Scripta
Spring 2024, Vol. 3, No. 1, 22-44
ISSN: 2769-9242

Scaling Up COVID-19 Testing: A Data Compression Approach

Miraslau Kavaliou * and Anatolii Metel'skii †

Abstract

This research paper presents a novel scheme for expanding COVID-19 testing capacity using fewer tests, based on the pooling of biological samples. The methodology innovatively applies a binary approach to testing, wherein each test tube is identified as negative or positive. By combining samples and utilizing a simple multiplication operation to determine the presence of virus RNA, this approach demonstrates the possibility of diagnosing 100 people with just 55 tests. The scheme is efficiency depends on the infection rate, demonstrating greater effectiveness when less than 1 per 100 people are infected. The paper also explores the application of this method for data compression, showing a potential reduction in data size without loss of information. The findings suggest that such a scheme could potentially increase the number of individuals tested for COVID-19, allowing for more efficient use of resources, while also opening new avenues for data compression in digital systems.

Introduction

In the face of a global pandemic, the efficiency of diagnostic testing for COVID-19 becomes paramount. At first glance, the straightforward solution to test 100 individuals appears to be conducting 100 separate tests. However, this approach, while intuitive, is not the most efficient or resourceful. The necessity to maximize testing capabilities without sacrificing accuracy has led to the exploration of innovative strategies that depart from traditional methodologies. This paper proposes an analysis scheme that significantly reduces the number of required tests while maintaining diagnostic precision.

The primary goal of this paper is to introduce and evaluate a novel approach to COVID-19 testing that leverages pooled sample analysis. By implementing this method, it is possible to dramatically increase the number of individuals tested with a finite number of tests. Such an approach is particularly valuable in scenarios where testing resources are limited or when aiming to scale up testing capacities rapidly.

*Junior Student, Fulton Science Academy.
Contact: mkavaliou@fultonscienceacademy.org
† Professor of Mathematics, Belarusian National Technical University.
Contact: ametelskii@gmail.com

Although the concept of pooled testing is not entirely new, its application and optimization for COVID-19 testing present distinct challenges and opportunities. This paper will explore these dimensions, comparing the proposed method against traditional testing approaches and highlighting its efficiency gains. A brief review of existing literature on pooled testing strategies provides context and underscores the innovation of the proposed scheme.

Organized into several key sections, this study begins with a detailed explanation of the pooled testing methodology, followed by an analysis of its impact on testing efficiency. A case study, focusing on the application of this method in the USA as of June 1, 2020, illustrates its potential to quadruple the number of individuals tested without additional resources. The final sections discuss the broader implications of these findings for public health policy and the management of current and future pandemics.

Through this introduction and the subsequent detailed exploration, this paper aims to contribute a pragmatic and scalable solution to a pressing global health challenge, offering a pathway to more efficient pandemic management through innovative testing strategies.

Text Extension

The crude idea is that two or more biological sample material can be combined in one test tube. This approach is not exactly new (see for example [1]), but the algorithm proposed prior uses the "degree positivity of the test", i.e. assumes a continuous or multi-valued scale when assessing the amount of antibodies to COVID-19, which is problematic. Our idea, however, of identifying test samples (or test tubes) uses a nominative scale: "negative" sample (Virus RNA not detected) or sample "positive" (Virus RNA detected).

Assigning the value 1 to a negative sample (or test tube), and 0 to a positive sample, we can describe the union of two samples x_1, x_2 can be described by a simple multiplication operation:

$$x_1 \times x_2 = 1 \times 1 = 1,$$

if both samples are negative, and

$$x_1 \times x_2 = 1 \times 0 = 0 \times 1 = 0 \times 0 = 0,$$

if at least one sample is positive.

Consider this case when all samples are divided into pairs. The proposed test scheme is as follows. We take two samples: x_1, x_2 and divide each into two parts (keeping the original sample for further action), then we combine samples x_1, x_2 into one test tube A. Remember that the variables x_1, x_2 can have only two values: 0 (positive) or 1 (negative).

The following situations are possible (a flowchart is provided at the end of this section for reference):

Case 1. Test tube A is tested to be negative (we'll write it like this: $A = 1$), then $x_1 \times x_2 = 1$. Solving this equation, we have $x_1 = x_2 = 1$, i.e. both samples are negative. Two samples are checked by one test: $(1, 1)$. Recall that 1 represents negative sample with no Virus RNA. Then let's denote the probability of a negative sample as p and the probability of a positive sample as $q = 1 - p$ (for example in USA on June 1, 2020 $q = 1828344/17449404 = 0.10478$, $p = 0.89522$).

The exact probability of this considered case is:

$$p_1 = P(1, 1) = p \times p = p^2.$$

In this case, the sequence of tests is: $A = 1$, for a total of 1 test.

Case 2. Test tube A is tested to be positive ($A = 0$): $x_1 \times x_2 = 0$. Checking the sample x_1. We see that sample x_1 is negative: $x_1 = 1$, then from the equation $x_1 \times x_2 = 0$ we get that $x_2 = 0$. Thus, through two tests we have: $(1, 0)$; the first sample is negative, the second is positive.

The likelihood of such a situation is:

$$p_2 = P(1, 0) = p \times q = pq.$$

In this case, the sequence of two tests: $A = 0, x_1 = 1$, for a total of 2 tests.

Case 3. Test tube A and sample x_1 are both positive ($A = 0, x_1 = 0$). Sample value x_2 will be set by additional testing: $x_2 = t_2$ ($t_2 = 0$ or 1). Through three tests, we will know $(0, t_2)$.

