

Multi-Level Group Testing with Application to One-Shot Pooled COVID-19 Tests

Alejandro Cohen, Nir Shlezinger, Amit Solomon, Yonina C. Eldar, and Muriel Médard

Abstract—One of the main challenges in containing the Coronavirus disease 2019 (COVID-19) pandemic stems from the difficulty in carrying out efficient mass diagnosis over large populations. The leading method to test for COVID-19 infection utilizes qualitative polymerase chain reaction, implemented using dedicated machinery which can simultaneously process a limited amount of samples. A candidate method to increase the test throughput is to examine pooled samples comprised of a mixture of samples from different patients. In this work we study pooling-based COVID-19 tests. We identify the specific requirements of COVID-19 testing, including the need to characterize the infection level and to operate in a one-shot fashion, which limit the application of traditional group-testing (GT) methods. We then propose a multi-level GT scheme, designed specifically to meet the unique requirements of COVID-19 tests, while exploiting the strength of GT theory to enable accurate recovery using much fewer tests than patients. Our numerical results demonstrate that multi-level GT reliably and efficiently detects the infection levels, while achieving improved accuracy over previously proposed one-shot COVID-19 pooled-testing methods.

I. INTRODUCTION

The Coronavirus disease 2019 (COVID-19) pandemic has already forced lockdowns all over the globe, and has claimed more than a million lives worldwide. In order to handle and contain the pandemic in the absence of a vaccine, large portions of the population should be frequently tested [1]. One of the main difficulties in doing so stems from the limited testing resources and the lengthy duration required to identify the presence of an infection [2].

The main diagnosis tool for COVID-19 tests is based on RNA extraction via qualitative reverse transcription polymerase chain reaction (RT-qPCR). Although various technological alternatives have been proposed [3], [4], the RT-qPCR process remains the leading and most widely spread method for COVID-19 testing. The output of this procedure represents an estimate of the viral load in the tested sample [5]. The main bottleneck associated with this form of COVID-19 testing follows from the fact that each RT-qPCR machine can simultaneously process a fixed number of samples, and its procedure tends to be quite lengthy, on the order of a few hours for each test.

A promising method to increase the efficiency of COVID-19 tests is based on pooling [6], [7]. Here, each sample processed by the RT-qPCR machine is comprised of a mixture of several samples taken from different patients. When the infected patients constitute a small subset of the overall tested individuals, pooling-based schemes allow accurate recovery using a number of tests which is notably smaller than the number of patients [8]. Several recovery schemes for pooled COVID-19 tests have been recently proposed [7], [9]–[15], which can be divided according to the two main mathematical frameworks for such recovery procedures: The first is group testing (GT) theory, originally derived for detecting infections in large populations [16], which was used in [6], [7], [9]; The second framework is compressed sensing (CS), which deals with the recovery of sparse signals [17],

This project received funding from the Benozio Endowment Fund for the Advancement of Science, the Estate of Olga Klein - Astrachan, the European Union's Horizon 2020 research and innovation program under grant No. 646804-ERC-COG-BNYQ, and from the Israel Science Foundation under grant No. 0100101. A. Cohen, A. Solomon and M. Médard are with the Research Laboratory of Electronics, MIT, Cambridge, MA (email: {cohenale, amitsol, medard}@mit.edu). N. Shlezinger is with the School of ECE, Ben-Gurion University of the Negev, Be'er-Sheva, Israel (e-mail: nirshl@bgu.ac.il). Y. C. Eldar is with the Faculty of Math and CS, Weizmann Institute of Science, Rehovot, Israel (e-mail: yonina@weizmann.ac.il).

and was utilized for pooled COVID-19 tests in [10]–[15].

The main difference between classical GT and CS is that GT deals with group detection problems, which results in binary variables, i.e., each subject can be either infected or not infected [8], while CS-based methods result in real-valued variables. Although group testing is traditionally adaptive, requiring multiple sequential tests based on previous ones [16], it can also be applied in a one-shot (non-adaptive) manner [18]. CS focuses on the one-shot recovery of sparse real-valued vectors from lower-dimensional linear projections, and thus each subject can take any real value number [17]. The additional domain knowledge of GT, namely, the more restricted domain over which it operates compared to CS, allows it in some applications to achieve improved recovery guarantees over CS, in terms of fewer measurements required, as suggested in [19], [20] for the quantization of sparse signals. Nonetheless, in the context of COVID-19 testing, one is often interested not only in detecting whether or not a patient is infected, but also in some score on the level of the viral load [11], as can be provided using CS tools. The fact that each of these mathematical frameworks has its own pros and cons for COVID-19 testing, motivates the design of a recovery method which combines GT with one-shot operation and multi-level detection, as in CS.

