited on page: 11.]

[51] S. Hong, H. Yang, and J. Lee, "Hierarchical group testing for byzantine attack identification in distributed matrix multiplication," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 1013–1029, 2022. [Cited on page: 11.]

[52] F. Xu, S.-I. Azuma, R. Ariizumi, and T. Asai, "Minimum number of measurements in enhanced group testing for failure-edge detection in networks," in *Asian Control Conf. (ASCC)*, 2022, pp. 1822–1827. [Cited on page: 11.]

[53] ——, "Performance limitation of group testing in network failure detection," *IEEE Access*, vol. 11, pp. 102 852–102 859, 2023. [Cited on page: 11.]

[54] N. J. A. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. W. S. Chan, "Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs," in *IEEE Int. Conf. on Comput. Commun. (INFOCOM)*, 2007, pp. 697–705. [Cited on page: 11.]

[55] M. Kundi, B. Ahmad, M. Inaythulaha, and J. A. Nasir, "Optimization of testing time in RUP: A group testing approach," in *Int. Conf. on Comput. and Inf. Sci. (ICCIS)*, vol. 2, 2012, pp. 848–851. [Cited on page: 11.]

[56] S. Bayram, H. T. Sencar, and N. Memon, "Sensor fingerprint identification through composite fingerprints and group testing," *IEEE Trans. Inf. Forensic. and Secur.*, vol. 10, no. 3, pp. 597–612, 2015. [Cited on page: 11.]

[57] M. Hahn-Klimroth, D. Kaaser, and M. Rau, "Distributed pooled data intrusion

detection: Lessons learned from quantitative group testing," in *Proc. Int. Conf. on Distrib. Comput. Syst.* Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2024, pp. 198–208. [Cited on page: 12.]

[58] A. R. Bharti, S. L. Letendre, K. P. Patra, and J. M. Vinetz, "Malaria diagnosis by a polymerase chain reaction-based assay using a pooling strategy," *Americal Journal of Tropical Medicine and Hygiene*, vol. 81, no. 5, pp. 754–757, Nov. 2009. [Cited on pages: 12, 14, and 83.]

[59] I. Yelin, N. Aharony, E. S. Tamar, A. Argoetti, E. Messer, D. Berenbaum, E. Shafran, A. Kuzli, N. Gandali, O. Shkedi, T. Hashimshony, Y. Mandel-Gutfreund, M. Halberthal, Y. Geffen, M. Szwarcwort-Cohen, and R. Kishony, "Evaluation of COVID-19 RT-qPCR test in multi sample pools." *Clin. Infect. Dis.*, vol. 71, no. 16, pp. 2073–2078, Nov. 2020, first published in PrePrint on 27-March, 2020. [Cited on pages: 12 and 84.]

[60] J. N. Eberhardt, N. P. Breuckmann, and C. S. Eberhardt, "Multi-stage group testing improves efficiency of large-scale COVID-19 screening," *J. Clin. Virol.*, vol. 128, pp. 3775–3797, Jul. 2020. [Cited on pages: 12 and 83.]

[61] B. Abdalhamid, C. R. Bilder, E. L. McCutchen, S. H. Hinrichs, S. A. Koepsell, and P. C. Iwen, "Assessment of specimen pooling to conserve SARS CoV-2 testing resources," *Am. J. Clin. Pathol.*, vol. 153, no. 6, pp. 715–718, Apr. 2020. [Cited on pages: 12 and 84.]

[62] A. Cohen, N. Shlezinger, A. Solomon, Y. C. Eldar, and M. Médard, "Multi-level group testing with application to one-shot pooled COVID-19 tests," in *Proc. ICASSP*. IEEE, 2021, pp. 1030–1034. [Cited on pages: 12 and 84.]

[63] C. Schumacher and M. Täufer, "The statistics of noisy one-stage group testing in outbreaks," 2020. [Online]. Available: https://arxiv.org/abs/2012.02101 [Cited on pages: 12 and 84.]

[64] J. Yi, R. Mudumbai, and W. Xu, "Low-cost and high-throughput testing of COVID-19 viruses and antibodies via compressed sensing: System concepts and computational experiments," 2020. [Online]. Available: https://arxiv.org/abs/2004.05759 [Cited on pages: 12 and 84.]

[65] K. A. Waldstein, J. Yi, M. M. Cho, R. Mudumbai, X. Wu, S. M. Varga, and W. Xu, "Use of compressed sensing to expedite high-throughput diagnostic testing for COVID-19 and beyond," *medRxiv*, 2021. [Cited on pages: 12 and 84.]