The chance of this situation is:

$$p_3 = P(0, x_2) = q.$$

In this case, the test sequence is: $A = 0, x_1 = 0, x_2 = t_2$.

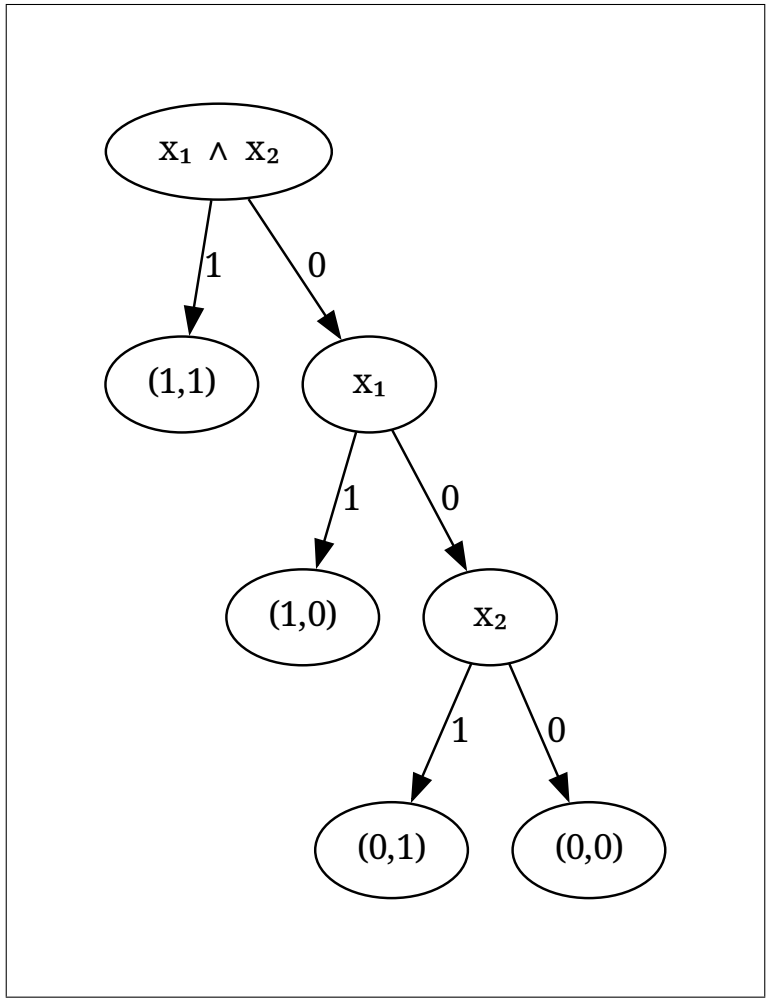

Figure 1: Flowchart of the two samples casework

So, with the testing scheme outlined, we have found p_1, p_2, p_3 to be the probabilities: p^2, pq, q. As $p + q = 1$, we can confirm that the sum of all probabilities is equal to 1, i.e., that all possible cases have been considered:

$$p_1 + p_2 + p_3 = p^2 + pq + q = p(p + q) + q = 1.$$

Now let's find the expected number $m_1(p)$ of tests to recognize two samples according to the proposed scheme. Let a_i be the number of tests

in the i-th situation, then solve:

$$\begin{aligned} m_1(p) &= a_1p_1 + a_2p_2 + a_3p_3 \\ &= 1p^2 + 2pq + 3q \\ &= p^2 + 2p(1-p) + 3(1-p) \\ &= 3 - p - p^2. \end{aligned}$$

Direct testing of two samples requires two tests. Let us now find which values of p (probability that a sample taken at random is negative) satisfy the inequality $m_1(p) < 2$. That is, we find for what values of p, our scheme is more efficient (more tested people) and more economical (less financial costs for the same number of contingent). For this we will solve the inequality

$$m_1(p) = 3 - p - p^2 < 2,\ p \in [0, 1].$$

We find that we need $p > 0.618034$ for the sample pooling scheme to be effective compared to the traditional scheme, when the number of tests is equal to the number of samples.

Example. Consider the sample set (01111011101111011111). Here $p = 16/20 = 0.8 > 0.618034$, i.e., we expect the sample pooling scheme to be effective.

We split the array into twos:

$$(01), (11), (10), (11), (10), (11), (11), (01), (11), (11).$$

To count the required number of tests, we identify each case and the corresponding number of tests. In this example: case 2, $(x_1, x_2) = (0, 1)$, appears 2 times, with 3 tests each, case 1, $(1, 1)$, appears 6 times with 1 test each, case 2, $(1, 0)$, appears 2 times with 2 tests each.

Therefore, there are 16 tests in total and the length of the array (number of samples) is 20.

Generalization

So, to test N people according to the proposed scheme it will take about $(N/2)m_1(p)$ tests. In this context, $m_1(p)$ represents the number of tests required per group of people based on the probability pp (the ratio of positive cases). Since the testing scheme likely involves grouping people and testing

groups rather than individuals, $N/2$ is the number of such groups formed from N people (assuming each group contains two people). Therefore, the total number of tests needed is the number of groups multiplied by the number of tests per group. Hence, M tests will allow $M(2/m_1(p))$ people to be tested.

The value $k_1(p) = 2/m_1(p)$ is called the efficiency coefficient a scheme that uses the splitting of samples into twos. Coefficient efficiency $k_1(p)$ shows how many samples can be identified by one test. When p changes from 0.618034 to 1, the efficiency factor $k_1(p)$ varies from 1 to 2.

For example,

$k_1(0.8) = 1.28205$, $k_1(0.9) = 1.55039$, $k_1(0.95) = 1.74292$, $k_1(0.99) = 1.94194$.

