

# DOPE: D-Optimal Pooling Experimental design with application for SARS-CoV-2 screening

Yair Daon\*<sup>1,2</sup>, Amit Huppert<sup>1,3</sup>, and Uri Obolski<sup>1,2</sup>

<sup>1</sup>School of Public Health, Tel Aviv University, Tel Aviv, Israel

<sup>2</sup>Porter School of the Environment and Earth Sciences, Tel Aviv University, Tel Aviv, Israel

<sup>3</sup>The Gertner Institute for Epidemiology and Health Policy Research, Tel Hashomer, Israel

March 8, 2021

## Abstract

Testing individuals for the presence of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), the pathogen causing the coronavirus disease 2019 (COVID-19), is crucial for curtailing transmission chains. Moreover, rapidly testing many potentially infected individuals is often a limiting factor in controlling COVID-19 outbreaks. Hence, *pooling* strategies, wherein individuals are grouped and tested simultaneously, are employed. We present a novel pooling strategy that implements D-Optimal Pooling Experimental design (DOPE). DOPE defines optimal pooled tests as those maximizing the mutual information between data and infection states. We estimate said mutual information via Monte-Carlo sampling and employ a discrete optimization heuristic for maximizing it. DOPE outperforms common pooling strategies both in terms of lower error rates and fewer tests utilized. DOPE holds several additional advantages: it provides posterior distributions of the probability of infection, rather than only binary classification outcomes; it naturally incorporates prior information of infection probabilities and test error rates; and finally, it can be easily extended to include other, newly discovered information regarding COVID-19. Hence, we believe that implementation of Bayesian D-optimal experimental design holds a great promise for the efforts of combating COVID-19 and other future pandemics.

*Keywords:* Group testing, Monte-Carlo, Bayesian, epidemiology, COVID-19, RT-PCR

---

\*Yair Daon was supported by a post-doctoral fellowship from the Tel Aviv University Center for Combating Pandemics and the Raymond and Beverly Sackler Dean's Post-Doctoral fellowship. The authors would like to thank Profs. David M. Steinberg and Arie Tamir for valuable discussion, advice and support.

# 1 Introduction

During the current COVID-19 pandemic, large-scale testing efforts for detecting the presence of the SARS-CoV-2 virus, the causative agent of the disease, are crucial. Testing allows isolating infected individuals, thus breaking transmission chains. Testing for SARS-CoV-2 is typically done using RT-PCR (reverse transcriptase polymerase chain reaction, see Section 2.2 for details). Testing via RT-PCR kits can be a limiting factor, thus creating a bottleneck in screening and isolation efforts [26, 25]. The most common way to increase efficiency and throughput of RT-PCR tests is *pooling*. Pooling is the act of using samples from several different individuals in one RT-PCR test, hereby referred to as a *pool*. Several pooling strategies have been previously suggested [13, 30], analyzed [20, 3], and applied [4, 27, 5, 14]. The modus operandi of pooling is as follows: A result is observed for one or several pools, and then further action is taken. Usually, a negative result for a pool means all members of said pool are declared negative without any further testing. A positive result, on the other hand, may render some individuals positive or require further testing.

## 1.1 Pooling strategies

Pooling originated in the seminal work of Dorfman [13] in 1943. Since then, pooling has evolved into what is known today as group testing [1]. There are several common pooling strategies, and they are outlined below. Implementation details can be found in Supplementary Material A.

*Dorfman* pooling [13] starts by testing a predetermined number of individuals in a pool. If the pooled test result is negative, all pool members are declared negative. Otherwise, each one is tested separately. A large scale testing effort [4] has shown that Dorfman pooling can save 76% of RT-PCR tests.

In *recursive* pooling [20], if the first pooled test is positive, the pool is split into two and the process repeats. Otherwise, all pool members are declared negative. Thus, an individual is only declared positive if they are eventually tested separately and the test result is positive. One study showed a recursive pooling can potentially result in a seven-fold increase in throughput [14].

*Matrix* pooling [22] arranges a population of size  $N = mn$  in an  $m \times n$  matrix. Each row and column are then pooled and individuals in the intersection of positive rows and columns are tested separately. We were not able to find data of a real world implementation of matrix pooling.

## 1.2 Contribution

We develop DOPE (D-Optimal Pooling Experimental design), a novel Bayesian pooling strategy. DOPE identifies which choice of pools maximizes the mutual information between population infection state and pooled test data. This choice of mutual information as an optimization objective categorizes DOPE as a D-optimal experimental design technique [7] and results in superior performance of DOPE compared to competing strategies.

DOPE is a Bayesian strategy and as such, enjoys the common advantages of Bayesian methods. Assumptions on the population and RT-PCR test error rates are easily incorporated into a

prior and a likelihood model, respectively. Furthermore, DOPE allows the probabilities of infection to be naturally quantified via the posterior. These probabilities convey more information and allow greater flexibility compared to a binary test result.

Precise quantification of the above-mentioned probabilities of infection allows DOPE to perform trade-offs between error rates and number of tests as required. Most competing pooling strategies do not allow for such an adaptive property and hence do not have control over the number of tests or error rates.

Another advantage of DOPE is evident when considering edge cases in competing strategies. Consider Dorfman pooling: how should one act if the first pooled test is positive, yet all subsequent tests are negative? Similar events arise for recursive and matrix pooling as well, see implementation details in Supplementary Material A. Such events all have nonnegligible probabilities under the empirically estimated test error rates, and are likely to result in implementation problems. In contrast, there are no ambiguous events when DOPE is the strategy of choice. All test results are used for updating one’s beliefs via Bayes’ theorem.

Lastly, DOPE is useful across both high and low infection prevalence scenarios. Some competing strategies lose efficiency at high infection prevalence [13, 20, 3, 27]; others may suffer from increased false-negative rates due to unmet assumptions of sparsity [30]. DOPE, in contrast, is inherently adaptable and suitable for a wide range of infection prevalence levels.

## 2 Methods

DOPE is comprised of several components. Briefly, a Bayesian model for pooling is formulated and a design is defined as a combination of pools. An optimal design is defined as maximizing mutual information between population infection state and pooled test data. Calculating said mutual information proceeds via Monte-Carlo simulations. Then an optimal design is found via discrete optimization, data are collected and the process repeats.

### 2.1 Prior

The prior encodes the probability of every possible infection state of the tested population. We assume the following structure: The population is divided to disjoint clusters (e.g. families, work places, classrooms), each contains a (potential) initial source of primary infection, which occurs with probability  $P_p$ . A secondary infection of other members of the cluster occurs with probability  $P_s$  for each. If no primary infection occurs, the probability that nonprimary members of the cluster are infected is the infection prevalence in the general population  $P_b$ . Our assumptions are given below, with their corresponding notation:

- Population members are denoted  $\{1, \dots, N\}$ .
- The **population state** is captured in  $\theta \in \{0, 1\}^N$ . Individual  $h \in \{1, \dots, N\}$  is either infected or not, with  $\theta_h = 1$  or  $\theta_h = 0$ , respectively.
- The population is partitioned into  $M$  disjoint **clusters**  $C_1, \dots, C_M$ . A single cluster represents, e.g., a household.