[66] J. Yi, M. Cho, X. Wu, W. Xu, and R. Mudumbai, "Error correction codes for COVID-19 virus and antibody testing: Using pooled testing to increase test reliability," 2020. [Online]. Available: https://arxiv.org/abs/2007.14919 [Cited on pages: 12 and 84.]

[67] D. Chen, K. Xing, D. Henson, and L. Sheng, "Group testing in the development of an expanded cancer staging system," in *2008 7th Int. Conf. on Mach. Learn. and Appl.*, 2008, pp. 589–594. [Cited on page: 12.]

[68] F. K. Hwang and D.-Z. Du, *Pooling designs and nonadaptive group testing: Important tools for DNA sequencing*. World Sci., 2006, vol. 18. [Cited on page: 13.]

[69] H. Q. Ngo and D.-Z. Du, “A survey on combinatorial group testing algorithms with applications to DNA library screening,” in *Discrete Mathematical Problems with Medical Applications*. Amer. Math. Soc. 2000, 1999. [Cited on page: 13.]

[70] R. Kainkaryam and P. Woolf, “poolHiTS: A shifted transversal design based pooling strategy for high-throughput drug screening,” *BMC Bioinfo.*, vol. 9, p. 256, 02 2008. [Cited on page: 13.]

[71] X. Xin, J.-F. Rual, T. Hirozane-Kishikawa, D. E. Hill, M. Vidal, C. Boone, and N. Thierry-Mieg, “Shifted transversal design smart-pooling for high coverage interactome mapping,” *Genome Res.*, vol. 19, no. 7, pp. 1262–1269, 2009. [Cited on page: 13.]

[72] Y. Li, H. Zhao, K. Wilkins, C. Hughes, and I. K. Damon, “Real-time PCR assays for the specific detection of monkeypox virus West African and Congo basin strain DNA.” *J. Virol. Methods*, vol. 99, no. 1, pp. 223–237, Oct. 2010. [Cited on page: 13.]

[73] S. N. Seifert, S. Bai, S. Fawcett, E. B. Norton, K. J. Zwezdaryk, J. Robinson, B. Gunn, and M. Letko, “An ACE2-dependent Sarbecovirus in Russian bats is resistant to SARS-CoV-2 vaccines,” *PLoS pathogens*, vol. 18, no. 9, pp. 1553–7374, Sep. 2022. [Cited on page: 13.]

[74] Y. Kim and A. Tewfik, “Low power detection on capacitive touch screens,” in *IEEE Global Conf. on Signal and Inf. Proc. (GlobalSIP)*, 12 2013, pp. 638–641. [Cited on page: 13.]

[75] T. Jiang, C. Sun, S. El Rouayheb, and D. Pompili, “FaceGroup: Continual face authentication via partially homomorphic encryption and group testing,” in *IEEE*

Int. Conf. on Mobile Ad Hoc and Smart Syst. (MASS), 2023, pp. 443–451. [Cited on page: 13.]

[76] E. Hong and R. Ladner, “Group testing for image compression,” in *Proc. Data Compression Conf. (DCC)*, 2000, pp. 3–12. [Cited on page: 13.]

[77] E. S. Hong, R. E. Ladner, and E. A. Riskin, “Group testing for block transform image compression,” in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, vol. 1, 2001, pp. 769–772. [Cited on page: 13.]

[78] G. Shavit, M. Ringenburg, J. West, R. Ladner, and E. Riskin, “Group testing for video compression,” in *Proc. Data Compression Conf. (DCC)*, 2004, pp. 212–221. [Cited on page: 13.]

[79] A. Agarwal, O. Milenkovic, S. Pattabiraman, and J. a. Ribeiro, “Group testing with runlength constraints for topological molecular storage,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 132–137. [Cited on page: 13.]

[80] S. Tabatabaei, B. Wang, N. Athreya, B. Enghiad, A. Hernandez, C. Fields, J.-P. Leburton, D. Soloveichik, H. Zhao, and O. Milenkovic, “DNA punch cards for storing data on native DNA sequences via enzymatic nicking,” *Nature Communications*, vol. 11, 04 2020. [Cited on page: 13.]

[81] A. Schliep, D. Torney, and S. Rahmann, “Group testing with DNA chips: Generating designs and decoding experiments,” in *Proc. IEEE Bioinformatics Conf. (Computational System Bioinformatics)*, 2003, pp. 84–91. [Cited on page: 13.]