In this work we propose a multi-level GT recovery scheme for pooled COVID-19 testing, which is a GT-based method designed to account for the unique characteristics of pooled COVID-19 tests. The proposed technique extends GT schemes to detect multiple levels of viral load, building upon our previous results on combining GT with CS concepts and multi-level discretization in [19]. The resulting multi-level GT scheme operates in a one-shot manner, and is designed to avoid dilution due to mixing too many samples in each pool [21].

We begin by identifying the specific requirements which arise from the setup of pooled COVID-19 testing. In light of these requirements, we derive the multi-level GT method, which is comprised of a dedicated testing matrix, i.e., pooling pattern, as well as a GT-based recovery method. Our numerical evaluations, which use the model proposed in [11] for pooled RT-qPCR testing, demonstrate that the proposed multi-level GT scheme reliably recovers the infection levels, while operating at a limited computational burden, and achieving improved accuracy over previously proposed CS-based approaches.

The rest of this paper is organized as follows: In Section II we review the pooled COVID-19 testing setup. Section III presents the proposed multi-level GT scheme. Section IV details the simulation study, and Section V provides concluding remarks.

Throughout the paper, we use boldface lower-case letters for vectors, e.g., \mathbf{x} . Matrices are denoted with boldface upper-case letters, e.g., \mathbf{M} . Sets are expressed with calligraphic letters, e.g., \mathcal{X} , and \mathcal{X}^n is the n th order Cartesian power of \mathcal{X} .

II. SYSTEM MODEL

In this section we present the system model for which we derive the recovery algorithm described in Section III. As our main application is pooled COVID-19 tests, we begin by identifying the specific characteristics which arise from this application in Subsection II-A, based on which we present our problem formulation in Subsection II-B.

A. Pooled COVID-19 Tests

The common approach in testing for the presence of COVID-19 is based on the RT-qPCR method. Here, a sample is collected, most

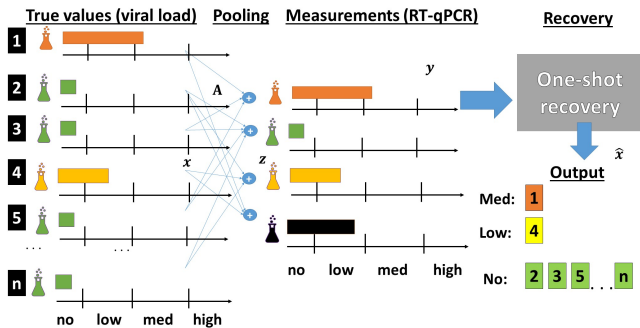


Fig. 1: Pooled RT-qPCR testing with one-shot recovery.

commonly swab-based. The presence of infection is then examined by RNA extraction via RT-qPCR measurements, quantifying the viral load in the sample. The RT-qPCR process is quite time consuming (on the typical order of several hours), and can simultaneously examine up to a given number of m inputs (commonly on the order of several tens of samples). This results in a major bottleneck, particularly when the number of patients, denoted by n , is much larger than m .

A candidate approach to reduce the test duration, which is considered in this paper, utilizes pooling [6]. Pooling is based on mixing the samples of groups of patients together, forming m mixed samples out of the overall n patients. Then, the presence of COVID-19 for each of the tested individuals is recovered from the mixed RT-qPCR measurements, either directly, i.e., in a *one-shot* fashion, or in an adaptive manner, which involves additional tests [22]. To formulate the problem of designing pooling-based recovery techniques for COVID-19 tests, we note the following characteristics of this testing procedure which have to be accounted for:

- A1: The number of infected measurements, denoted by k , is much smaller than the number of tested individuals n . Typically $k \leq 0.1n$, i.e., up to 10% of the tested population is infected.
- A2: One is interested not in only identifying whether a subject is infected or not, but also in some discrete score on the viral load. For example, possible outputs are *no* (no virus), *low* (borderline), *mid* (infected), and *high* (highly infected).
- A3: The RT-qPCR measurements are noisy, i.e., some level of random distortion is induced in the overall process, encapsulating the randomness in the acquisition of the samples, their mixing, and the RT-qPCR reading.
- A4: It is preferable to carry out one-shot tests, i.e., fully identify all subjects from a single RT-qPCR operation, without having to carry out additional tests based on the results.
- A5: There is a limit, denoted by $L > 1$, on the number of subjects which can be mixed together in a single measurement. A typical limit on the number of subjects mixed together is $L = 32$ [6]. Furthermore, the portion taken from each sample for the pooled measurements is identical, e.g., one cannot mix 30% from one patient with 70% from another patient into a single pool.