- A **cluster**  $C$  is a tuple:  $C = (h_0, h_1, \dots, h_n)$ . We assume here, for sake of notation only, that all clusters contain the same number of members  $n + 1$ .
- For cluster  $C$  denote  $\theta^{(C)} := (\theta_{h_0}, \dots, \theta_{h_n})$ .
- A **primary** infection of  $h_0$  occurs with probability  $P_p$ .
- A **secondary** infection of any of  $h_1, \dots, h_n$  by  $h_0$  occurs independently with probability  $P_s$ .
- If no primary infection occurs,  $h_1, \dots, h_n$  are infected with the **basal** prevalence of infection in the general population  $P_b$ .

Since clusters are disjoint, their prior probabilities are independent:

$$\mathbb{P}(\theta) = \prod_{C \in \mathcal{C}_1, \dots, \mathcal{C}_M} \mathbb{P}(\theta^{(C)}). \quad (1)$$

Turning our attention to cluster  $C$ :

$$\begin{aligned} \mathbb{P}(\theta^{(C)}) &= \mathbb{P}(\theta_{h_0}) \prod_{j=1}^n \mathbb{P}(\theta_{h_j} | \theta_{h_0}) \\ &= [P_p \prod_{j=1}^n P_s^{\theta_{h_j}} (1 - P_s)^{1 - \theta_{h_j}}]^{\theta_{h_0}} [(1 - P_p) \prod_{j=1}^n P_b^{\theta_j} (1 - P_b)^{1 - \theta_{h_j}}]^{1 - \theta_{h_0}}. \end{aligned} \quad (2)$$

An explicit expression for  $\mathbb{P}(\theta)$  is easily found from equations (1) and (2). One can rightfully claim that our prior does not allow co-infection between nonprimary household members (e.g.  $h_2$  and  $h_3$ ). However, the difference in probabilities is negligible, see Supplementary Material A.

## 2.2 Likelihood

The likelihood encodes our assumptions on pooled tests and how they can err. Before delving into our probabilistic assumptions, we briefly explain the process of testing for SARS-CoV-2 by RT-PCR. Our exposition intentionally avoids many details and a comprehensive review of RT-PCR can be found in [6].

PCR is a process in which a targeted DNA strand's frequency is amplified to create billions of copies in a reaction mixture. At the end of this *amplification* process, the presence of targeted DNA molecules in a reaction mixture can be confidently determined. Arguably, the best way to understand PCR is as a chain reaction: Each targeted DNA molecule is copied into two, that are copied into four and so forth. This cascade gives PCR its name: polymerase chain reaction.

Specifically, the PCR process is comprised of cycles, with each cycle doubling the abundance of the targeted DNA. In each cycle, each double-strand DNA molecule is broken into two separate strands. An enzyme called DNA polymerase generates a new double-strand DNA replica from each separate strand. Replication cannot start without a specific short DNA sequence, called a primer, that has to be introduced into the reaction mixture. Introducing the right primer into

the reaction mixture ensures (almost) only the targeted DNA sequence is copied. A protein is added to the reaction mixture, which emits light upon successfully binding the targeted DNA. Once enough light is emitted the tested reaction mixture is declared positive and a *detection* is said to occur. If, on the other hand, a predefined number of cycles pass without a detection event then the reaction mixture is declared negative.

Since the genetic material of SARS-CoV-2 is RNA rather than DNA, an extra preprocessing step is required. An enzyme called reverse-transcriptase (hence the RT in RT-PCR) replicates existing RNA in the reaction mixture into DNA. Once DNA replicas are made, the PCR process proceeds as described above, targeting SARS-CoV-2 DNA replicas.

Failed detection of SARS-CoV-2 RNA in pooled RT-PCR testing is referred to below as a false-negative. One possible source of false-negatives in pooling is sample dilution. When pooling, several samples are mixed, so the concentration of viral RNA is reduced. This effect may cause a delay in amplification [4], no detection and, consequently, a false-negative. However, [35] showed that this effect can be safely ignored when mixing up to 32 samples, which we correspondingly assume.

Previous studies of group testing strategies assumed that the false-negative probability does not depend on the number of infected samples, but merely on the existence of at least one such sample in a pool [20, 3]. Current studies of pooling in the context of SARS-CoV-2 also employ similar assumptions [28, 8]. Specifically, these studies assume that the probability of a negative result is the same for a pool with a single sample coming from an infected individual and (e.g.) five. However, in a previous study, we have shown that this assumption does not align with experimental data [12]. Thus, we assume that viral RNA from each positive individual in a pool undergoes the RT-PCR amplification process independently. Consequently, the probability of (failed) amplification and/or detection for every sample whose source was an infected individual is considered separately.

Erroneous detection of SARS-CoV-2 RNA (a false-positive) in pooled RT-PCR testing can also occur. A common assumption [28, 8, 20, 3] is that the false-positive probability does not depend on the number of negative samples in a pool. We incorporate this assumption in our likelihood model with a small modification. We assume that an erroneous amplification can occur in *any* pool. Specifically, it is possible that correct amplifications fail and an erroneous one occurs simultaneously. This assumption is relatively specific for the current application of screening for SARS-CoV-2 via RT-PCR. For example, cross-reactivity with other coronaviruses would have violated this assumption, but it was ruled out in [32].

To summarize, we assume that for a single pool, a positive test result is generated in one of two paths. Either, SARS-CoV-2 RNA from an infected individual’s sample is correctly amplified and detected, and this can happen for each positive sample in a pool. Or, some erroneous amplification occurs (e.g. contaminant viral RNA is introduced), an event that occurs at most once per pool. Our model is illustrated in Figure 1. We now proceed to formulate the likelihood, so we require some definitions and notations:

- A **pool** is a collection of individuals  $\{h_1, \dots, h_m\} \subseteq \{1, \dots, N\}$ .
- A **design**  $T$  is a collection of pools. The  $k^{\text{th}}$  pool is denoted  $T_k$ .
- Data are denoted  $\mathbf{d} \in \{0, 1\}^{|T|}$ . We let  $\mathbf{d}_k = 1$  if upon testing  $T_k$  a positive result was