[82] J. Fang, Z. L. Jiang, S. M. Yiu, and L. C. K. Hui, “Hard disk integrity check by

hashing with combinatorial group testing," in *2nd Int. Conf. on Comput. Sci. and its Appl.*, 2009, pp. 1–6. [Cited on page: 13.]

[83] J. Wang, E. Lo, and M. L. Yiu, "Identifying the most connected vertices in hidden bipartite graphs using group testing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2245–2256, 2013. [Cited on page: 13.]

[84] O. Yildiz, A. Khalili, and E. Erkip, "Hybrid beam alignment for multi-path channels: A group testing viewpoint," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, 2022, pp. 6–10. [Cited on page: 13.]

[85] S. Ubaru and A. Mazumdar, "Multilabel classification with group testing and codes," in *Proc. Int. Conf. on Mach. Learn.*, ser. PMLR, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3492–3501. [Cited on pages: 14 and 31.]

[86] W. Liang and J. Zou, "Neural group testing to accelerate deep learning," in *Proc. IEEE Int. Symp. Inf. Theory*, 07 2021, pp. 958–963. [Cited on page: 14.]

[87] C. S. Kaushik Valmeekam, K. R. Narayanan, and A. Sprintson, "Over-the-air neural group testing," in *IEEE Int. Conf. on Mach. Learn. for Commun. and Netw. (ICMLCN)*, 2024, pp. 360–366. [Cited on page: 14.]

[88] S. Ghosh, R. Agarwal, M. A. Rehan, S. Pathak, P. Agarwal, Y. Gupta, S. Consul, N. Gupta, R. Ritika, R. Goenka, A. Rajwade, and M. Gopalkrishnan, "A compressed sensing approach to pooled RT-PCR testing for COVID-19 detection," *IEEE Open J. Signal Process.*, 2021. [Cited on pages: 14, 15, 83, 90, 108, 109, 120, 126, 127, 128, 131, and 158.]

[89] Y. Kawaguchi, T. Osa, S. Barnwal, H. Nagano, and M. Togami, "Information-based pool size control of Boolean compressive sensing for adaptive group testing," in *Euro. Sig. Process. Conf. (EUSIPCO)*, 2014, pp. 2280–2284. [Cited on page: 15.]

[90] J. Scarlett, "Noisy adaptive group testing: Bounds and algorithms," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3646–3661, 2019. [Cited on pages: 15 and 25.]

[91] J. Scarlett and V. Cevher, "Converse bounds for noisy group testing with arbitrary measurement matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 2868–2872. [Cited on page: 15.]

[92] ——, "How little does non-exact recovery help in group testing?" in *Proc. ICASSP*, 2017, pp. 6090–6094. [Cited on pages: 15, 16, and 27.]

[93] ——, "Limits on support recovery with probabilistic models: An information-theoretic framework," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 593–620, 2017. [Cited on pages: 15, 26, 27, and 47.]

[94] O. Johnson, M. Aldridge, and J. Scarlett, "Performance of group testing algorithms with near-constant tests per item," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 707–723, 2019. [Cited on pages: 15, 25, and 26.]

[95] M. B. Malyutov, "The separating property of random matrices," *Mat. Zametki*, vol. 23, no. 1, pp. 155–167, 1978. [Cited on pages: 24 and 25.]

[96] A. G. D'yachkov, "Lectures on designing screening experiments," Ph.D. dissertation, Combinatorial Comput. Math. Center, Pohang Univ. Sci. Technol., Gyeongsangbuk-do, South Korea, Feb. 2004. [Cited on page: 24.]

[97] M. Aldridge, "The capacity of Bernoulli nonadaptive group testing," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7142–7148, 2017. [Cited on page: 24.]

[98] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Optimal group testing," in *Proc. 29th Ann. Conf. on Learn. Theory*, ser. PMLR, vol. 125. PMLR, 09–12 Jul 2020, pp. 1374–1388. [Cited on page: 24.]

[99] ——, "Information-theoretic and algorithmic thresholds for group testing," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7911–7928, 2020. [Cited on page: 24.]

[100] C. Shangguan and G. Ge, "New bounds on the number of tests for disjunct matrices," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7518–7521, 2016. [Cited on page: 25.]

[101] M. Aldridge, O. Johnson, and J. Scarlett, "Improved group testing rates with constant column weight designs," in *Proc. IEEE Int. Symp. Inf. Theory*. IEEE Press, 2016, p. 1381–1385. [Cited on page: 25.]