That is, if $p = 0.8$, then 100 tests can test about 128 people, etc.

For the value $p = 0.89522$ (as of June 1, 2020), $k_1(0.89522) = 1.53449$, therefore, the 300,000 tests performed daily in the US allow examine $300,000 \times k_1(0.89522) = 300,000 \times 1.53449 = 460347$ people, i.e. 160,347 more! Daily and without a reduction in credibility!

We can consider other options for combining n samples: $x_1, x_2, \ldots, x_n$. The combination of samples will correspond to the product of some set of variables x_i. The testing procedure is reduced to the study uniqueness of the solution of some system of algebraic equations, the left side of which is the product of a set of variables with two possible values: 0 or 1. If the solution is not unique, then we must determine an additional variable, which corresponds to an additional test.

For example, we detail the scheme of dividing samples into quadruples.

We take four samples: x_1, x_2, x_3, x_4 and divide each into two parts (we must save the source material in case further testing is required), combine all samples x_1, x_2, x_3, x_4 into tube A, and if required, samples x_1, x_2 into tube B, samples x_3, x_4 into tube C. We divide into cases:

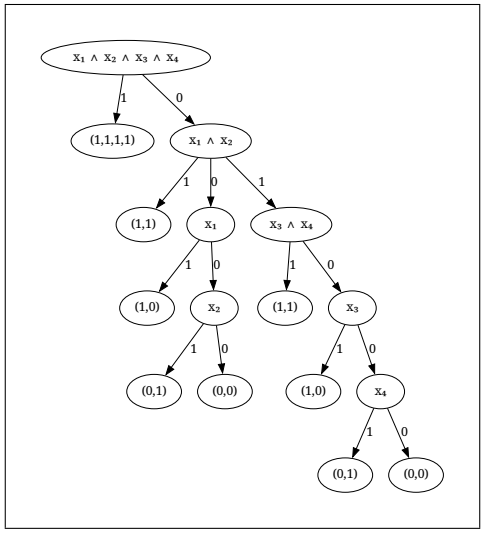

Figure 2: Flowchart of the four samples casework

Case 1. Tube A negative ($A = 1$):

$$x_1 \times x_2 \times x_3 \times x_4 = 1.$$

Solving this equation, we have $x_1 = x_2 = x_3 = x_4 = 1$, i.e., all four samples are negative. Four samples are checked by one test: $(1, 1, 1, 1)$. Recall that 1 means negative sample. In this case, the sequence of tests is $\{A = 1\}$.

The likelihood of such a situation is

$$p_1 = P(1, 1, 1, 1) = p \times p \times p \times p = p^4.$$

Case 2. Tube A is positive and tube B is negative. This gives the system:

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 1 \end{cases}.$$

Hence we conclude that $x_1 = x_2 = 1$, $x_3 \times x_4 = 0$. This necessitates another test, so we test x_3. In this case, we assume $x_3 = 1$, so $x_4 = 0$. Therefore, the unique solution is $(x_1, x_2, x_3, x_4) = (1, 1, 1, 0)$.

The likelihood of such a situation

$$p_2 = P(1, 1, 1, 0) = p \times p \times p \times q = p^3 q.$$

In this case, the test sequence is $\{A = 0, B = 1, x_3 = 1\}$, using three tests total.

Case 3. Similar to case 2, we test tube A to be positive, and B to be negative. Now we test $x_3 = 0$, which necessitates the use of a fourth test on x_4. This gives the system:

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 1 \\ x_3 = 0 \\ x_4 = t_4 \end{cases},$$

with solution $(1, 1, 0, t_4)$.

The likelihood of such a situation

$$p_3 = P(1, 1, 0, x_4) = p \times p \times q \times 1 = p^2 q.$$

In this case, the test sequence is $\{A = 0, B = 0, x_3 = 0, x_4 = t_4\}$, using four tests total.

Case 4. Tubes A, B test positive, while C tests negative. We use our fourth test on x_1. Consider when $x_1 = 1$, then

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 1 \\ x_1 = 1 \end{cases},$$

which has unique solution $(1, 0, 1, 1)$.

The likelihood of such a situation is

$$p_4 = P(1, 0, 1, 1) = p \times q \times p \times p = p^3 q.$$

In this case, the test sequence is $\{A = 0, B = 0, C = 1, x_1 = 1\}$ requiring four tests.

Case 5. Tubes A, B test positive, while C tests negative. We use our fourth

test on x_1. Consider when $x_1 = 0$, then

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 1 \\ x_1 = 0 \end{cases},$$

which has solution $(0, t_2, 1, 1)$.

The likelihood of such a situation

$$p_5 = P(0, x_2, 1, 1) = q \times 1 \times p \times q = p^2 q.$$

In this case, the test sequence is $\{A = 0, B = 0, C = 1, x_1 = 0, x_2 = t_2\}$ using four tests.

Case 6. All three tubes A, B, C test positive. In the first case, we test $x_1 = 1$ and $x_3 = 1$. We have a system:

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 0 \\ x_1 = 1 \\ x_3 = 1 \end{cases},$$

with the only solution $(1, 0, 1, 0)$.

The likelihood of such a situation

$$p_6 = P(1, 0, 1, 0) = p \times q \times p \times q = p^2 q^2.$$

In this case, the test sequence $\{A = 0, B = 0, C = 0, x_1 = 1, x_3 = 1\}$ using five tests.

Case 7. All three tubes A, B, C test positive. Let $x_1 = 1$ and $x_3 = 0$. Sample x_4 is not defined, therefore, its value is set by additional testing.