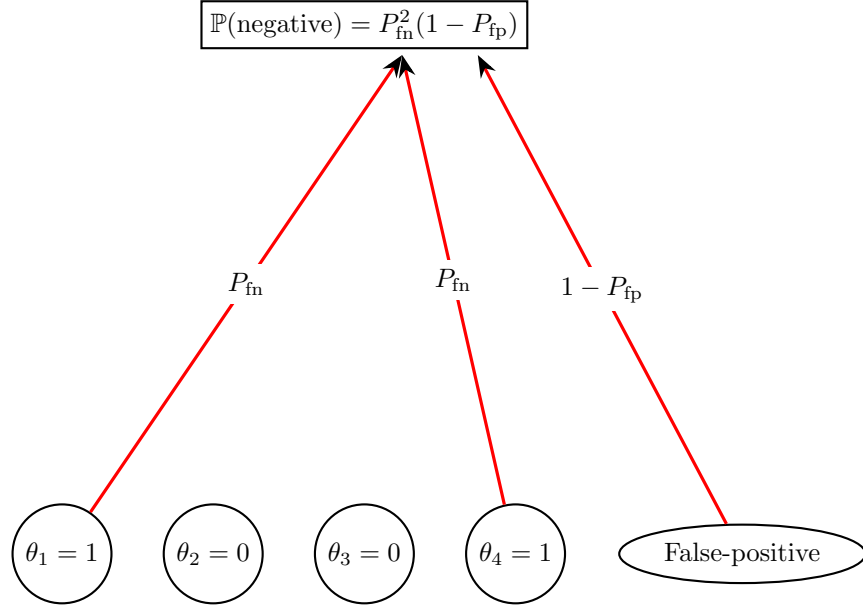


Figure 1: Illustration of the likelihood model of a pooled test result. A pool contains individuals  $\{1, 2, 3, 4\}$  with state  $\theta = (1, 0, 0, 1)$  (i.e. individuals 1 and 4 are infected). A negative pooled test implies that three detection paths failed. A false negative occurred for 1 and 4, each with probability  $P_{\text{fn}}$ . Additionally, no erroneous detection of SARS-CoV-2 occurred, with probability  $1 - P_{\text{fp}}$ . Individuals 2 and 3 are not infected and do not contribute to the probability of the pooled test result. A positive pooled test arises if any one of the above mentioned paths results in a detection.

observed and let  $\mathbf{d}_k = 0$  otherwise.

- The probability that the detection process fails for one sample taken from an infected individual is  $P_{\text{fn}}$ .
- The probability of an erroneous amplification and detection in a pooled test is  $P_{\text{fp}}$ .

The probability of a negative pooled test result is presented in (3a) (along with its complement (3b)), and explained below.

$$\mathbb{P}(\mathbf{d}_k = 0 | T_k, \theta) = (1 - P_{\text{fp}}) \prod_{h \in T_k} P_{\text{fn}}^{\theta_h}, \quad (3a)$$

$$\mathbb{P}(\mathbf{d}_k = 1 | T_k, \theta) = 1 - (1 - P_{\text{fp}}) \prod_{h \in T_k} P_{\text{fn}}^{\theta_h}. \quad (3b)$$

A negative pooled test result occurs when all detection paths (both correct and erroneous) fail. The probability of no false-detection accounts for the  $1 - P_{\text{fp}}$  term. The probability of no correct

detection is  $P_{\text{fn}}$  per infected individual. The probability that all such paths fail is the product of the above mentioned terms, displayed in (3a). The probability of a positive result, presented in (3b), is simply the complement. Combining (3a) and (3b), and recalling that  $\mathbf{d}_k \in \{0, 1\}$  yields:

$$\mathbb{P}(\mathbf{d}_k|T_k, \theta) = \left[ (1 - P_{\text{fp}}) \prod_{h \in T_k} P_{\text{fn}}^{\theta_h} \right]^{1 - \mathbf{d}_k} \left[ 1 - (1 - P_{\text{fp}}) \prod_{h \in T_k} P_{\text{fn}}^{\theta_h} \right]^{\mathbf{d}_k}. \quad (4)$$

Since different tests are assumed independent, the full likelihood is the product:

$$\mathbb{P}(\mathbf{d}|T, \theta) = \prod_{k=1}^{|T|} \mathbb{P}(\mathbf{d}_k|T_k, \theta). \quad (5)$$

### 2.3 D-optimal design

In Bayesian experimental design, a design is called *D-optimal* if it maximizes any one of several equivalent information theoretic design criteria [19, 7, 24, 29]. For convenience, we consider the mutual information between parameters and data as the optimization criterion. For a given design  $T$ , the mutual information between data  $\mathbf{d}$  and population infection state  $\theta$  is denoted  $\Psi(T)$ :

$$\Psi(T) := I(\theta; \mathbf{d}|T) = \sum_{\theta, \mathbf{d}} \mathbb{P}(\theta, \mathbf{d}|T) \log \frac{\mathbb{P}(\theta, \mathbf{d}|T)}{\mathbb{P}(\theta)\mathbb{P}(\mathbf{d}|T)}. \quad (6)$$

It is known that maximizing mutual information is equivalent to minimizing the expected posterior entropy and maximizing expected relative entropy between posterior and prior [7, 19, 17, 29]. Some details of the D-optimal approach are discussed in Section 4.

There is no closed form expression for  $\Psi$  and we estimate it via Monte-Carlo sampling. We start with a straightforward calculation:

$$\begin{aligned} \Psi(T) &= \sum_{\theta, \mathbf{d}} \mathbb{P}(\theta, \mathbf{d}|T) \log \frac{\mathbb{P}(\theta, \mathbf{d}|T)}{\mathbb{P}(\theta)\mathbb{P}(\mathbf{d}|T)} \\ &= \sum_{\theta, \mathbf{d}} \mathbb{P}(\theta, \mathbf{d}|T) \log \frac{\mathbb{P}(\mathbf{d}|\theta, T)\mathbb{P}(\theta)}{\mathbb{P}(\theta)\mathbb{P}(\mathbf{d}|T)} \\ &= \sum_{\theta, \mathbf{d}} \mathbb{P}(\theta, \mathbf{d}|T) \left( \log \mathbb{P}(\mathbf{d}|\theta, T) - \log \mathbb{P}(\mathbf{d}|T) \right). \end{aligned} \quad (7)$$

Estimating the last sum requires three steps [29, 19]:

1. Sample  $\mathbb{P}(\theta, \mathbf{d}|T)$ .
2. Evaluate  $\log \mathbb{P}(\mathbf{d}|\theta, T)$  and estimate  $\log \mathbb{P}(\mathbf{d}|T)$  for each sample.

### 3. Average.

We carry out the first step by sampling the prior  $L$  times:  $\eta_k \sim \mathbb{P}(\theta), k = 1, \dots, L$ . Once we obtain the prior samples  $\eta_k$ , the likelihood is sampled  $Y_k \sim \mathbb{P}(\mathbf{d}|\eta_k, T), k = 1, \dots, L$ . This procedure results in  $L$  pairs of samples from the joint distribution of states and data:  $(\eta_k, Y_k) \sim \mathbb{P}(\theta, \mathbf{d}|T)$ .