[102] N. Tan and J. Scarlett, "An analysis of the DD algorithm for group testing with size-constrained tests," in *Proc. IEEE Int. Symp. Inf. Theory*, 2021, pp. 1967–1972. [Cited on page: 25.]

[103] V. Gandikota, E. Grigorescu, S. Jaggi, and S. Zhou, "Nearly optimal sparse group testing," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2760–2773, 2019. [Cited on page: 25.]

[104] O. Gebhard, M. Hahn-Klimroth, O. Parczyk, M. Penschuck, M. Rolvien, J. Scarlett, and N. Tan, "Near-optimal sparsity-constrained group testing: Improved bounds and

algorithms," *IEEE Trans. Inf. Theory*, vol. 68, no. 5, pp. 3253–3280, 2022. [Cited on page: 25.]

[105] M. Malyutov, *Search for Sparse Active Inputs: A Review.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 609–647. [Cited on pages: 25 and 29.]

[106] A. Agarwal, S. Jaggi, and A. Mazumdar, "Novel impossibility results for group-testing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 2579–2583. [Cited on page: 25.]

[107] M. Aldridge, "Rates of adaptive group testing in the linear regime," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 236–240. [Cited on page: 25.]

[108] ——, "Individual testing is optimal for nonadaptive group testing in the linear regime," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2058–2061, 2019. [Cited on page: 25.]

[109] A. Ganesan, S. Jaggi, and V. Saligrama, "Non-adaptive group testing with inhibitors," in *Proc. IEEE Inf. Theory Workshop*, 2015, pp. 1–5. [Cited on page: 25.]

[110] S. Jain, M. Cardone, and S. Mohajer, "Improved bounds for efficiently decodable probabilistic group testing with unreliable items," in *Proc. IEEE Inf. Theory Workshop*, 2023, pp. 359–364. [Cited on page: 25.]

[111] M. Fan, B.-J. Yoon, F. J. Alexander, E. R. Dougherty, and X. Qian, "Adaptive group testing with mismatched models," in *Proc. ICASSP*, 2022, pp. 4533–4537. [Cited on pages: 25 and 26.]

[112] S. Sihag, A. Tajer, and U. Mitra, “Adaptive graph-constrained group testing,” *IEEE Trans. Signal Process.*, vol. 70, pp. 381–396, 2022. [Cited on pages: 25 and 26.]

[113] C. Yao, P. Nikolopoulos, and C. Fragouli, “A diagonal splitting algorithm for adaptive group testing,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2023, pp. 1842–1847. [Cited on page: 25.]

[114] S. Ahn, W.-N. Chen, and A. Özgür, “Adaptive group testing on networks with community structure: The stochastic block model,” *IEEE Trans. Inf. Theory*, vol. 69, no. 7, pp. 4758–4776, 2023. [Cited on pages: 25 and 26.]

[115] S. D. Fiore, M. Dalai, and U. Vaccaro, “Achievable rates and algorithms for group testing with runlength constraints,” in *Proc. IEEE Inf. Theory Workshop*, 2022, pp. 576–581. [Cited on page: 26.]

[116] H. Nikpey, J. Kim, X. Chen, S. Sarkar, and S. S. Bidokhti, “Group testing with correlation via edge-faulty graphs,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 880–885. [Cited on page: 26.]

[117] B. A. Berendsohn and L. Kozma, “Group testing with geometric ranges,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 868–873. [Cited on page: 26.]

[118] S.-J. Cao, R. Goenka, C.-W. Wong, A. Rajwade, and D. Baron, “Group testing with side information via generalized approximate message passing,” *IEEE Trans. Signal Process.*, vol. 71, pp. 2366–2375, 2023. [Cited on pages: 26 and 86.]

[119] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. N. Diggavi,

"Community-aware group testing," *IEEE Trans. Inf. Theory*, vol. 69, no. 7, pp. 4361–4383, 2023. [Cited on pages: 26 and 86.]

[120] O. Gebhard, O. Johnson, P. Loick, and M. Rolvien, "Improved bounds for noisy group testing with constant tests per item," *IEEE Trans. Inf. Theory*, vol. 68, no. 4, pp. 2604–2621, 2022. [Cited on page: 26.]