We have the system

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 0 \\ x_1 = 1 \\ x_3 = 0 \\ x_4 = t_4 \end{cases},$$

which has solution $(1, 1, 0, t_4)$.

The likelihood of such a situation

$$p_7 = P(1, 1, 0, x_4) = p \times p \times q \times 1 = p^2 q.$$

In this case, the test sequence is $\{A = 0, B = 0, C = 0, x_1 = 1, x_3 = 0, x_4 = t_4\}$, necessitating six tests.

Case 8. All three tubes A, B, C test positive. Let $x_1 = 0$, then testing $x_2 = t_2$. Testing x_3, let $x_3 = 1$.

We have the system

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 0 \\ x_1 = 0 \\ x_2 = t_2 \\ x_3 = 1 \end{cases}$$

The likelihood of such a situation

$$p_8 = P(0, x_2, 1, 0) = q \times 1 \times p \times q = pq^2.$$

In this case, the test sequence is $\{A = 0, B = 0, C = 0, x_1 = 0, x_2 = t_2, x_3 = 1\}$, which includes six tests.

Case 9. All three tubes A, B, C test positive. Let $x_1 = 0$, then testing

$x_2 = t_2$. Testing x_3, let $x_3 = 0$. Then we test $x_4 = t_4$. We have the system

$$\begin{cases} x_1 \times x_2 \times x_3 \times x_4 = 0 \\ x_1 \times x_2 = 0 \\ x_3 \times x_4 = 0 \\ x_1 = 0 \\ x_2 = t_2 \\ x_3 = 0 \\ x_4 = t_4 \end{cases},$$

which has solution $(0, t_2, 0, t_4)$. The likelihood of such a situation

$$p_9 = P(0, x_2, 0, x_4) = q \times 1 \times q \times 1 = q^2.$$

In this case, the test sequence is $\{A = 0, B = 0, C = 0, x_1 = 0, x_2 = t_2, x_3 = 0, x_4 = t_4\}$ and includes seven tests.

At this point, we have exhausted all possibilities, as we can verify by finding the sum $p_1 + \cdots + p_9$ and verifying it is 1.

Let us find the expected number $m_2(p)$ of tests for recognizing four samples according to the proposed scheme (a_i denoting the number of tests in the i-th situation):

$$\begin{aligned} m_2(p) &= \sum_i a_i p_i \\ &= p^4 + 3p^3q + 4(p^2q + p^3q) + 5(p^2q^2 + p^2q) + 12pq^2 + 7q^2 \\ &= 7 - 2p - 3p^2 - p^4. \end{aligned}$$

Let us now find out for what values of the probability p the inequality $m_2(p) < 4$ is satisfied. That is, we find for what values of $p \in [0, 1]$ the given method is more efficient than direct testing using four tests.

Solving the inequality

$$m_2(p) = 7 - 2p - 3p^2 - p^4 < 4,\ p \in [0, 1].$$

We get that the condition $p > 0.685291$ is necessary for this scheme was effective compared to traditional scheme when the number of tests is equal to the number of samples.

When p changes from 0.685291 to 1, the efficiency factor $k_2(p) = 4/m_2(p)$

varies from 1 to 4. Compare the coefficients efficiencies $k_1(p) = 2/m_1(p)$ and $k_2(p) = 4/m_2(p)$ of circuits divided into twos and fours for different values of p. For example, when splitting into twos:

$k_1(0.8) = 1.28205$, $k_1(0.9) = 1.55039$, $k_1(0.95) = 1.74292$, $k_1(0.99) = 1.94194$.

In the second scheme (with division into quadruples) we would have:

$k_2(0.8) = 1.30276$, $k_2(0.9) = 1.89224$, $k_2(0.95) = 2.53486$, $k_2(0.99) = 3.57429$.

Solving the inequality

$$4/(7 - 2p - 3p^2 - p^4) > 2/(3 - p - p^2),$$

it can be found that the second scheme is more efficient for $p > 0.786151$. This is confirmed by comparing the given values of the coefficients efficiency $k_1(p)$ and $k_2(p)$ at $p = 0.8, 0.9, 0.95, 0.99$.

As of June 1, 2020 in the US, $p = 0.89522$, hence for the latest scheme expected number of tests to identify four samples $m_2(p) = 2.16303$. Accordingly, to diagnose 100 people one needs about 55 tests: $(100/4)2.16303 = 54.0758$. But if $p < 0.786151$, then the first scheme is more efficient. Let's say $k_1(0.7) = 1.10497 > k_2(0.7) = 1.0283$.

Recursive Formula for the Expected Number of Tests when Split into Groups of 2^{n+1} Trials

Let's get a recursive formula for calculating the expected number of tests when splitting into groups of 2^{n+1} samples.

According to the scheme proposed above, if the test of the four is 0, then we split it into two parts. Similarly, testing groups by 2^{n+1} trials will be reduced to testing groups of 2^n trials.