Calculating the left summand  $\log \mathbb{P}(Y_k|\eta_k, T)$  is straightforward and only requires evaluating the likelihood. The right summand satisfies:

$$\mathbb{P}(Y_k|T) = \mathbb{E}_\theta \left[ \mathbb{P}(Y_k, \theta|T) \right] = \sum_{\theta} \mathbb{P}(\theta) \mathbb{P}(Y_k|\theta, T), \quad (8)$$

and we estimate it via Monte-Carlo, taking advantage of existing samples:

$$\widehat{\mathbb{P}(Y_k|T)} := \frac{1}{L} \sum_{r=1}^L \mathbb{P}(Y_k|\eta_r, T). \quad (9)$$

The third step is realized by first utilizing the samples  $\eta_k, Y_k$  and (9) to define:

$$\begin{aligned} \widehat{\Psi}(T) &:= \frac{1}{L} \sum_{k=1}^L \left( \log \mathbb{P}(Y_k|\eta_k, T) - \log \widehat{\mathbb{P}(Y_k|T)} \right) \\ &= \frac{1}{L} \sum_{k=1}^L \left( \log \mathbb{P}(Y_k|\eta_k, T) - \log \frac{1}{L} \sum_{r=1}^L \mathbb{P}(Y_k|\eta_r, T) \right). \end{aligned} \quad (10)$$

Calculating  $\widehat{\Psi}$  via equation (10) constitutes one of the main computational difficulties in finding an optimal design. The logic is that the number of likelihood evaluations is  $L^2$ , so calculating  $\widehat{\Psi}$  is  $\mathcal{O}(L^2)$ . The estimator  $\widehat{\Psi}$  is biased and its bias is  $\mathcal{O}(L^{-1})$ . See [29, 19] for a full discussion of convergence and bias of  $\widehat{\Psi}$ . See Supplementary Material A for a discussion of the choice of number of samples  $L$ .

## 2.4 Posterior

Once data  $\mathbf{d}'$  for design  $T'$  have been observed, we would like to define  $\Psi$  for a new design  $T$ . The definition is a natural extension of (6), with the posterior  $\mathbb{P}(\theta|T', \mathbf{d}')$  taking the place of the prior  $\mathbb{P}(\theta)$ :

$$\Psi(T; T', \mathbf{d}') := I(\theta, \mathbf{d}|T', \mathbf{d}') = \sum_{\theta, \mathbf{d}} \mathbb{P}(\theta, \mathbf{d}|T', \mathbf{d}', T) \log \frac{\mathbb{P}(\theta, \mathbf{d}|T', \mathbf{d}', T)}{\mathbb{P}(\mathbf{d}|T', \mathbf{d}', T) \mathbb{P}(\theta|T', \mathbf{d}')}, \quad (11)$$

where  $\mathbf{d}$  is the data for  $T$ . Before data are observed and design generated, we write  $T' = \emptyset$  and  $\mathbf{d}' = \emptyset$ . Therefore,  $\mathbb{P}(\theta) = \mathbb{P}(\theta|\emptyset, \emptyset)$  and indeed  $\Psi(T) = \Psi(T; \emptyset, \emptyset)$ .

The calculation of  $\widehat{\Psi}(T; T', \mathbf{d}')$  proceeds verbatim as in Section 2.3. The only difference is that instead of sampling  $\eta_k \sim \mathbb{P}(\theta), k = 1, \dots, L$ , we sample from the posterior  $\eta_k \sim \mathbb{P}(\theta|\mathbf{d}', T'), k = 1, \dots, L$ .



Sampling the posterior is achieved by Gibbs sampling. Denote all  $\theta_j$ 's except the  $i^{\text{th}}$  by  $\theta_{-i} = \{\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_N\}$ . Gibbs sampling requires repeatedly sampling from  $\mathbb{P}(\theta_i|T', \mathbf{d}', \theta_{-i})$ , which are calculated as follows:

$$\mathbb{P}(\theta_i|T', \mathbf{d}', \theta_{-i}) = \frac{\mathbb{P}(\theta_i, \theta_{-i}|T', \mathbf{d}')}{\mathbb{P}(\theta_{-i}|T', \mathbf{d}')} = \frac{\mathbb{P}(\theta_i, \theta_{-i}|T', \mathbf{d}')}{\sum_{x \in \{0,1\}} \mathbb{P}(\theta_i = x, \theta_{-i}|T', \mathbf{d}')}, \quad (12)$$

and the normalization constant cancels out, making the calculation possible.

Naively utilizing samples from the Gibbs sampler for calculating  $\widehat{\Psi}(T; T', \mathbf{d}')$  is wasteful. Recall, from the discussion in Section 2.3, that said calculation is  $\mathcal{O}(L^2)$ , where  $L$  is the number of Monte-Carlo samples utilized. Since Gibbs sampler does not generate independent samples, naively taking  $L$  samples from the Gibbs sampler would require huge  $L$  to cover all state space for  $\theta$ , thus rendering the calculation of  $\widehat{\Psi}$  prohibitively expensive. A remedy is found in [31]: “The number of ‘effectively independent samples’ in a run of length  $n$  is roughly  $n/(2\tau_{int,f})$ ”, where  $\tau_{int,f}$  is the integrated autocorrelation time for function  $f$ . Thus, we first estimate  $\tau_{int,f_i}$  for the coordinate projections  $f_i(\theta) = \theta_i$  and take  $\tau := \max_i \tau_{int,f_i}$ . The calculation of  $\tau_{int,f_i}$  is carried out using emcee’s [16] method `autocorr` during the chain’s burn-in time. We then run the Gibbs sampler for  $\tau L$  steps and discard all but every  $\tau^{\text{th}}$  sample, thus keeping computational costs and variance for  $\widehat{\Psi}$  low. Pseudocode for our Gibbs sampler can be found in Supplementary Material A.

## 2.5 Optimization

Given a routine that calculates  $\widehat{\Psi}(T; T', \mathbf{d}')$  for any design  $T$ , we need to find a way to maximize  $\widehat{\Psi}$  over all valid designs. Designs are restricted to have a fixed number of pools, denoted  $K$ . Optimizing over all valid designs results in a difficult discrete-optimization problem, which we solve via a heuristic hill-climbing approach. Although hill-climbing is a heuristic, we have found it to work sufficiently well. In each step, we take the current best design and randomly perturb it several times. We then keep the design with maximal  $\widehat{\Psi}$  as the new best design and repeat. See Supplementary Material A for details and pseudocode.

## 2.6 DOPE

We now present DOPE: D-Optimal Pooling Experimental-design, summarized in Algorithm 1 below. DOPE requires two parameters: First, the number of pooled tests per step  $K$ . Second, a decision interval  $I \subset [0, 1]$ . The decision interval defines the required certainty levels to serve as a stopping criterion for DOPE. The meaning of  $\mathbb{P}(\theta_i = 1|T, \mathbf{d}) \in I$  is that the state of individual  $i$  is still uncertain, so further testing is required. DOPE stops when there is no uncertainty regarding the state of any individual (read:  $\forall i, \mathbb{P}(\theta_i = 1|T, \mathbf{d}) \notin I$ ).