An illustration of the overall flow of pooled RT-qPCR-based COVID-19 testing along with the desired one-shot recovery operation is depicted in Fig. 1. On the left side of the figure, we see the true viral loads of all n items. In particular, we see that the first item is infected in a medium level, the fourth item is infected in a low level, and all other items are not infected. Next, pooling is done based on a testing matrix, which is generated prior to obtaining the samples. For example, the first pooled test involves samples from the first, third, and fifth items. This results in a measurements vector, denoted by z . This vector is fed to the recovery algorithm, which is able to tell in one-shot that the first item is infected in a medium level, the fourth item is infected in a low level, and all other items are not infected.

B. Problem Formulation

Based on the characteristics of pooled COVID-19 tests detailed above, we consider the following problem: Let $\mathbf{x} \in \mathcal{R}_+^n$ be a vector whose entries are the viral loads of the n patients. By A1 it holds that \mathbf{x} is k -sparse, i.e., $\|\mathbf{x}\|_0 \leq k$. The pooling operation is represented by the matrix $\mathbf{A} \in \{0, 1\}^{m \times n}$. Let $l(i) \leq L$ denote the number of subjects mixed together in the i th individual pool, $i \in \{1, \dots, m\}$. This implies that the i th row of \mathbf{A} , denoted \mathbf{A}_i^T , is $l(i)$ -sparse by A5. The viral loads of the pooled samples are represented by the vector $\mathbf{z} \in \mathcal{R}_+^m$, whose entries are given by

$$z_i = \frac{1}{l(i)} \mathbf{A}_i^T \mathbf{x}, \quad i \in \{1, \dots, m\}, \quad (1)$$

where the factor $\frac{1}{l(i)}$ and the structure of \mathbf{A} guarantee that identical portions are taken from each sample in a pool-test, as required in A5. The RT-qPCR measurements, denoted by $\mathbf{y} \in \mathcal{R}_+^m$, are given by some stochastic mapping $f: \mathcal{R}_+ \mapsto \mathcal{R}_+$ of \mathbf{z} , which represents the distortion detailed in A3. We write the measurement as $\mathbf{y} = f(\mathbf{z})$, where $f(\cdot)$ is carried out element-wise.

To formulate our objective, we note that by A2, we are interested in recovering a discrete representation of the viral load. We thus define the discretization mapping $Q: \mathcal{R}_+ \mapsto \mathcal{Q}$, where \mathcal{Q} is a finite set containing the possible decisions. Our goal is thus to design an algorithm which maps the RT-qPCR measurements \mathbf{y} into an estimate of the discretized viral loads, denoted by $\hat{\mathbf{x}} \in \mathcal{Q}^n$, for the objective of minimizing the error probability, defined as

$$e(\hat{\mathbf{x}}) \triangleq \frac{1}{n} \sum_{i=1}^n \Pr(Q(x_i) \neq \hat{x}_i). \quad (2)$$

The fact that $\hat{\mathbf{x}}$ is obtained directly from \mathbf{y} indicates that the algorithm operates in a one-shot fashion, as required in A4.

To conclude, for the subset of infected items, of size k , from n , the goal in multi-level pooled COVID-19 tests is to design an $m \times n$ one-shot measurement matrix which guarantees that at most L subjects are mixed in each pool-test and a recovery algorithm, such that by observing \mathbf{z} we can identify the subset of infected items and their discrete representation of the viral load.