To write down the desired formula, let's start with the value $n = 2$, i.e. with splitting samples into quadruples. Recall that p is the probability of a negative samples, and $q = 1 - p$ is the probability of a positive sample. When splitting into fours ($n = 2$), each two is complemented by all possible (three) other twos. Therefore, $3 \times 3 = 9$ are possible different combinations (on the right is the probability of such combinations and required number

of tests):

$$\begin{aligned}
(1,1,1,1) &\Rightarrow P(1,1,1,1) = p\times p\times p\times p = p^4 && 1\,\text{test}\\
(1,1,1,0) &\Rightarrow P(1,1,1,0) = p\times p\times p\times q = p^3q && 3\,\text{tests}\\
(1,1,0,x_4) &\Rightarrow P(1,1,0,x_4) = p\times p\times q = p^2q && 4\,\text{tests}\\
(1,0,1,1) &\Rightarrow P(1,0,1,1) = p\times q\times p\times p = p^3q && 4\,\text{tests}\\
(1,0,1,0) &\Rightarrow P(1,0,1,0) = p\times q\times p\times q = p^2q^2 && 5\,\text{tests}\\
(1,0,0,x_4) &\Rightarrow P(1,0,0,x_4) = p\times q\times q = pq^2 && 6\,\text{tests}\\
(0,x_2,1,1) &\Rightarrow P(0,x_2,1,1) = q\times p\times p = p^2q && 5\,\text{tests}\\
(0,x_2,1,0) &\Rightarrow P(0,x_2,1,0) = q\times p\times q = pq^2 && 6\,\text{tests}\\
(0,x_2,0,x_4) &\Rightarrow P(0,x_2,0,x_4) = q\times q = q^2 && 7\,\text{tests.}
\end{aligned}$$

From here we get the expected number of tests for four samples

$$m_2(p) = p^4 + 3p^3q + 4p^2q + 4p^3q + 5p^2q^2 + 6pq^2 + 5p^2q + 6pq^2 + 7q^2 \quad (1)$$

and using $q = 1 - p$ we obtain

$$m_2(p) = 7 - 2p - 3p^2 - p^4.$$

Note that when dividing the samples into groups of eight, each set of four samples is paired with all possible combinations of the remaining four samples, resulting in nine possible pairings. Therefore, when splitting into groups of 2^3 samples, the number such eights are $9\times 9 = 81$. In general, the number of different sets containing 2^n samples will be equal to $s_n = 3^{2^{n-1}}$.

After analyzing the two cases considered above ($n = 1, 2$), we can get a recursive formula for the expected number of tests when splitting into groups of 2^{n+1} samples.

The expected number of tests when split into groups of 2^n sample has the form

$$m_n(p) = a_1p^{2^n} + a_2r_2(p,q) + \cdots + a_{s_n}r_{s_n}(p,q),\ a_1 = 1,\ q = 1-p, \quad (2).$$

Here p^{2^n}, $r_2(p,q),\ldots,r_{s_n}(p,q)$ are the probabilities of different cases each containing 2^n samples, while $a_1, a_2, \ldots, a_{s_n}$ are the number of tests required for evaluating a particular set. For example, for $n = 2$, expression (2) has the form of equation (1).

Note that using $q = 1 - p$ gives

$$p^{2^n} + r_2(p, 1-p) + \cdots + r_{s_n}(p, 1-p) = 1, \tag{3}$$

since it is the sum of the probabilities (proportions) of all the different sets 2^n samples.

Let us express $m_{n+1}(p)$ in terms of $m_n(p)$. To do this, we will create a function

$$f_n(k) = (a_1+k+1)p^{2^n} + (a_2+k+1)r_2(p,q) + \cdots + (a_{s_n}+k+1)r_{s_n}(p,q). \tag{4}$$

Let's simplify the last expression. Expanding the brackets, we get

$$\begin{aligned} f_n(k) &= (a_1 + k + 1)p^{2^n} + (a_2 + k + 1)r_2(p,q) + \cdots + (a_{s_n} + k + 1)r_{s_n}(p,q) \\ &= p^{2^n} + a_2 r_2(p,q) + \cdots + a_{s_n} r_{s_n}(p,q) \\ &+ (k+1)(p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)). \end{aligned}$$

In this way,

$$f_n(k) = m_n(p) + (k+1)(p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)),$$

and in view of (3) we use $q = 1 - p$ to obtain

$$f_n(k) = m_n(p) + (k+1).$$

Taking into account that when passing from n to $n+1$, each set of 2^n samples are padded with all possible 2^n others sets, we get that the expected number of tests with 2^{n+1} samples in the group will be equal to

$$\begin{aligned} m_{n+1}(p) = p^{2^n}(p^{2^n} &+ r_2(p,q)(a_2+1) + \cdots + r_{s_n}(p,q)(a_{s_n}+1)) \\ &+ r_2(p,q) f_n(a_2) + \cdots + r_{s_n}(p,q) f_n(a_{s_n}). \end{aligned} \tag{5}$$

Indeed, a set of the form $(\underbrace{1,1,\ldots,1}_{n},\underbrace{1,1,\ldots,1}_{n})$ has probability $p^{2^{n+1}}$ and is identified by one test. Sets $(\underbrace{1,1,\ldots,1}_{n},\underbrace{*,*,\ldots,*}_{n})$, where among the characters marked with $(*)$ there is at least one 0, have probabilities $p^{2^n} r_2(p,q), \ldots, p^{2^n} r_{s_n}(p,q)$. Testing the whole set $(1,1,\ldots,1,*,*,\ldots,*)$ get 0 by testing a subset $(1,1,\ldots,1)$ we get 1. Therefore, we conclude a priori that subset test $(*,*,\ldots,*)$ is 0. Probability $r_k(p,q)$ k-th subset