DOPE typically proceeds to find  $K$  optimal pools, perform the corresponding RT-PCR tests, and repeat the process if any individual’s posterior infection probability is in  $I$ . However, DOPE can also be executed in a nonsequential manner, where no retesting is allowed. Such a nonsequential implementation can be achieved in Algorithm 1 by letting  $K$  be the total number of allotted tests and  $I = \emptyset$ .

---

**Algorithm 1** DOPE: D-Optimal Pooling Experimental design.

---

```

1: procedure DOPE( $K, I$ )
2:    $T, \mathbf{d} \leftarrow \emptyset, \emptyset$ 
3:   repeat
4:      $T' \leftarrow \text{OPTIMALDESIGN}(K, T, \mathbf{d})$ 
5:      $\mathbf{d}' \leftarrow \text{PCR}(T')$  ▷ Perform RT-PCR tests for  $T'$ 
6:      $T \leftarrow (T_1, \dots, T_{|T|}, T'_1, \dots, T'_{|T'|})$  ▷ Concatenate
7:      $\mathbf{d} \leftarrow (\mathbf{d}_1, \dots, \mathbf{d}_{|T|}, \mathbf{d}'_1, \dots, \mathbf{d}'_{|T'|})$  ▷ Concatenate
8:   until  $\forall i \mathbb{P}(\theta_i = 1 | T, \mathbf{d}) \notin I$ 
9:   return  $\begin{bmatrix} \mathbb{P}(\theta_1 = 1 | T, \mathbf{d}) > 0.5 \\ \vdots \\ \mathbb{P}(\theta_N = 1 | T, \mathbf{d}) > 0.5 \end{bmatrix}$  ▷ Classification via posterior marginals

```

---

## 2.7 Software tools

All computations were performed using Python 3 and NumPy 1.19.4 [18]. Numba 0.50.1 [23] was used to accelerate the Gibbs sampler. Integrated autocorrelation time was calculated with emcee’s [16] method `autocorr`.

## 3 Results

We compare DOPE to three prominent pooling strategies: Dorfman, recursive, and matrix pooling. We present extensive simulation results, and consider a large number of parameter choices for DOPE. We choose  $K = 1$ , so DOPE always finds a single optimal pool in each step. decision intervals’ lower and upper bounds are chosen from  $\{0.01, 0.02, \dots, 0.15\}$  and  $\{0.3, 0.35, \dots, 0.95\}$ , respectively. We take a percentage nomenclature, so (e.g.) DOPE  $\alpha, \beta$  utilizes the decision interval  $I = [\alpha/100, \beta/100]$ .

There are three performance metrics with which one can evaluate pooling strategies: false-negative rate, false-positive rate and number of tests. We plot false-negative rates against average number of tests and delegate plots of false-positive rates to Supplementary Material A (the reason is explained in Section 4). In addition, we present the average posterior entropy for each strategy. Although the posterior entropy is not a performance metric per se, we choose to present it in plots. The reason is that presenting posterior entropy shows that indeed DOPE succeeds in maximizing this well-defined statistical criterion.

We say a pooling strategy A *dominates* another strategy B for false-negative rates if A achieves lower false-positive rates than B, while utilizing a smaller (or equal) number of tests. Similarly, we say that A dominates B for posterior entropy if similar conditions apply for posterior entropy. In the results below, we show that for both false-negative rates, as well as for posterior entropy, there are decision intervals for which DOPE dominates Dorfman, recursive, and matrix pooling.

In all simulations presented in this section, we used the commonly observed test error rates  $P_{\text{fn}} = 0.2, P_{\text{fp}} = 0.01$  [21, 33, 10]. Infection prevalence in the tested population was realized

by varying population connectivity parameter  $P_p$  and  $P_s$ , which took values in  $[0.05, 0.4]$  — see Supplementary Material A for a summary of estimates of connectivity parameters, and Supplementary Material B for a table of simulations parameters. The basal prevalence in the population was always set to  $P_b = 0.01$ .

### 3.1 DOPE dominates

We compare DOPE to Dorfman, recursive, and matrix pooling. Results are shown in Figure 2. We consider a population of size  $N = 32$ , which is the maximal pool size that can be employed without interfering the RT-PCR process by sample dilution [35]. Figure 2 shows that a decision interval can be found for which DOPE dominates competing strategies for both false-negative rate, as well as for posterior entropy. For presentation purposes, choices of decision intervals are grouped according to their lower bound, with colors corresponding to the color bar on the right of Figure 2. E.g. DOPE 60,10, DOPE 70,10 and DOPE 90,10 share the same color. Results are shown for  $P_{fn} = 0.2, P_{fp} = 0.01, P_p = 0.2, P_s = 0.2, P_b = 0.01$ , which results in average disease prevalence  $\approx 7.7\%$ . Number of samples in Monte-Carlo estimation is 20000. Number of populations simulated is 123.

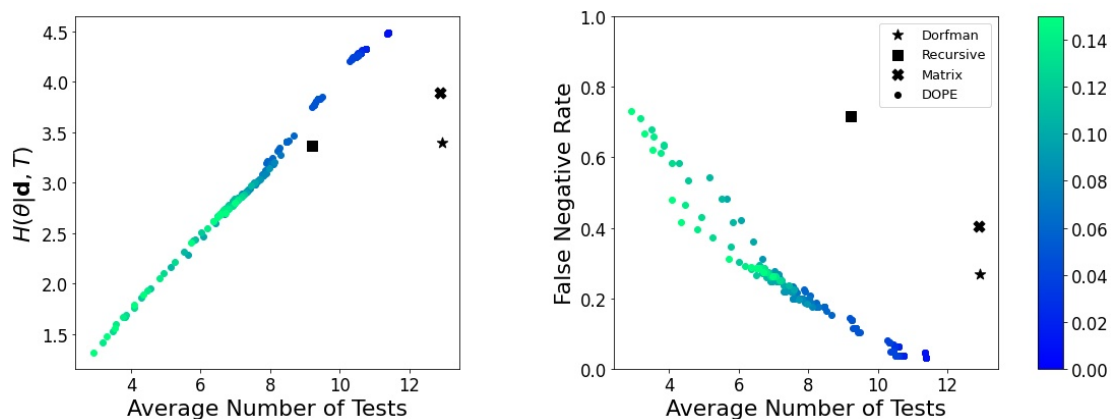


Figure 2: Comparison of DOPE against other pooling strategies. We plot the posterior entropy (left) and false-negative rates (right) achieved by different pooling strategies against the number of tests used (x-axis). The results for DOPE ( $\bullet$ ) are shown for various decision intervals (colors represent the lower bound of the decision interval). Dorfman ( $\star$ ), recursive ( $\blacksquare$ ), and matrix ( $\times$ ) pooling are shown for their only configuration, in black. We see that there are always decision intervals for which DOPE dominates competing strategies.