III. MULTI-LEVEL GROUP TESTING

In this section, we provide an efficient scheme which implements GT with multiple decisions. The sparsity assumption A1 implies that the recovery of pooled RT-qPCR tests can be treated as a sparse recovery problem, as was also noted in previous works on pooled COVID-19 testing [7], [9]–[15]. Sparse recovery is typically studied under either the framework of GT [8], or that of CS [17]. Broadly speaking, GT deals with sparse recovery of binary variables, i.e., it can recover whether a subject is infected or not. In order to evaluate the actual levels of each tested subject, as requested in A2, one would have to re-run the RT-qPCR test, violating requirement A4. The alternative approach of CS operates over the domain of real numbers, namely, it attempts to identify the exact cycle threshold or viral load for each subject, and thus tends to be less accurate compared to GT, as it does not exploit the fact that one is only interested in a discrete grade value by A2. This motivates the derivation of a dedicated algorithm for pooled COVID-19 recovery, which harnesses the strength of GT theory while extending it to the multi-level domain. The proposed multi-level GT method designed in light of the model assumptions detailed in the previous section, is presented in Subsection III-A, followed by a discussion in Subsection III-B.

A. Pooled-Testing Algorithm

Multi-level GT is comprised of three components: The design of the testing matrix \mathbf{A} ; the pooling operation; and the recovery algorithm which determines the discrete level associated with each

subject based on the results of the pooled tests. We next elaborate on each of these components.

1) *Testing Matrix*: To determine the testing matrix \mathbf{A} , we first set the number of the pool-tests m to be $(1+\epsilon)k \log_2 n$, for some $\epsilon > 0$. This is the sufficient number of pool-test for reliable recovery in GT using the optimal Maximum likelihood (ML) decoder [23], [24].

Once m is fixed, we proceed to setting \mathbf{A} . The traditional GT method of generating \mathbf{A} is to draw its elements in an i.i.d. fashion according to a Bernoulli distribution with parameter p . A Common choice for p in GT theory is $p = 1 - 2^{-1/k}$, for which the probability of each element in \mathbf{z} to be zero is $1/2$. This choice of p is obviously impractical, as k is unknown, so p is chosen using a rough approximation of k . A major drawback of this approach is that **A5** is not necessarily satisfied. We therefore consider an alternative method of generating \mathbf{A} , which forces the columns of \mathbf{A} , as well as the rows of \mathbf{A} , to be “typical”. That is, we want every column/row to have exactly $\lceil p \cdot m \rceil \leq L$ and $\lfloor p \cdot n \rfloor$ ones, respectively. Since in practical COVID-19 testing setups, one is interested in using a fixed deterministic matrix, rather than having to work with random matrices, we generate \mathbf{A} once before the pooling starts. That is, the same testing matrix can be used for multiple pooling experiments.

2) *Pooling*: After the testing matrix \mathbf{A} is generated, it determines which samples are pooled together. The pooling process is done as detailed in Section II.

3) *Recovery Algorithm*: The proposed recovery algorithm is given in Algorithm 1. It operates in two main steps as detailed next.

At the first step, the algorithm identifies efficiently all of the definitely defective (DD) items in two stages, without determining the infection level. In the first stage of the first step, the definitely not defective (DND) algorithm [25] is used (lines 2 and 14-22). Recall that the number of pool-tests m , corresponding to the testing matrix \mathbf{A} , is fixed to allow ML detection. The DND algorithm attempts to match the columns of \mathbf{A} with the vector \mathbf{y} . In particular, if column j of \mathbf{A} has a non-zero entry while the corresponding element in \mathbf{y} is zero, the column is assumed not to correspond to a defective subject. This algorithm finds most of the subjects that are DND and drastically reduces the number of possible defective (PD) items. In practice, as demonstrated empirically in Fig. 2b, in the non-asymptomatic regime as we consider herein for COVID-19, the set of subjects declared as PD after this stage, denoted \mathcal{P} , approximately satisfies $|\mathcal{P}| = \mathcal{O}(k)$. The remaining $n - |\mathcal{P}|$ subjects are declared not defective. In the second stage of the first step, the ML algorithm [23] is used only over the smaller set of PD subjects \mathcal{P} , to identify exactly the set \mathcal{D} of k DD subjects (lines 3 and 23-25). The ML algorithm looks for a collection of k columns in \mathbf{A} , for which \mathbf{y} is most likely. In the formulation $Q(\cdot)_{0,1}$ is a binary quantization mapping, and in the ML rule we denote by \mathcal{K} the set of actual defective subjects, and by $\Omega(\mathcal{P}, k)$ the set of $\binom{|\mathcal{P}|}{k}$ combinations of k defective subjects in \mathcal{P} .

