Noisy Adaptive Group Testing using Bayesian Sequential Experimental Design

Marco Cuturi **Olivier Teboul** **Quentin Berthet** **Arnaud Doucet** **Jean-Philippe Vert**
{cuturi,oliviert,qberthet,arnauddoucet,jpvert}@google.com

Abstract

When the infection prevalence of a disease is low, Dorfman showed 80 years ago that testing groups of people can prove more efficient than testing people individually. Our goal in this paper is to propose new group testing algorithms that can operate in a *noisy* setting (tests can be mistaken) to decide *adaptively* (looking at past results) which groups to test next, with the goal to converge to a good detection, as quickly, and with as few tests as possible. We cast this problem as a Bayesian sequential experimental design problem. Using the posterior distribution of infection status vectors for n patients, given observed tests carried out so far, we seek to form groups that have a maximal utility. We consider utilities such as mutual information, but also quantities that have a more direct relevance to testing, such as the AUC of the ROC curve of the test. Practically, the posterior distributions on $\{0,1\}^n$ are approximated by sequential Monte Carlo (SMC) samplers and the utility maximized by a greedy optimizer. Our procedures show in simulations significant improvements over both adaptive and non-adaptive baselines, and are far more efficient than individual tests when disease prevalence is low. Additionally, we show empirically that loopy belief propagation (LBP), widely regarded as the SoTA decoder to decide whether an individual is infected or not given previous tests, can be unreliable and exhibit oscillatory behavior. Our SMC decoder is more reliable, and can improve the performance of other group testing algorithms.

1 Introduction

Singling out infected individuals in a population that has little immunity to a pathogen is of paramount importance to control the propagation of an epidemic. When tests are expensive and the base infection rate is low, an approach first pioneered by Dorfman [17] consists in pooling individuals in disjoint groups (e.g. by pooling 5 nasal swabs) and test only those pooled samples first (e.g. to detect traces of virus RNA in each pool). In a second stage, only samples that belonged to positive groups are re-tested, one-by-one, to single out positives. Dorfman showed that this two-stage group testing procedure was optimal in an idealized setup, by choosing a group size that is a (decreasing) function of the disease prevalence. Dorfman's procedure is therefore well motivated mathematically, and reportedly in use to test for SARC-CoV-2 infection at scale [41, 46]. Since Dorfman's seminal work on medical testing, the field of group testing at large has significantly grown, with applications considered in quality control [43], communications [7, 45], molecular biology [6, 34], pattern matching [27, 12], database systems [13], traitor tracing [31, 24], or machine learning [47]; see [3] for a recent review.

Group testing regimes: adaptiveness and noise. Group testing strategies can be *non-adaptive*, when every group to be tested is decided beforehand, or *adaptive*, when the tests are performed in several stages, and when groups to be tested at the next stage are decided using results from all tests performed previously [37]. For example, Dorfman's strategy is adaptive and has two stages. Group testing strategies can be also be designed to handle *noisy* tests, *i.e.* account for the fact that tests can make mistakes, or, like Dorfman's, expect on the contrary that tests are *noiseless*. There exists a large body of work on adaptive and non-adaptive group testing in the noiseless setting [33, 18], where adaptive strategies tend to have better theoretical guarantees and result in more practical algorithms than non-adaptive ones [38, 2, 37]. For instance, Hwang [21] proposed a multi-stage adaptive binary splitting algorithm, which achieves the information-theoretical asymptotic lower bound on the number of tests needed to identify all infected individuals when the population size increases, and the proportion of infected individuals vanishes [5]. Additionally, it is also known, in the noiseless case, that non-adaptive designs can be suboptimal compared to adaptive strategies in some regimes [1], while optimal two-stage adaptive algorithms are also well understood [32, 15].

Noisy, adaptive group testing. With Covid-19 as a backdrop, where RT-PCR tests are known to be both in short supply and noisy [44, 46], the *noisy adaptive* setting is relevant: As noise increases, the possibly contradictory results of noisy tests can put a spoke in the wheel of combinatorial approaches. In the noisy regime, information-theoretic limits of group testing are well understood [28, 29, 4, 5, 38, 2] but most existing group testing strategies are non-adaptative [29, 10, 11, 39], with the exception of Cai et al. [8] and Scarlett [37]. These two algorithms have various optimality properties in an asymptotic regime, when the population size increases and the fraction of infected individuals vanishes. However, little is known about the quality of these methods in a non-asymptotic regime, with a finite horizon, and a small but non-vanishing proportion of infections in the population.