### 3.2 Varying prevalence

We examine the performance of DOPE under a wide range of disease prevalence rates. Performance is demonstrated for a population of size  $N = 10$  with infection prevalences in  $[0.02, 0.18]$ .

Connectivity parameters generating these prevalences can be found in Supplementary Material B. Test error rates of  $P_{fn} = 0.2, P_{fp} = 0.01$  were used, with  $L = 12000$  samples for the Monte-Carlo estimation.

In Figure 3, we show the performance of DOPE for four decision intervals. Each decision interval was chosen so that DOPE's expected number of tests was closest to one of the competing pooling strategies (Dorfman, recursive, and matrix). We also show such a comparison for separate testing. For each choice of decision interval, DOPE mostly dominates the corresponding competing strategy. The only exceptions occur when we were not able to find decision intervals that closely match the behavior of matrix pooling and separate testing. The reason we cannot find such decision intervals is that the number of tests used in matrix pooling and separate testing happens to be relatively constant in the range of disease prevalence rates we consider. DOPE is more adaptive, hence the number of tests it utilizes is increasing in the disease prevalence rate. Consequently, it is difficult to find a decision interval for which DOPE utilizes a number of tests close enough to the number of tests utilized by either separate testing or matrix pooling throughout the prevalence range considered.

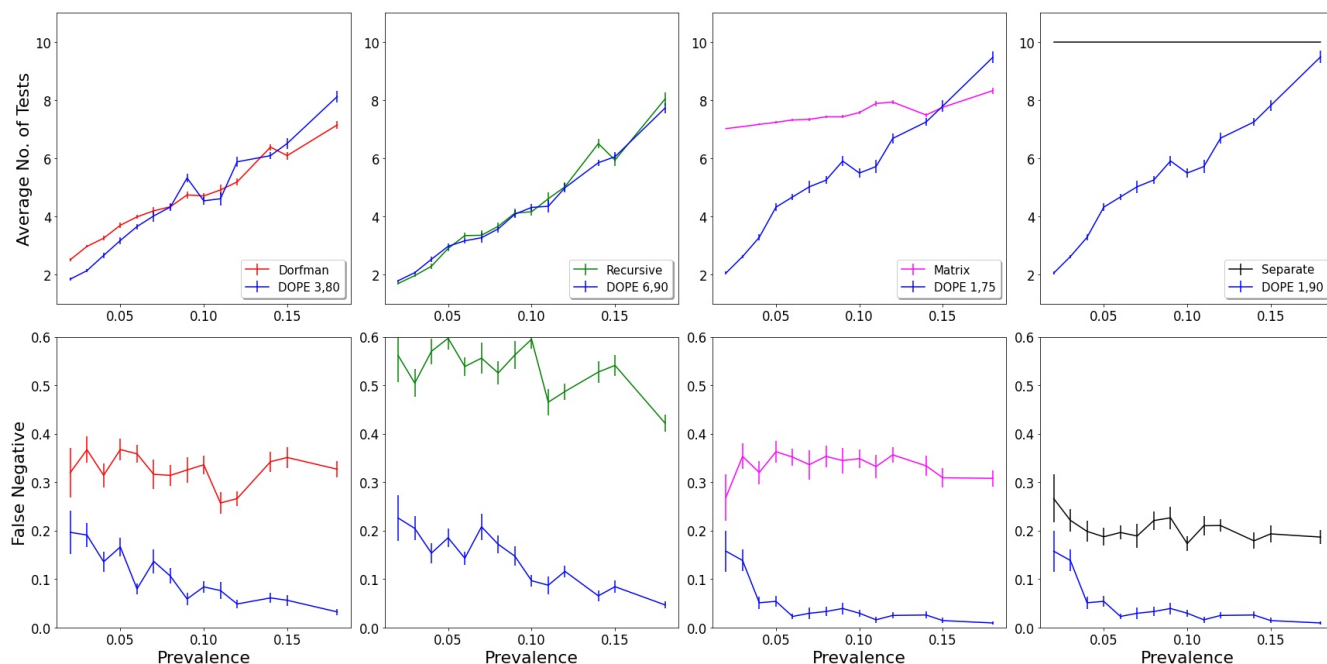


Figure 3: Comparison of DOPE to other strategies under varying disease prevalence rates. DOPE decision intervals are chosen so that the number of tests taken is as close as possible to the competing strategies. Rows, top and bottom: number of tests and false-negative rate, respectively, plotted against disease prevalence. Columns show performance for each pair of strategies (see legend in the top row). DOPE, with the chosen decision intervals, consistently dominates each competing strategy except for matrix pooling and the highest prevalence.

## 4 Discussion

In this manuscript we have presented DOPE, a novel pooling strategy that has the potential to substantially improve the performance of RT-PCR pooling in terms of number of tests and error rates. DOPE was developed with the aim of maximizing the information gained from pooled tests, precisely defined in Section 2.3. DOPE is a Bayesian method, and as such enjoys many of the advantages Bayesian analysis has to offer. For example, DOPE offers seamless integration of probabilistic assumptions on population connectivity and test errors into its underlying probabilistic model. Thus, we can apply DOPE to tests with different error models/rates (e.g. COVID-19 antigen test [34, 15]). Moreover, DOPE can make the trade-off between the number of tests and test error rates explicit. Lastly, DOPE’s error rates are lower compared to common pooling methods, while utilizing the same amount of, or fewer, tests. Last but not least, DOPE can return posterior infection probabilities, giving a very refined tool for decision making under uncertainty.

DOPE is based on an information theoretic experimental design criterion, maximizing mutual information between population infection state  $\theta$  and pooled tests results  $\mathbf{d}$ . There are multiple motivations for this definition. If we view the testing procedure as a communication channel [11], where we wish to transmit  $\theta$ , then a D-optimal design maximizes the channel capacity. The channel capacity is the upper bound for the amount of information that can be transmitted through the channel with vanishing error probability, so maximizing it is sensible. Alternatively, a quick calculation [7] verifies that  $\Psi(T) = H(\theta) - \mathbb{E}_{\mathbf{d}|T} [H(\theta|\mathbf{d}, T)]$ , where  $H(\cdot)$  is the Shannon entropy. Hence, D-optimal designs minimize the expected posterior entropy. Since entropy is a common measure for uncertainty, minimizing it is reasonable. Yet another calculation [19] shows that  $\Psi(T)$  is the expected relative entropy between the posterior and the prior. Relative entropy is a common measure of “distance” between probability distributions. Maximizing it roughly means we have learned as much as possible going from prior to posterior. Our results show that DOPE’s performance is considerably superior to competing strategies which are based on heuristics.