In the second step, the algorithm estimates the infection level of each subject in the set of identified defective subjects \mathcal{D} , in an iterative fashion (lines 4-13). Let $\mathcal{D}(i)$ be the i -th element of \mathcal{D} , and $\mathbf{A}_{\mathcal{D}}$ denote the matrix created by taking the columns of \mathcal{D} from \mathbf{A} . For a test in which only one infected subject participates according to the testing matrix (lines 7-8), the algorithm can recover the viral load directly from the measurement (lines 9-10). To obtain a discrete score, the measured value is quantized using a threshold-based quantization mapping $Q(\cdot)$. Then the algorithm subtracts the viral load of that subject from all the tests in which it participates (line 11), and repeats until it recovers the infection levels of all declared infected subjects, denoted by \mathcal{S} (line 6).

B. Discussion

The novelty of our algorithm stems from the efficiency in the first step with few pool-tests, and the iterative process in the second step,

Algorithm 1 Multi-level GT recovery

Input: \mathbf{A}, \mathbf{y}

Output: \mathcal{S} ▷ \mathcal{S} contains index and infection level tuples

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
   Step 1: Detection of defective subjects
2:  $\mathcal{P} \leftarrow \text{DND}(Q_{0,1}(\mathbf{y}), \mathbf{A})$  ▷  $\mathcal{P}$  contains PD items
3:  $\mathcal{D} \leftarrow \text{ML}(Q_{0,1}(\mathbf{y}), \mathbf{A}, \mathcal{P})$  ▷  $\mathcal{D}$  contains DD items
   Step 2: Iterative viral load detection
4:  $\tilde{\mathbf{A}} \leftarrow \mathbf{A}_{\mathcal{D}}$ 
5:  $\tilde{\mathbf{y}} \leftarrow \sum_{i \in \mathcal{D}} \mathbf{A}_i$ 
6: while  $|\mathcal{S}| < K$  do
7:   for all  $i$  s.t.  $\tilde{\mathbf{y}}_i = 1$  do
8:      $j \leftarrow \text{index s.t. } \tilde{\mathbf{A}}_{i,j} = 1$ 
9:      $\hat{\mathbf{x}}_i \leftarrow Q(\mathbf{y}_j)$ 
10:     $\mathcal{S} = \mathcal{S} \cup \{(\mathcal{D}(j), \hat{\mathbf{x}}_i)\}$ 
11:     $\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}} - \mathbf{A}_j$ 
12:   end for
13: end while
   Definitely Not Defective (DND) [18] for  $m = (1 + \epsilon)k \log_2 n$ 
14: procedure  $\text{DND}(Q_{0,1}(\mathbf{y}), \mathbf{A})$ 
15:    $\mathcal{P} \leftarrow \{1, \dots, n\}$ 
16:   for all  $i$  s.t.  $Q_{0,1}(\mathbf{y})_i = 0$  do
17:     for all  $j$  s.t.  $\mathbf{A}_{i,j} = 1$  do
18:        $\mathcal{P} \leftarrow \mathcal{P} \setminus \{j\}$ 
19:     end for
20:   end for
21:   return  $\mathcal{P}$ 
22: end procedure
   Maximum Likelihood (ML) [23] only for  $\mathcal{P}$  (not for  $n$ )
23: procedure  $\text{ML}(Q_{0,1}(\mathbf{y}), \mathbf{A}, \mathcal{P})$ 
24:   return  $\arg \max_{\mathcal{D} \in \Omega(\mathcal{P}, k)} P(Q_{0,1}(\mathbf{y}) \mid \mathbf{A}, \mathcal{K} = \mathcal{D})$ 
25: end procedure

```

which determines the discrete infection level of each item. Here, we note a few remarks arising from the proposed multi-level GT scheme.

In Subsection III-A we describe how the testing matrix \mathbf{A} is generated. The description involves a random generation procedure, for which the resulting matrix is not guaranteed to satisfy **A5**. The motivation for using such random procedures as in GT theory stems from their provable performance guarantees [8]. Once a typical testing matrix satisfying **A5** is selected, one does not have to generate a new matrix for each group of n patients.