[121] E. F. Ben-Knaan, Y. C. Eldar, and N. Shlezinger, "Recovery of noisy pooled tests via learned factor graphs with application to COVID-19 testing," in *Proc. ICASSP*, 2022, pp. 4518–4522. [Cited on pages: 26 and 86.]

[122] O. Johnson, "Strong converses for group testing from finite blocklength results," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5923–5933, 2017. [Cited on page: 26.]

[123] L. Flodin and A. Mazumdar, "Probabilistic group testing with a linear number of tests," in *Proc. IEEE Int. Symp. Inf. Theory*. IEEE Press, 2021, p. 1248–1253. [Cited on page: 26.]

[124] G. Atia and V. Saligrama, "Noisy group testing: An information theoretic perspective," in *Proc. Allerton Conf. on Commun., Control and Comput.*, 2009, pp. 355–362. [Cited on page: 27.]

[125] J. Scarlett and V. Cevher, *Phase Transitions in Group Testing*. ACM, 2016, pp. 40–53. [Cited on page: 27.]

[126] L. V. Truong, M. Aldridge, and J. Scarlett, "On the all-or-nothing behavior of Bernoulli group testing," *IEEE J. Sel. Inf. Theory*, vol. 1, no. 3, pp. 669–680, 2020. [Cited on page: 28.]

[127] J. Niles-Weed and I. Zadik, “It was “all” for “nothing”: Sharp phase transitions for noiseless discrete channels,” *IEEE Trans. Inf. Theory*, vol. 69, no. 8, pp. 5188–5202, 2023. [Cited on page: 28.]

[128] K. Lee, K. Chandrasekher, R. Pedarsani, and K. Ramchandran, “SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes,” *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4649–4664, 2019. [Cited on page: 28.]

[129] J. Scarlett and V. Cevher, “Near-optimal noisy group testing via separate decoding of items,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 902–915, 2018. [Cited on page: 28.]

[130] N. Tan, W. Tan, and J. Scarlett, “Performance bounds for group testing with doubly-regular designs,” *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 1224–1243, 2023. [Cited on page: 28.]

[131] F. Iliopoulos and I. Zadik, “Group testing and local search: Is there a computational-statistical gap?” in *Proc. 29th Ann. Conf. on Learn. Theory*, ser. PMLR, M. Belkin and S. Kpotufe, Eds., vol. 134. PMLR, 15–19 Aug 2021, pp. 2499–2551. [Cited on pages: 28 and 29.]

[132] M. Aldridge, “Pooled testing to isolate infected individuals,” in *Proc. Conf. on Inform. Sci. and Syst. (CISS)*, 2021, pp. 1–5. [Cited on page: 29.]

[133] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning.* The MIT Press, 2012. [Cited on pages: 29, 30, 31, 32, and 35.]

[134] S. Shalev-Shwartz and S. Ben-David, *PAC-Bayes*. Cambridge University Press, 2014, pp. 364–368. [Cited on pages: 29 and 30.]

[135] A. Pant, T. K. Maiti, D. Mahajan, and B. Das, "Human gut microbiota and drug metabolism," *Microb. Ecol.*, vol. 86, no. 1, pp. 97–111, 2022. [Cited on page: 31.]

[136] D. Malioutov and K. Varshney, "Exact rule learning via Boolean compressed sensing," in *Proc. Int. Conf. on Mach. Learn.*, ser. PMLR, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 765–773. [Cited on page: 31.]

[137] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134—1142, nov 1984. [Cited on pages: 33 and 35.]

[138] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," in *Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 1987, pp. 68–77. [Cited on page: 34.]

[139] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. USA: Cambridge University Press, 2005. [Cited on pages: 41 and 50.]

[140] N. Zoroa, E. Lesigne, M. J. Fernández-Sáez, P. Zoroa, and J. Casas, "The coupon collector urn model with unequal probabilities in ecology and evolution," *Journal of The Royal Society Interface*, vol. 14, no. 127, p. 20160643, 2017. [Cited on page: 41.]

[141] A. Boneh and M. Hofri, "The coupon-collector problem revisited — a survey of

engineering problems and computational methods," *Stochastic Models*, vol. 13, pp. 39–66, 1997. [Cited on page: 41.]

[142] F. Comets, F. Delarue, and R. Schott, "Information transmission under random emission constraints," *Combinatorics, Probability and Computing*, vol. 23, no. 6, p. 973–1009, 2014. [Cited on page: 41.]

[143] E. Anceaume, Y. Busnel, E. Schulte-Geers, and B. Sericola, "Optimization results for a generalized coupon collector problem," *Journal of Applied Probability*, vol. 53, no. 2, pp. 622 – 629, 2016. [Cited on page: 41.]