$(*, *, \ldots, *)$ is taken from (2) and for its identification will require $a_k - 1$ tests, since testing the entire subset of $(*, *, \ldots, *)$ is not necessary — its test is known to is 0. Thus, a set of the form $(1, 1, \ldots, 1, *, *, \ldots, *)$ in (5) match amount

$$p^{2^n}(p^{2^n} + r_2(p,q)(a_2+1) + \ldots + r_{s_n}(p,q)(a_{s_n}+1)).$$

Now consider a set of the form $(\underbrace{\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x}}_{n}, \underbrace{*, *, \ldots, *}_{n})$, where among characters, labeled (x) there is at least one 0, and a subset $(*, *, \ldots, *)$ — arbitrary of possible s_n pieces. We will assume that the subset $(\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x})$ has the probability $r_2(p,q)$, and the subset $(*, *, \ldots, *)$ has the probability $r_k(p,q), k = \overline{1, s_n}$ $(r_1(p,q) = p^{2^n})$. Then the probabilities of sets of the form $(\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x}, *, *, \ldots, *)$ are $r_2(p,q)r_1(p,q), \ldots, r_2(p,q)r_{s_n}(p,q)$. Testing the whole set $(\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x}, *, *, \ldots, *)$ get 0, identification subsets $(\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x})$, $(*, *, \ldots, *)$ according to (2) requires a_2 and a_k tests. Therefore, a set of the form $(\mathrm{x}, \mathrm{x}, \ldots, \mathrm{x}, *, *, \ldots, *)$ in (5) match amount

$$r_2(p,q)((a_1 + a_2 + 1)p^{2^n} + \ldots + r_{s_n}(p,q)(a_{s_n} + a_2 + 1))) = r_2(p,q)f_n(a_2)$$

according to (4).

By analogous reasoning, we obtain the remaining terms in (5):

$$r_3(p,q)f_n(a_3), \ldots, r_{s_n}(p,q)f_n(a_{s_n}).$$

We transform expression (5) in this way

$$\begin{aligned} m_{n+1}(p) = {} & p^{2^n}(p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)) - p^{2^{n+1}} \\ & + p^{2^n}(p^{2^n} + r_2(p,q)a_2 + \cdots + r_{s_n}(p,q)a_{s_n}) \\ & + r_2(p,q)f_n(a_2) + \cdots + r_{s_n}(p,q)f_n(a_{s_n}). \end{aligned}$$

Taking into account (2) and $f_n(a_i) = m_n(p) + a_i + 1$, $i = \overline{2, s_n}$, write

down

$$\begin{aligned} m_{n+1}(p) = {} & p^{2^n}(p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)) - p^{2^{n+1}} \\ & + p^{2^n} m_n(p,q) \\ & + r_2(p,q)(m_n(p) + a_2 + 1) \\ & + \cdots \\ & + r_{s_n}(p,q)(m_n(p) + a_{s_n} + 1). \end{aligned} \tag{6}$$

Expanding the brackets in (6) and grouping, we have

$$\begin{aligned} m_{n+1}(p) = {} & p^{2^n}(p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)) - p^{2^{n+1}} + m_n(p,q) \\ & + (p^{2^n} + r_2(p,q)a_2 + \cdots + r_{s_n}(p,q)a_{s_n}) - p^{2^n} \\ & + (p^{2^n} + r_2(p,q) + \cdots + r_{s_n}(p,q)) - p^{2^n}. \end{aligned}$$

Replacing $q = 1 - p$ and taking into account equations (2) and (3), we obtain the expected number of tests via p, the proportion of negative samples,

$$m_{n+1}(p) = 2m_n(p) + 1 - p^{2^n} - p^{2^{n+1}}, \tag{7}$$

where $m_1(p) = 3 - p - p^2$.

It is easy to verify that formula (7) is true for $n = 2$ by comparing the result obtained using (7) with the expression for $m_2(p)$ from section 1.

From the expressions for $m_n(p)$, $n = 1, 2, 3, 4$, we find that the partitions into twos, threes, fours, eights are effective for $p > 0.61804$, $p > 0.68529$, $p > 0.72726$, $p > 0.75104$ respectively.

Interval $p \in (p_*, p^*)$, where $k_n(p_*) = 1$, $k_n(p^*) = k_{n+1}(p^*)$, we call the efficiency interval 2^n partitions. Inside this interval $k_n(p) > 1$, but for $p > p^*$: $k_n(p) < k_{n+1}(p)$.

Thus, we have the following results.

$$m_1(p) = 3 - p - p^2,$$

with efficiency interval (0.61804; 0.78615): splitting into 2 samples is effective starting from 0.61804 and by 0.78615, and then splitting into 4 samples is effective.

$$m_2(p) = 7 - 2p - 3p^2 - p^4,$$

with efficiency interval (0.68529; 0.88665).

$$m_3(p) = 15 - 4p - 6p^2 - 3p^4 - p^8,$$

with efficiency interval (0.72726; 0.94162).

$$m_4(p) = 31 - 8p - 12p^2 - 6p^4 - 3p^8 - p^{16},$$

with efficiency interval (0.75104; 0.97037).

$$m_5(p) = 63 - 16p - 24p^2 - 12p^4 - 6p^8 - 3p^{16} - p^{32},$$

with efficiency interval (0.76329; 0.98507).

$$m_6(p) = 127 - 32p - 48p^2 - 24p^4 - 12p^8 - 6p^{16} - 3p^{32} - p^{64},$$

with efficiency interval (0.76933; 0.99251).

$$m_7(p) = 255 - 64p - 96p^2 - 48p^4 - 24p^8 - 12p^{16} - 6p^{32} - 3p^{64} - p^{128},$$

with efficiency interval (0.77232; 0.99624).

$$m_8(p) = 511 - 128p - 192p^2 - 96p^4 - 48p^8 - 24p^{16} - 12p^{32} - 6p^{64} - 3p^{128} - p^{256},$$

with efficiency interval (0.77380; 0.99812).

$$m_9(p) = 1023 - 256p - 384p^2 - 192p^4 - 96p^8 - 48p^{16} - 24p^{32} - 12p^{64} - 6p^{128} - 3p^{256} - p^{512},$$

effectively starting at $p = 0.77454$ etc.