According to GT theory, for n i.i.d. tested individuals, the algorithm which maximizes the probability of finding the infected items (though not necessarily their levels) is the ML algorithm. However, its complexity is burdensome, as it has to consider $\binom{n}{k}$ options [26], [27]. An efficient alternative is the DND algorithm, also known as column matching [18], [25], [28], whose time complexity is $\mathcal{O}(kn \log n)$ [24]. However, it requires a greater amount of pooled measurements compared to the ML algorithm in order to reliably identify the defective items. Our proposed multi-level GT method combines the concepts of DND with the ML algorithm, while extending them to operate over non-binary fields, i.e., recover multiple levels rather than just identifying which subject is defective. Performing DND on all n items using the number of tests set to allow ML detection, i.e., $m = (1+\epsilon)k \log_2 n$, results in a smaller set of PD subjects \mathcal{P} . Given \mathcal{P} , the ML decoder has to consider significantly less options, $\binom{|\mathcal{P}|}{k}$, which is likely to be computationally feasible and considerably faster than considering all $\binom{n}{k}$ combinations.

In the second step of the algorithm proposed, we analyze the tests in which only one infected subject participates according to \mathbf{A} . This process, when restricted to operate over binary variables, was proposed as means to identify the DD subjects in GT [25]. Unlike [25], we utilize this process in iterative fashion after the DD subjects

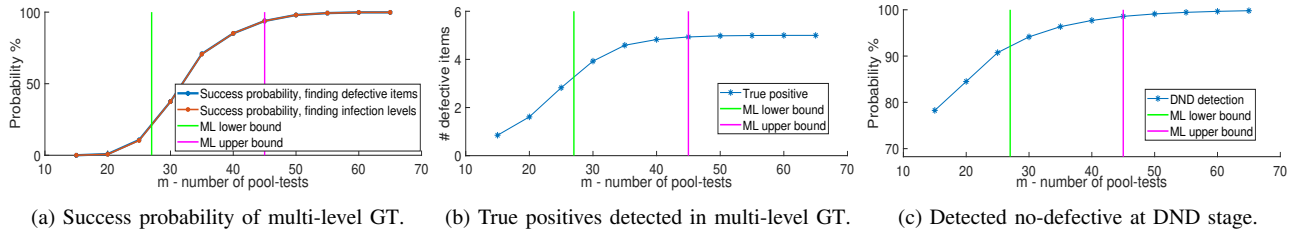


Fig. 2: Performance evaluation of one-shot multi-level GT over 200 iterations, with $n = 105$, $k = 5$ and with 4 infection levels.

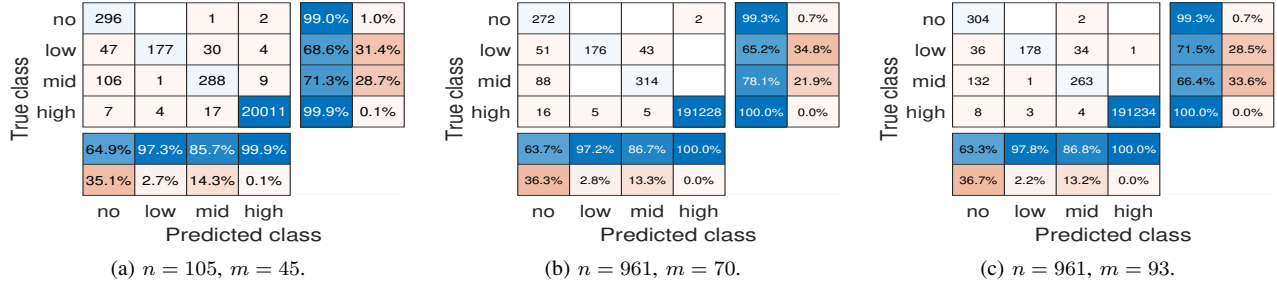


Fig. 3: Confusion matrices of one-shot multi-level GT.

are already detected to determine the viral load infection over the non-binary field. By subtracting the viral load of each detected subject in the iterative algorithm, we reduce the number of pool-tests in [25] to the one required in the ML algorithm. In the case where it is possible to identify the number of infected subjects in a pool-test, by observing its measurement, we noticed empirically that it is possible to achieve the lower bound number of pool test suggested for GT using ML algorithm, i.e., $m \approx \log_2 \binom{n}{k} \leq k \log_2(n/k)$ [18], [23]. We leave this interesting case and the analytical and complexity analysis of the proposed iterative algorithm as future work.

IV. NUMERICAL EVALUATIONS

We assessed the performance of our method using the RT-qPCR-based COVID-19 test model of [11]. The elements of the test matrix are chosen such that the rows and columns of \mathbf{A} are typical. The number of items selected in the first example is $n = 105$, out of which $k = 5$ items are defective, and with 4 infection levels. The viral load of each defective item is drawn from a uniform distribution between $[0, 1000]$ as in [11]. The infection level score is based on the following division of this interval into 4 regions: $[0, 50) = \text{no}$; $[50, 300) = \text{low}$ (borderline); $[300, 700) = \text{mid}$; and $> 700 = \text{high}$.

The success probability of Algorithm 1 versus m is depicted in Fig. 2a, compared to the ML lower/upper bound on the number of tests m from GT theory, as described in [23]. We evaluate forms of the success probability measure: here an error is declared when there is at least a single defective subject who is not detected out of a set of n patients, as well as when there is at least one patient whose infection level is not correctly recovered. From the plot, we see that the success probability of finding the defective items and the success probability of finding the infection levels coincide. That is, whenever the defective set was recovered successfully, the correct infection levels were also estimated successfully, indicating the validity of step 2 of Algorithm 1. We also see that when the number of tests is at the ML upper bound, the probability of success approaches one.