The Bayesian framework of DOPE also allows us to easily incorporate test error rates into our considerations. Error rates are usually not taken into account in the development of most pooling strategies, and hence such strategies are not adaptive to varying error rates [28, 8, 3, 20, 12]. The Bayesian formulation also allows DOPE to readily incorporate any prior knowledge obtained with regards to infection probabilities of different sub-populations. Although we have only considered connectivity of sub-populations in this manuscript, other covariates can potentially also be incorporated, e.g. prior data of the likelihood of infection based on symptoms, age groups, etc.

Another important advantage of DOPE is its potential to inform quarantine decisions in a fine-grained manner. This can be achieved by examining DOPE’s posterior infection probabilities  $\mathbb{P}(\theta|T, \mathbf{d})$ , instead of its binary classification. Utilizing this additional information, various quarantine policies can be implemented with respect to the policy makers’ utility functions. For example, individuals with higher posterior infection probability can be subject to a strict and prolonged quarantine and vice versa.

By selecting appropriate decision intervals, DOPE can gauge the number of tests it utilizes, giving rise to varying error rates, potentially even lower than a single test’s a priori error rate. We

find the required decision interval for given  $P_{\text{fn}}, P_{\text{fn}}, P_{\text{p}}, P_{\text{s}}, P_{\text{b}}$  and cluster sizes by first simulating DOPE for many decision intervals, e.g.  $I = [\alpha, \beta]$  for  $\alpha$  and  $\beta$  in  $\{0.01, 0.02, \dots, 0.99\}$ . We then choose  $I$  that utilizes the minimal number of tests among all decision intervals that achieved error rates lower than the desired error rate.

False-positive rates are omitted from the plots in the main text (but are found in Supplementary Material A) since these are not the main concern in an epidemiological context. A false-negative result has far worse implications than a false-positive result for the spread of an infectious disease in a susceptible population. A false-negative implies an infected individual is not identified as such and consequently can continue to spread the disease. In contrast, a false-positive only implies that a noninfected individual is unnecessarily quarantined or retested. False-positive rates are not entirely meaningless, of course, as superfluous isolation can have economic and social costs. However, the false-positive rates achieved by all strategies are still very low ( $\leq 1.5\%$ ). This is partially because RT-PCR false-positive rates are very low to begin with [10]. Thus, we believe this parameter adds very little to the comparison of competing strategies, given the vast discrepancies in the average number of tests and false-negative rates.

DOPE, as any other strategy, has some limitations. First, epidemiological data of population connectivity is not always available. In this case, one can assume the population is disconnected and use this assumption as a prior. Results for such a population are presented in Supplementary Material A. Our simulations show that even in this case, DOPE dominates competing strategies.

It is possible that the iterative steps required by DOPE (find optimal pool, retest, repeat) would be difficult to implement in a real testing scenario. In this case, a nonsequential pooling strategy, where pools are chosen a priori and no retesting is conducted can be implemented. We can take  $K$ , the number of tests per step, to equal the number of allotted tests. Then, taking the decision interval  $I = \emptyset$  makes DOPE consequential. This is a potential future research direction which was not in the scope of the current study.

Furthermore, DOPE requires substantial computational efforts, contrary to Dorfman, recursive and matrix pooling that are easily implemented. We have alleviated most computational obstacles and currently a full DOPE run, with 10 initial starting points, population size of  $N = 32$  and  $L = 20000$  samples takes less than five hours to run on seven Intel(R) Xeon(R) Gold 6252 2.1GHz CPUs. For a population size  $N = 10$ , utilizing  $L = 12000$  samples, a full DOPE run takes less than a half hour. These numbers can be considerably reduced if more CPUs are available. Further improvements to the run time of DOPE can be introduced. For example, it is possible that other approximations for  $\Psi$ , e.g. [17], could further reduce DOPE's running time. Alternatively, speeding up the optimization is also potentially possible via, e.g., solving a continuous surrogate optimization problem. To this end, we tried employing the  $\ell_0$ -sparsification method of [2], but this did not yield significant improvements in run time. Regardless, utilizing DOPE currently requires familiarity with programming. In the future we plan to create a GUI for easy use in facilities where frequent testing is performed, so that large-scale use of DOPE is possible.

Another limitation of DOPE stems from the assumptions we make. As with all models, ours does not capture reality exactly. We neglect sample dilution effects, and ignore the temporal progression of the disease [9] (see the first figure of [26] for a great illustration of this subject).

To summarize, we have shown that Bayesian experimental design holds a great potential for improving RT-PCR pooling. DOPE's potential to drastically increase test throughput and

decrease testing error rates is evident. We believe further research efforts in this direction can be very conducive to help mitigate the current pandemic, as well as future ones.

## 5 Supplementary Materials

**A:** Details of the Gibbs sampler and discrete optimization can be found in Section A.1. A sensitivity analysis for varying  $P_{fn}$ ,  $P_{fp}$  is presented in Section A.2. Simulation results for a disconnected population are presented in Section A.3. Reproducing Figure 2 and Figure 3 with false-positive data is presented in Section A.4. Estimates of number of Monte-Carlo simulations is in Section A.5. Some details on our implementation of competing strategies can be found in Section A.6. In Section A.7 we consider secondary infection probabilities, as discussed at the end of Section 2.1. Some parameter estimates for  $P_{fn}$ ,  $P_{fp}$ ,  $P_p$ ,  $P_s$  are collected in Section A.8. (pdf)

**B:** Table of parameters  $P_p$ ,  $P_s$  used to generate populations. (csv)

## References

- [1] Matthew Aldridge, Oliver Johnson, and Jonathan Scarlett, *Group testing: an information theory perspective*, arXiv preprint arXiv:1902.06002 (2019).
- [2] Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas, *A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized  $\ell_0$ -sparsification*, SIAM Journal on Scientific Computing **36** (2014), no. 5, A2122–A2148.
- [3] Hrayr Aprahamian, Douglas R Bish, and Ebru K Bish, *Optimal group testing: Structural properties and robust solutions, with application to public health screening*, INFORMS Journal on Computing (2020).
- [4] Netta Barak, Roni Ben-Ami, Tal Sido, Amir Perri, Aviad Shtoyer, Mila Rivkin, Tamar Licht, Ayelet Peretz, Judith Magenheimer, Irit Fogel, et al., *Lessons from applied large-scale pooling of 133,816 SARS-CoV-2 RT-PCR tests*, Science Translational Medicine (2021).
- [5] Roni Ben-Ami, Agnes Klochendler, Matan Seidel, Tal Sido, Ori Gurel-Gurevich, Moran Yassour, Eran Meshorer, Gil Benedek, Irit Fogel, Esther Oiknine-Djian, et al., *Large-scale implementation of pooled RNA extraction and RT-PCR for SARS-CoV-2 detection*, Clinical Microbiology and Infection **26** (2020), no. 9, 1248–1253.
- [6] Stephen A Bustin et al., *A-Z of quantitative PCR*, International University Line La Jolla, CA, 2004.
- [7] Kathryn Chaloner and Isabella Verdinelli, *Bayesian experimental design: A review*, Statistical Science **10** (1995), no. 3, 273–304.