Our contributions. In this work, we depart from the standard asymptotic analysis, to propose a sequential Bayesian optimal experimental design (BOED) approach to group testing, consisting of:
- We derive a BOED [9] approach for group testing, in which groups are sequentially selected to maximize the *expected* utility of their hypothetical test results at the next stage. We consider two utility functions: the information gain (or mutual information) provided by a new wave of tests, or, closer to health professionals' requirements, the AUC of the ROC curve given by the marginal posterior distribution. Both are optimized using greedy forward-backward selection.
- Evaluating these utilities requires having access to an approximation of the posterior distribution of infection states of all n individuals, given all group tests observed up to that stage, with known priors on infection and noise. We use SMC samplers [16] at *each* testing stage, to approximate that posterior as a cloud of particles in $\{0,1\}^n$. SMC was used previously for other Bayesian design problems [36]; ours builds upon [40], using a Gibbs sampler as Markov chain Monte Carlo (MCMC) kernel.
- Noisy group testing approaches use a *decoder*, an algorithm tasked with outputting an infection probability vector from test results. We show that the marginal distribution produced by our SMC samplers outperform those produced by LBP, the SoTA decoder [42, 3, §3.3]. Open source code: `https://github.com/google-research/google-research/tree/master/grouptesting`

2 Background on Group Testing

Prior on infection. We consider a population of n individuals, who can be either infected or not. The infection status of the i-th individual is modeled with a binary random variable (r.v.) X_i, where $X_i = 1$ if that individual is infected and $X_i = 0$ otherwise. We write $\mathbf{X} = (X_1, \dots, X_n) \in \{0,1\}^n$ for the infection status of the whole population. We assume that a prior probability distribution for $\mathbf{X}$ is given. For example, each infection may be modelled as an independent Bernoulli r.v. $X_i \sim \mathbb{B}(q_i)$. Here, q_i is a prior infection rate, either shared across individuals, or estimated for each using other covariates. Under this model, the probability mass function (pmf) of the prior probability would satisfy, for any $\mathbf{x} = (x_1, \dots, x_n) \in \{0,1\}^n$, $\pi_0(\mathbf{x}) := \mathbb{P}_0(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^n q_i^{x_i}(1-q_i)^{1-x_i}$. More informed and non-independent priors (relating for instance two people in the same household) may be considered; as discussed later in §A.5, we approximate the prior π_0 with a weighted cloud of particles in $\{0,1\}^n$, giving us the flexibility to consider any sort of initial prior.

Group *vs.* individual testing in the presence of testing noise. Our goal is to infer which individuals are infected and which ones are not. A straightforward approach to do so would be to test each individual one-by-one. However, this raises two issues: *(i)* this requires n tests, which is costly if n is large, and inefficient if infection prevalence is low; *(ii)* tests can be noisy (e.g., nose swabs tested with RT-PCR create false negatives and, to a lesser extent, false positives), so by testing only once each individual, there is a risk of error. In this paper we solve both issues by relying on group tests. We assume that for any given group $\mathbf{g} \subset \{1, \dots, n\}$, we can pool samples from that group and test that "mixture of samples" to reveal the group's binary status: it is either *negative*, when none of the individuals in the group is infected, or *positive*, when one or more individuals are infected.

Probabilistic inference from a single group. For an integer n, we write $[\![n]\!] := \{1, \ldots, n\}$. $\mathcal{G}$ is the set of all non-empty groups, i.e. non-empty subsets of $[\![n]\!]$. With a slight overload of notations, a group $\mathbf{g} \in \mathcal{G}$ is also equivalently represented as a binary vector $\mathbf{g} \in \{0,1\}^n$, where the i-th element of $\mathbf{g}$ is 1 if and only if $i \in \mathbf{g}$. We write g for $\mathbf{g}^T \mathbf{1}_n$, the size of group $\mathbf{g}$. We write

$$\text{for all } \mathbf{g}, \mathbf{x} \in \{0,1\}^n,\ [\mathbf{g}, \mathbf{x}] := \bigvee_{i \in \mathbf{g}} x_i = 1 - \prod_{i \in \mathbf{g}} (1 - x_i) = \max(1, \mathbf{g}^T \mathbf{x}) \ \in \{0,1\}\ , \tag{1}$$

the binary status of group $\mathbf{g}$ given the binary status of individuals $\mathbf{x}$. Indeed, $[\mathbf{g}, \mathbf{x}]$ is equal to 0 if and only if all entries in $\mathbf{x}$ indexed by $\mathbf{g}$ are equal to 0. Given a group $\mathbf{g} \in \mathcal{G}$, the output of a *group test* associated to $\mathbf{g}$ is a binary r.v. $Y_\mathbf{g}$ that assesses the group status $[\mathbf{g}, \mathbf{X}]$. We assume that conditioned on $\mathbf{X}$, all considered group tests are independent from each other, and that each group test suffers from noise due to the specificity σ_g and sensitivity s_g parameters of the testing device, which both depend on the size g of $\mathbf{g}$. For any group $\mathbf{g} \in \mathcal{G}$, writing $\rho_g := s_g + \sigma_g - 1$, we have