Splitting into quadruples is more efficient than splitting into twos when $p > 0.78615$. Splitting into eights is more efficient than splitting into fours at $p > 0.88665$. Splitting into sixteen is more efficient than splitting into eights for $p > 0.94162$.

If we accept $p = 0.89522$ (USA, June 1, 2020), then it is more efficient splitting into eights $k_3(0.89522) = 1.87298$. So 300 000 tests performed daily in the United States, allow you to examine $300\,000 \times k(0.89522) = 300\,000 \times 1.87298 = 561\,894$ people, i.e., an astounding 261 894 more people daily and without loss of reliability.

When split into sixteen

$$k_4(0.95) = 3.04613,\ k_4(0.99) = 8.26646.$$

That is, at the beginning of the epidemic, 100 tests can check about 826 people.

Experimental Verification

For experimental verification of the proposed results, we have created [2] a Python script that simulates the optimization algorithm testing for COVID-19. A brief description of this script's steps is as follows:

1. Defining the total number of samples and the ratio of positive samples (`samplesNum`, `positiveRatio`).

2. Generating a pseudo-random sequence of samples with the specified `positiveRatio` (`False = negative`, `True = positive`).

3. Determining the optimal size of the group by a predefined function

   $$\texttt{groupSize} = 2^e,\ e = -\texttt{round(log2(positiveRatio)+0.643856)}.$$

   This function has been defined experimentally by running this script a total of 400 times with varying `positiveRatio` by 0.1, and with a `samplesNum` of 250,000. Leading to simulatively testing 100 million COVID-19 tests.

4. Splitting the sequence into sub-sequences by group size.

   Checking every sub-sequence using logical AND operation, so if any of the samples in the group is positive, the result is positive.

   If the result is negative, then every sample in the group must be negative and we move to the next sub-sequence.

   If the result is positive, the next step would depend on the size of the sub-sequence:

   i. if 2 samples then we check the first, if it is positive we have to check the second as well.

   ii. if 4 samples or more then we break it down further, by calling step 4 recursively with sub-sequence as an input and so on.

The function for calculating the optimal group size is not perfect compared to the simulation. In most cases, however, it does provide the best

group size value, and errors have a minimal effect on efficiency. We are able to see this error in Figure 3, where red represents our function and blue our simulation. In general, anywhere we see red there is error, yet this error is insignificant for use since it is never off by an amount larger than one.

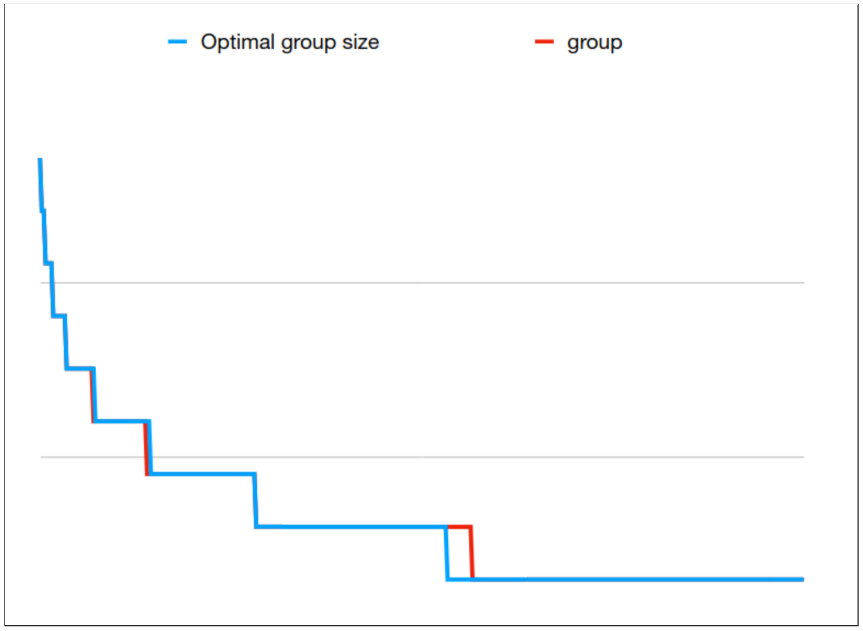

Figure 3: Showcase of minor error in function (red) and simulation (blue) [3]

Data Compression

We consider a set of samples as an array of zeros (positive sample) and units (negative sample). Our goal was to have this array match an array of tests of *shorter length*! That's why the proposed approach to sample analysis is applicable to the construction of the algorithm compression of binary data.

Based on the first testing algorithm, we split the file into pairs digits, and we have the following encoding table.

1. $(1,1) \Longleftrightarrow (1)$,
2. $(1,0) \Longleftrightarrow (0,1)$,
3. $(0,x_2) \Longleftrightarrow (0,0,x_2)$.

The expected number of characters to encode two (x_1, x_2) was calculated above to be $m_1(p) = 3 - p - p^2$, and for $p > 0.618034$ this the table ensures the inequality $m_1(p) < 2$, i.e., it ensures data compression. Here, p is the probability that a randomly taken symbol from the file is 1. p equals the ratio of the number of ones in the file to the total number of characters.

The compression coefficient $z_1(p)$, i.e., the ratio of the length of the compressed file to the length of the original $z_1(p) = m_1(p)/2$ changes from 1 to 0.5 when p changes from 0.618034 to 1.

The file is decoded based on the same table. If the file starts with 1, then it corresponds to (1, 1). If (0,1), then replace with (1,0). If (0,0), then we take the next symbol x_2 and put the triple $(0, 0, x_2)$ in matching two $(0, x_2)$. After, we take a segment of the file up to the next instance of "1", etc.