[144] J. Jockovic and B. Todiĉ, "Coupon collector problem with reset button," *Mathematics*, vol. 12, p. 239, 01 2024. [Cited on page: 41.]

[145] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995. [Cited on page: 50.]

[146] F.-H. Chang, H.-B. Chen, J.-Y. Guo, and Y.-P. Huang, "Multigroup testing for items with real-valued status under standard arithmetic," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 1084–1092, 2015. [Cited on page: 50.]

[147] A. Sharma and C. R. Murthy, "Computationally tractable algorithms for finding a subset of non-defective items from a large population," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7149–7165, 2017. [Cited on page: 66.]

[148] J. P. A. Ioannidis, "Infection fatality rate of COVID-19 inferred from seroprevalence

data," *Bulletin of the World Health Organization*, vol. 99, no. 1, pp. 19–33F, Jan. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7947934/ [Cited on page: 80.]

[149] V. Caturano, B. Manti, F. Carbone, V. A. Lasorsa, R. Colicchio, M. Capasso, A. Leonardi, G. Matarese, T. Russo, and P. Salvatore, "Estimating asymptomatic SARS-CoV-2 infections in a geographic area of low disease incidence," *BMC Infect. Dis.*, vol. 21, no. 350, Apr. 2021. [Cited on page: 80.]

[150] R. Subramanian, Q. He, and M. Pascaul, "Quantifying asymptomatic infection and transmission of COVID-19 in New York city using observed cases, serology, and testing capacity," *Proc. Natl. Acad. Sci. U.S.A (PNAS)*, vol. 118, no. 9, Mar. 2021. [Cited on page: 80.]

[151] S. A. McDonald, F. Miura, E. R. A. Vos, M. B. van, M. H de, K. F. van der, B. R van, H. G den, and J. Wallinga, "Estimating the asymptomatic proportion of SARS-CoV-2 infection in the general population: Analysis of a nationwide serosurvey in the Netherlands," *medRxiv*, Mar. 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/03/31/2021.03.29.21254334 [Cited on page: 80.]

[152] S. Liu, H. Luo, Y. Wang, D. Wang, S. Ju, and Y. Yang, "Characteristics and associations with severity in COVID-19 patients: A multicentre cohort study from Jiangsu province, China," *The Lancet PrePrint*, Feb. 2020. [Online]. Available: https://ssrn.com/abstract=3548753 [Cited on page: 80.]

[153] K. Mizumoto, K. Kagaya, A. Zarebski, and G. Chowell, “Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan,” *Euro Surveill.*, vol. 25, no. 10, Feb. 2020. [Cited on page: 80.]

[154] M. A. Johansson, T. M. Quandelacy, S. Kada, P. V. Prasad, M. Steele, J. T. Brooks, R. B. Slayton, M. Biggerstaff, and J. C. Butler, “SARS-CoV-2 transmission from people without COVID-19 symptoms,” *JAMA*, vol. 4, no. 1, pp. 1406–1407, Jan. 2021. [Cited on page: 80.]

[155] Y. Bai, L. Yao, T. Wei, F. Tian, D.-Y. Jin, L. Chen, and M. Wang, “Presumed asymptomatic carrier transmission of COVID-19,” *JAMA*, vol. 323, no. 14, pp. 1406–1407, Feb. 2020. [Cited on page: 80.]

[156] E. Sanchez, “COVID-19 science: Why testing is so important,” *American Heart Association*, Apr. 2020. [Online]. Available: https://www.heart.org/en/news/2020/04/02/covid-19-science-why-testing-is-so-important [Cited on page: 80.]

[157] Centre for Disease Control and Prevention (CDC), “About Variants of the Virus that Causes COVID-19,” *Centre for Disease Control and Prevention*, Feb 2022. [Online]. Available: https://www.cdc.gov/covid/php/variants/index.html [Cited on page: 80.]

[158] A. Atkeson, M. Droste, M. J. Mina, and J. H. Stock, “Economic benefits of COVID-19 screening tests,” *medRxiv*, Oct. 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/11/01/2020.10.22.20217984 [Cited on page: 81.]

[159] B. Udugama, P. Kadhiresan, H. N. Kozlowski, A. Malekjahani, M. Osborne, V. Y. C. Li, H. Chen, S. Mubareka, J. B. Gubbay, and W. C. W. Chan, "Diagnosing COVID-19: The disease and tools for detection," *ACS Nano.*, vol. 14, no. 4, pp. 3822–3835, Mar. 2020. [Cited on page: 81.]