- [8] Alhaji Cherif, Nadja Grobe, Xiaoling Wang, and Peter Kotanko, *Simulation of pool testing to identify patients with coronavirus disease 2019 under conditions of limited test availability*, JAMA network open **3** (2020), no. 6, e2013075–e2013075.
- [9] Brian Cleary, James A Hay, Brendan Blumenstiel, Maegan Harden, Michelle Cipicchio, Jon Bezney, Brooke Simonton, David Hong, Madikay Senghore, Abdul K Sesay, et al., *Using viral load and epidemic dynamics to optimize pooled testing in resource-constrained settings*, Science translational medicine (2021).
- [10] Andrew N Cohen and Bruce Kessel, *False positives in reverse transcription PCR testing for SARS-CoV-2*, medRxiv (2020).
- [11] Thomas M Cover and Joy A Thomas, *Elements of information theory*, John Wiley & Sons, 1999.
- [12] Yair Daon, Amit Huppert, and Uri Obolski, *Inflated false-negative rates in pooled RT-PCR tests of SARS-CoV-2*, medRxiv (2020).
- [13] Robert Dorfman, *The detection of defective members of large populations*, The Annals of Mathematical Statistics **14** (1943), no. 4, 436–440.
- [14] Jens Niklas Eberhardt, Nikolas Peter Breuckmann, and Christiane Sigrid Eberhardt, *Multi-stage group testing improves efficiency of large-scale COVID-19 screening*, Journal of Clinical Virology **128** (2020), 104382.
- [15] Centers for Disease Control, Prevention, et al., *Interim guidance for antigen testing for SARS-CoV-2*, Update Dec **16** (2020).
- [16] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman, *emcee: the MCMC hammer*, Publications of the Astronomical Society of the Pacific **125** (2013), no. 925, 306.
- [17] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman, *Variational bayesian optimal experimental design*, **32** (2019).
- [18] Charles R. Harris, K. Jarrod Millman, St’efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern’andez del R’io, Mark Wiebe, Pearu Peterson, Pierre G’erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant, *Array programming with NumPy*, Nature **585** (2020), no. 7825, 357–362.
- [19] Xun Huan and Youssef M Marzouk, *Simulation-based optimal Bayesian experimental design for nonlinear systems*, Journal of Computational Physics **232** (2013), no. 1, 288–317.
- [20] Hae-Young Kim, Michael G Hudgens, Jonathan M Dreyfuss, Daniel J Westreich, and Christopher D Pilcher, *Comparison of group testing algorithms for case identification in the presence of test errors*, Biometrics **63** (2007), no. 4, 1152–1163.



- [21] Lauren M Kucirka, Stephen A Lauer, Oliver Laeyendecker, Denali Boon, and Justin Lessler, *Variation in false-negative rate of reverse transcriptase polymerase chain reaction—based SARS-CoV-2 tests by time since exposure*, *Annals of Internal Medicine* (2020).
- [22] Thomas J Kwiatkowski Jr, Huda Y Zoghbi, Susan A Ledbetter, Kimberly A Ellison, and A Craig Chinault, *Rapid identification of yeast artificial chromosome clones by matrix pooling and crude lysate PCR.*, *Nucleic acids research* **18** (1990), no. 23, 7191.
- [23] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert, *Numba: A LLVM-based python JIT compiler*, *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 1–6.
- [24] Dennis V Lindley, *On a measure of the information provided by an experiment*, *The Annals of Mathematical Statistics* (1956), 986–1005.
- [25] Michael J Mina and Kristian G Andersen, *COVID-19 testing: One size does not fit all*, *Science* **371** (2021), no. 6525, 126–127.
- [26] Michael J Mina, Roy Parker, and Daniel B Larremore, *Rethinking Covid-19 test sensitivity — A strategy for containment*, *New England Journal of Medicine* **383** (2020), no. 22, e120.
- [27] Leon Mutesa, Pacifique Ndishimye, Yvan Butera, Jacob Souopgui, Annette Uwineza, Robert Rutayisire, Ella Larissa Ndoricimpaye, Emile Musoni, Nadine Rujeni, Thierry Nyatanyi, et al., *A pooled testing strategy for identifying SARS-CoV-2 at low prevalence*, *Nature* **589** (2021), no. 7841, 276–280.
- [28] Alexander Pikovski and Kajetan Bentele, *Pooling of coronavirus tests under unknown prevalence*, *Epidemiology & Infection* **148** (2020).
- [29] Kenneth J Ryan, *Estimating expected information gains for experimental designs with application to the random fatigue-limit model*, *Journal of Computational and Graphical Statistics* **12** (2003), no. 3, 585–603.
- [30] Noam Shental, Shlomia Levy, Vered Wuvshet, Shosh Skorniakov, Bar Shalem, Aner Ottolenghi, Yariv Greenshpan, Rachel Steinberg, Avishay Edri, Roni Gillis, et al., *Efficient high-throughput SARS-CoV-2 testing to detect asymptomatic carriers*, *Science advances* **6** (2020), no. 37, eabc5961.
- [31] Alan D Sokal, *Monte Carlo methods in statistical mechanics: Foundations and new algorithms*, *Functional Integration: Basics and Applications* (1996 Cargese School) (C. DeWitt Morette, P. Cartier, and A. Folacci, eds.), 1996.
- [32] L Wijsman, R Molenkamp, CBEM Reusken, A Meijer, et al., *Comparison of seven commercial RT-PCR diagnostic kits for COVID-19*, *Journal of clinical virology: the official publication of the Pan American Society for Clinical Virology* **128** (2020), 104412–104412.
- [33] Paul Wikramaratna, Robert S Paton, Mahan Ghafari, and Jose Lourenco, *Estimating false-negative detection rate of SARS-CoV-2 by RT-PCR*, *medRxiv* (2020).

- [34] World Health Organization et al., *Antigen-detection in the diagnosis of SARS-CoV-2 infection using rapid immunoassays: interim guidance, 11 September 2020*, Tech. report, World Health Organization, 2020.
- [35] Idan Yelin, Noga Aharony, Einat Shaer Tamar, Amir Argoetti, Esther Messer, Dina Berenbaum, Einat Shafran, Areen Kuzli, Nagham Gandali, Omer Shkedi, et al., *Evaluation of covid-19 rt-qpcr test in multi sample pools*, *Clinical Infectious Diseases* **71** (2020), no. 16, 2073–2078.