Fig. 2b shows the number of true positives, i.e. the number of defective subjects that are declared defective. We can see that when the number of tests m matches the upper bound, we have no false negatives with high probability. In the context of COVID-19, false negatives are considered to be the worst outcomes of a test. Fig. 2c illustrates the probability of the detected non-defective subjects after the DND stage in Algorithm 1. This is calculated as the number of non-defective items declared by the DND algorithm, divided by the

total number of non-defective items. We see that when the number of tests is as the upper bound of ML dictates, the DND algorithm identifies $\approx \frac{95}{100} = 95\%$ of the subjects as non-defective, i.e., $|\mathcal{P}| \approx 2k$. These will be candidates to be tested in the second stage of ML algorithm as PD, demonstrating the notable complexity reduction achieved by the two-step process.

The results in Figs. 2a-2c evaluate the error probability, but do not capture which forms of errors are produced when using Algorithm 1. Therefore, we depict in Fig. 3 the confusion matrix for the considered scenario, as well as when repeating the setup with a much larger amount of patients of $n = 961$, using merely $m \in \{70, 93\}$ pool-tests. The values $(n, m) \in \{(105, 45), (961, 93)\}$ were also used in [11], which applied CS tools for recovery. Observing Fig. 3, we note that most of the errors reported in fact correspond to identifying low-level and mid-level subjects as mid and high, respectively. Such errors are much less harmful in COVID-19 tests compared to reporting non-infected subjects as defective, which occurs only $\approx 0.1\%$ of the times for $(n, m) = (105, 45)$ in Fig. 3a, which is similar to the results achieved in [11] for such setups. This behavior is more notable when jointly testing $n = 961$ subjects in Figs. 3b-3c. Comparing these results to [11], it is noted that multi-level GT achieves improved false positive and false negative probabilities with only $m = 70$ test-pools compared to that achieved using all CS methods examined in [11] with $m = 93$ test-pools. For instance, for $(n, m) = (961, 93)$ [11] reported false positive probabilities varying from 0.1% to 0.8%, while the corresponding probability in Fig. 3b is 0.0%. This indicates the potential of multi-level GT in facilitating pooled testing of large numbers of subjects. The noise in the RT-qPCR measurements causes degradation in the level's detection. In further work, we will consider an additional algorithmic step to reduce this noise degradation.

V. CONCLUSIONS

In this work we proposed a scheme coined multi-level GT for one-shot pooled COVID-19 tests. We first identified the unique characteristics and requirements of RT-qPCR-based COVID-19 tests. Based on these requirements, we designed multi-level GT to combine traditional GT methods with one-shot operation and multi-level outputs, while implementing a preliminary DND detection mechanism to facilitate recovery at reduced complexity. Our numerical evaluations demonstrate that multi-level GT reliably identifies the infection levels when examining a much smaller number of samples compared to the number of tested subjects.

REFERENCES

- [1] M. Salathé, C. L. Althaus, R. Neher, S. Stringhini, E. Hodcroft, J. Fellay, M. Zwahlen, G. Senti, M. Battegay, A. Wilder-Smith *et al.*, “COVID-19 epidemic in switzerland: on the importance of testing, contact tracing and isolation.” *Swiss medical weekly*, vol. 150, no. 11-12, p. w20225, 2020.
- [2] E. J. Emanuel, G. Persad, R. Upshur, B. Thome, M. Parker, A. Glickman, C. Zhang, C. Boyle, M. Smith, and J. P. Phillips, “Fair allocation of scarce medical resources in the time of Covid-19,” 2020.
- [3] C. Lucia, P.-B. Federico, and G. C. Alejandra, “An ultrasensitive, rapid, and portable coronavirus SARS-CoV-2 sequence detection method based on CRISPR-Cas12,” *bioRxiv*, 2020.