If there are more zeros in the file than ones, namely, the proportion of zeros satisfies the inequality $p > 0.618034$, then we swap "0" and "1":

1. $(0, 0) \Longleftrightarrow (0)$,
2. $(0, 1) \Longleftrightarrow (1, 0)$,
3. $(1, x_2) \Longleftrightarrow (1, 1, x_2)$.

Example. Let's take a file of 20 characters

$$(0111,\ 1011,\ 1011,\ 1101,\ 1111).$$

Clearly, $p = 16/20 = 0.8 > 0.618034$, so we expect the compressed file to be shorter length.

Splitting the file into two

$$(01,\ 11,\ 10,\ 11,\ 10,\ 11,\ 11,\ 01,\ 11,\ 11),$$

according to the given encoding table in compressed form, it will be encoded as

$$(001,\ 1,\ 01,\ 1,\ 01,\ 1,\ 1,\ 001,\ 1,\ 1),$$

i.e. it contains 16 characters.

Similarly, one can write an encoding table for compression data based on any other test protocol by means of sample pooling.

According to the second algorithm, we split the file into fours (x_1, x_2, x_3, x_4), and we have the following encoding table.

1. $(1,1,1,1) \Longleftrightarrow (1)$,
2. $(1,1,1,0) \Longleftrightarrow (0,1,1)$,
3. $(1,1,0,x_4) \Longleftrightarrow (0,1,0,x_4)$,
4. $(1,0,1,1) \Longleftrightarrow (0,0,1,1)$,
5. $(1,0,1,0) \Longleftrightarrow (0,0,0,1,1)$,
6. $(1,0,0,x_4) \Longleftrightarrow (0,0,0,1,0,x_4)$,
7. $(0,x_2,1,1) \Longleftrightarrow (0,0,1,0,x_2)$,
8. $(0,x_2,1,0) \Longleftrightarrow (0,0,0,0,x_2,1)$,
9. $(0,x_2,0,x_4) \Longleftrightarrow (0,0,0,0,x_2,0,x_4)$.

The expected number of characters to encode a quadruple (x_1, x_2, x_3, x_4) is equal to $m_2(p) = 7 - 2p - 3p^2 - p^4$. At $p > 0.685291$ we have the inequality $m_2(p) < 4$, which provides compression. As before, p is equal to the ratio of the number of ones in the file to the number of all characters.

The compression coefficient $z_2(p)$, i.e., the ratio of the length of the compressed file to the length of the original $z_2(p) = m_2(p)/4$ here changes from 1 to 0.25 when p changes from 0.685291 to 1.

Example. Let's take the same file of 20 characters

(0111, 1011, 1011, 1101, 1111).

We have $p = 16/20 = 0.8 > 0.685291$, so based on the encoding table get compressed file

(00101, 0011, 0011, 0101, 1)

of 18 characters.

The file is decoded according to the following rule. If a the file starts with 1, then we replace it with (1, 1, 1, 1). If (0,1,1), then replace with (1,1,1,0). If (0,1,0), then we take the next character x_4 and put this set in matching quadruple $(1,1,0,x_4)$. Next, we take a segment of the file up to next 1. If this is $(0,0,0,1,1)$, then it corresponds to $(1,0,1,0)$. If it is $(0,0,0,1,0)$ then take the next character x_4 and to this set we assign the quadruple $(1,0,0,x_4)$, etc.

Discussion/Author's comments

Of course, the implementation of the proposed optimization of testing on COVID-19 requires appropriate technological support, and the proposed data compression algorithm has rather a demonstrative than a practical purpose. However, the proposed scheme testing and associated random variable (number of tests) applicable to other tasks. For example, product quality control (the classical problem of identifying counterfeit coins that differ from suitable, say, with less weight) or the health of a chain composed of series-connected elements, etc.

The main purpose of this publication is to demonstrate a mathematical view of everyday life. Nurturing a mathematical style of thinking is the original goal of learning math! The latter is necessary for efficient and correct using the capabilities of computer technology and information technologies. "Stop Teaching Calculating! Start Teaching Maths!"—called by the Wolfram brothers, authors of the famous Mathematica package. The mathematical style of thinking is the perception of any situation as problematic, from the point of view of its possible optimization. In an advanced form, the mathematical way of thinking is the ability to analyze and formulate a problem situation on the language of mathematical concepts, the ability to synthesize a solution using not only well-known algorithms but analogies and borrowings from other fields of science, perhaps not directly related to this problem.

According to the classification of sciences presented by the Nobel laureate awards (1962) by the famous physicist Lev Landau, all sciences are divided into: 1) natural—physics, chemistry, etc., 2) unnatural—humanitarian, and 3) one supernatural—mathematics. "Numbers govern the world", said the Pythagoreans. The power of mathematics is in its universality: "Mathematics is the art of naming different things with the same name" (Henri Poincare, French mathematician) and beauty: "I love mathematics not only because it finds use in technology, but also because it is beautiful!" (Rosa Peter, Hungarian mathematician).

"What a strange world we live in!"

References

1. Daley, B. *The maths logic that could help test more people for coronavirus* https://theconversation.com/the-maths-logic-that-could-help-test-more-people-for-coronavirus-134287. 2020.
2. https://drive.google.com/file/d/1u7gfawiuJx8wuvLBWwaKWlEjGVYWVmr1/view?usp=sharing.
3. https://drive.google.com/file/d/1KlxhtJRlsf6AeIqOKqCDMcxHp4gh2gjl/view.