[160] J. H. Schefe, K. E. Lehmann, I. R. Buschmann, T. Unger, and H. Funke-Kaiser, "Quantitative real-time RT-PCR data analysis: Current concepts and the novel "gene expressions' CT difference" formula," *J. Mol. Med.*, vol. 84, no. 11, pp. 901–910, Nov. 2006. [Cited on page: 81.]

[161] B. W. Buchan, J. S. Hoff, C. G. Gmehlin, A. Perez, M. L. Faron, L. S. Munoz-Price, and N. A. Ledeboer, "Distribution of SARS-CoV-2 PCR cycle threshold values provide practical insight into overall and target-specific sensitivity among symptomatic patients," *Am. J. Clin. Pathol.*, vol. 154, no. 4, Sep. 2020. [Cited on page: 82.]

[162] P. K. Chatki and A. G. Pise, "Perspective on the SARS-CoV-2 and its diagnostic tests," *Int. J. Curr. Res. (IJCRR)*, vol. 12, no. 19, Oct. 2020. [Cited on page: 82.]

[163] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1880–1901, 2012. [Cited on pages: 82 and 83.]

[164] J. Scarlett and O. Johnson, "Noisy non-adaptive group testing: A (near-)definite defectives approach," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3775–3797, 2020. [Cited on page: 83.]

[165] H. B. Petersen, B. Bah, and P. Jung, "Compressed sensing-based SARS-CoV-2

pool testing," 2020. [Online]. Available: https://arxiv.org/abs/2007.09171 [Cited on pages: 84, 108, 109, 120, 126, and 127.]

[166] J. L. Gevertz, S. M. Dunn, and C. M. Roth, "Mathematical model of real-time PCR kinetics," *Biotechnol. Bioeng.*, vol. 92, no. 3, pp. 346–355, Nov. 2005. [Cited on pages: 84, 87, 88, and 128.]

[167] H.-P. Wang, R. Gabrys, and A. Vardy, "PCR, tropical arithmetic, and group testing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 963–968. [Cited on page: 85.]

[168] ——, "Tropical group testing," *IEEE Trans. Inf. Theory*, vol. 69, no. 9, pp. 6098–6120, 2023. [Cited on page: 85.]

[169] V. Paligadu, O. Johnson, and M. Aldridge, "Small error algorithms for tropical group testing," *IEEE Trans. Inf. Theory*, vol. 70, no. 10, pp. 7232–7250, 2024. [Cited on page: 85.]

[170] N. Kwan and L. Wang, "Constructions for nonadaptive tropical group testing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2023, pp. 1627–1632. [Cited on page: 85.]

[171] X. Xia, Y. Liu, B. Yang, Y. Liu, J. Cui, and Y. Zhang, "An Expectation Maximization based adaptive group testing method for improving efficiency and sensitivity of large-scale screening of COVID-19," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 2, pp. 482–493, 2022. [Cited on page: 86.]

[172] H. Matsushima, Y. Tajima, X.-N. Lu, and M. Jimbo, "Efficient pooling designs and screening performance in group testing for two types of defectives," in *IEEE Annu.*

Comput. Softw. and Appl. Conf. (COMPSAC). Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2024, pp. 1996–2000. [Cited on page: 86.]

[173] W. Chen, C. Tatsuoka, and X. Lu, "HiBGT: High-performance Bayesian group testing for COVID-19," in *IEEE Int. Conf. on High Perf. Comput. Data, and Analytics (HiPC)*, 2022, pp. 176–185. [Cited on page: 86.]

[174] W. Chen, H. Qi, X. Lu, and C. Tatsuoka, "SBGT: Scaling Bayesian-based group testing for disease surveillance," in *IEEE Int. Parallel and Distrib. Process. Symp. (IPDPS)*, 2023, pp. 951–962. [Cited on page: 86.]

[175] D. L. Donoho, "Compressed Sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006. [Cited on pages: 87, 90, 94, and 112.]

[176] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008. [Cited on pages: 87, 90, and 112.]