- [4] N. Ben-Assa, R. Naddaf, T. Gefen, T. Capucha, H. Hajjo, N. Mandelbaum, L. Elbaum, P. Rogov, D. A. King, S. Kaplan *et al.*, “SARS-CoV-2 on-the-spot virus detection directly from patients,” *medRxiv*, 2020.
- [5] T. Nolan, R. E. Hands, and S. A. Bustin, “Quantification of mRNA using real-time RT-PCR,” *Nature protocols*, vol. 1, no. 3, pp. 1559–1582, 2006.
- [6] I. Yelin, N. Aharoni, E. Shaer-Tamar, A. Argoetti, E. Messer, D. Berenbaum, E. Shafran, A. Kuzli, N. Gandali, T. Hashimshony *et al.*, “Evaluation of COVID-19 RT-qPCR test in multi-sample pools,” *MedRxiv*, 2020.
- [7] R. Hanel and S. Thurner, “Boosting test-efficiency by pooled testing strategies for SARS-CoV-2,” *arXiv preprint arXiv:2003.09944*, 2020.
- [8] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, “Group testing and sparse signal recovery,” in *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE, 2008, pp. 1059–1063.
- [9] R. Ben-Ami, A. Klochendler, M. Seidel, T. Sido, O. Gurel-Gurevich, M. Yassour, E. Meshorer, G. Benedek, I. Fogel, E. Oiknine-Djian *et al.*, “Large-scale implementation of pooled RNA extraction and RT-PCR for SARS-CoV-2 detection,” *Clinical Microbiology and Infection*, 2020.
- [10] N. Shental, S. Levy, S. Skorniakov, V. Wuvshet, Y. Shemer-Avni, A. Porgador, and T. Hertz, “Efficient high throughput SARS-CoV-2 testing to detect asymptomatic carriers,” *medRxiv*, 2020.
- [11] S. Ghosh, R. Agarwal, M. A. Rehan, S. Pathak, P. Agrawal, Y. Gupta, S. Consul, N. Gupta, R. Goyal, A. Rajwade *et al.*, “A compressed sensing approach to group-testing for COVID-19 detection,” *arXiv preprint arXiv:2005.07895*, 2020.
- [12] 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,” *arXiv preprint arXiv:2004.05759*, 2020.
- [13] H. B. Petersen, B. Bah, and P. Jung, “Practical high-throughput, non-adaptive and noise-robust SARS-CoV-2 testing,” *arXiv preprint arXiv:2007.09171*, 2020.
- [14] 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,” *arXiv preprint arXiv:2007.14919*, 2020.
- [15] J. Zhu, K. Rivera, and D. Baron, “Noisy pooled PCR for virus testing,” *arXiv preprint arXiv:2004.02689*, 2020.
- [16] R. Dorfman, “The detection of defective members of large populations,” *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.
- [17] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge university press, 2012.
- [18] 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.
- [19] A. Cohen, N. Shlezinger, S. Salamatian, Y. C. Eldar, and M. Médard, “Serial quantization for sparse time sequences,” *arXiv preprint arXiv:1907.01691*, 2019.
- [20] —, “Distributed quantization for sparse time sequences,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5580–5584.
- [21] J. Yi, M. Cho, X. Wu, R. Mudumbai, and W. Xu, “Optimal pooling matrix design for group testing with dilution (row degree) constraints,” *arXiv preprint arXiv:2008.01944*, 2020.
- [22] J. M. Hughes-Oliver and W. H. Swallow, “A two-stage adaptive group-testing procedure for estimating small proportions,” *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 982–993, 1994.
- [23] 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.
- [24] A. Cohen, A. Cohen, and O. Gurewitz, “Secure group testing,” *IEEE Transactions on Information Forensics & Security*. DOI:10.1109/TIFS.2020.3029877, 2020.
- [25] 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.
- [26] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, “Information-theoretic and algorithmic thresholds for group testing,” *IEEE Trans. Inf. Theory*, 2020.
- [27] M. Aldridge, O. Johnson, and J. Scarlett, “Group testing: an information theory perspective,” *arXiv preprint arXiv:1902.06002*, 2019.
- [28] W. Kautz and R. Singleton, “Nonrandom binary superimposed codes,” *IEEE Tran. Inf. Theory*, vol. 10, no. 4, pp. 363–377, 1964.