[177] C. B. F. Vogels, A. F. Brito, A. L. Wyllie, J. R. Fauver, I. M. Ott, C. C. Kalinich, M. E. Petrone, A. Casanovas-Massana, M. C. Muenker, A. J. Moore, J. Klein, P. Lu, A. Lu-Culligan, X. Jiang, D. J. Kim, E. Kudo, T. Mao, M. Moriyama, J. E. Oh, A. Park, J. Silva, E. Song, T. Takahashi, M. Taura, M. Tokuyama, A. Venkataraman, O.-E. Weizman, P. Wong, Y. Yang, N. R. Cheemarla, E. B. White, S. Lapidus, R. Earnest, B. Geng, P. Vijayakumar, C. Odio, J. Fournier, S. Bermejo, S. Farhadian, C. S. D. Cruz, A. Iwasaki, A. I. Ko, M. L. Landry, E. F. Foxman, and N. D. Grubaugh, "Analytical sensitivity and efficiency comparisons of SARS-CoV-2

RT–qPCR primer–probe sets," *Nat. Microbiol.*, vol. 5, no. 10, pp. 1299—-1305, Jul. 2020. [Cited on pages: 88, 101, 120, 127, and 128.]

[178] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing.* Springer New York, 2013. [Cited on pages: 90 and 159.]

[179] N. G. Polson and L. Sun, "Bayesian l_0-regularized least squares," *Applied Stochastic Models in Business and Industry*, vol. 35, no. 3, pp. 717–731, 2019. [Cited on page: 94.]

[180] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge Univ. Press, 1990. [Cited on pages: 94 and 96.]

[181] D. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2845–2862, 2001. [Cited on page: 94.]

[182] M. Elad and A. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2558–2567, 2002. [Cited on page: 94.]

[183] M. J. Abdi, *Cardinality Optimization Problems.* University of Birmingham, 2013. [Online]. Available: https://books.google.co.in/books?id=8d3-oAEACAAJ [Cited on page: 94.]

[184] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms.* Cambridge Univ. Press, 2014, pp. 150—166. [Cited on page: 94.]

[185] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Proc. 29th Ann. Conf. on Learn. Theory*, vol. 49. PMLR, 23–26 Jun 2016, pp. 1246–1257. [Cited on pages: xvi, 96, 160, and 161.]

[186] I. Panageas and G. Piliouras, "Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions," in *Proc. 8th Innovations in Theor. Comp. Sc. (ITCS)*, vol. 67, Dagstuhl, Germany, 2017, pp. 2:1–2:12. [Cited on page: 96.]

[187] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016. [Cited on page: 97.]

[188] Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012. [Cited on page: 97.]

[189] T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimization problems," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3466–3474, 2013. [Cited on page: 98.]

[190] R. Kueng and P. Jung, "Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 689–703, 2018. [Cited on page: 100.]

[191] M. A. Khajehnejad, A. G. Dimakis, W. Xu, and B. Hassibi, "Sparse recovery of nonnegative signals with minimal expansion," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 196–208, 2011. [Cited on page: 100.]

[192] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2006. [Cited on pages: 110, 157, and 158.]

[193] P. Sasmal, P. Jampana, and C. S. Sastry, "Sparse recovery guarantees for block orthogonal binary matrices constructed via generalized Euler squares," 2019. [Online]. Available: https://arxiv.org/abs/1907.07396 [Cited on pages: 110 and 158.]

[194] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. USA: Prentice-Hall, Inc., 1989. [Cited on page: 112.]

[195] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Machine Learning.*, vol. 3, no. 1, Jan. 2011. [Cited on pages: 115 and 116.]

[196] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014. [Cited on page: 117.]

[197] Z.-Q. Luo, W.-K. Ma, A. M.-c. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010. [Cited on page: 119.]

[198] C. Thrampoulidis, E. Abbasi, and B. Hassibi, "LASSO with non-linear measurements is equivalent to one with linear measurements," in *Proc. 28th Int. Conf. on Neur. Inf. Process. Syst. (NeurIPS)*, ser. NIPS '15, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 3420–3428. [Cited on page: 126.]

[199] X. Jiang, G. Raskutti, and R. Willett, “Minimax optimal rates for Poisson inverse problems with physical constraints,” *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4458–4474, 2015. [Cited on page: 126.]

[200] M. A. Attia, W.-T. Chang, and R. Tandon, “Heterogeneity aware two-stage group testing,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3977–3990, 2021. [Cited on page: 134.]

[201] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine learning*, vol. 79, pp. 151–175, 2010. [Cited on page: 144.]

[202] H. Karimi, J. Nutini, and M. Schmidt, “Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition,” *arXiv e-prints*, Aug. 2016. [Cited on page: 161.]