$$\mathbb{P}(Y_\mathbf{g} = 1 \,|\, [\mathbf{g}, \mathbf{X}] = 1) = s_g\ , \quad \mathbb{P}(Y_\mathbf{g} = 0 \,|\, [\mathbf{g}, \mathbf{X}] = 0) = \sigma_g\ , \tag{2}$$

$$\forall \mathbf{x} \in \{0,1\}^n, y \in \{0,1\}, \mathbb{P}(Y_\mathbf{g} = y \,|\, \mathbf{X} = \mathbf{x}) = (\sigma_g - \rho_g[\mathbf{g}, \mathbf{x}])^{(1-y)}\ (1 - \sigma_g + \rho_g[\mathbf{g}, \mathbf{x}])^y\ . \tag{3}$$

Inference for batches of groups. We assume that up to k tests can be run simultaneously, in parallel, on a testing device. Consequently, we tailor our strategies so that they propose a *batch* of k groups $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_k) \in \mathcal{G}^k$, equivalently represented as a $n \times k$ binary membership matrix. Given a batch $\mathbf{G}$ of k groups, we define the random vector $\mathbf{Y}_\mathbf{G} := (Y_{\mathbf{g}_1}, \ldots, Y_{\mathbf{g}_k})$ of its k independent test outcomes. The probability of $\mathbf{Y}_\mathbf{G}$ taking values $\mathbf{y} \in \{0,1\}^k$ conditionally on $\mathbf{X}$ is, using (3):

$$\mathbb{P}(\mathbf{Y}_\mathbf{G} = \mathbf{y} \,|\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{k} (\sigma_{g_i} - \rho_{g_i}[\mathbf{g}_i, \mathbf{x}])^{(1-y_i)}\ (1 - \sigma_{g_i} + \rho_{g_i}[\mathbf{g}_i, \mathbf{x}])^{y_i}\ . \tag{4}$$

3 Bayesian Optimal Experimental Design to Select Useful Groups

In T-stage adaptive group design, given a finite horizon $T \in \mathbb{N}$, our goal is to select sequentially batches of groups $\mathbf{G}^t \in \mathcal{G}^k$ at each stage $1 \le t \le T$. At the end of stage $t \ge 1$, batches $\mathbf{G}^1, \ldots, \mathbf{G}^t$ were selected previously, and tested with observed outcomes $\mathbf{Y}_{\mathbf{G}^1} = \mathbf{y}^1, \ldots, \mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t$. Let us denote by $\mathbb{P}_t$ the probability conditioned to all tests seen up to stage t, i.e., for any new batch $\mathbf{G}$,

$$\begin{aligned}\mathbb{P}_t(\mathbf{X} = \mathbf{x}\,, \mathbf{Y}_\mathbf{G} = \mathbf{y}_\mathbf{G}) &:= \mathbb{P}\left(\mathbf{X} = \mathbf{x}\,, \mathbf{Y}_\mathbf{G} = \mathbf{y}_\mathbf{G} \,|\, \mathbf{Y}_{\mathbf{G}^1} = \mathbf{y}^1, \ldots, \mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t\right) \\ &= \pi_t(\mathbf{x}) \times \mathbb{P}\left(\mathbf{Y}_\mathbf{G} = \mathbf{y}_\mathbf{G} \,|\, \mathbf{X} = \mathbf{x}\right)\ ,\end{aligned}$$

where $\mathbb{P}\left(\mathbf{Y}_\mathbf{G} = \mathbf{y}_\mathbf{G} \,|\, \mathbf{X} = \mathbf{x}\right)$ is given by (4) and π_t is the posterior pmf of the vector $\mathbf{X}$ of infection states, given all those test results revealed up to stage t, i.e.:

$$\pi_t(\mathbf{x}) := \mathbb{P}_t(\mathbf{X} = \mathbf{x}) = \mathbb{P}\left(\mathbf{X} = \mathbf{x} \,|\, \mathbf{Y}_{\mathbf{G}^1} = \mathbf{y}^1, \ldots, \mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t\right)\ . \tag{5}$$

From π_t, we propose to follow a myopic approach [9] by choosing for stage $t+1$ a new batch $\mathbf{G}$ that has largest *utility* $U(\mathbf{G}, \pi_t)$ w.r.t. π_t. We introduce first a simple utility grounded on information theory, before presenting a more general and flexible formulation for utilities U in (9) below.

Maximizing mutual information. Ideally, a batch of k tests to be tested at time $t+1$ should be such that $\mathbf{Y}_\mathbf{G}$ reveals as much information as possible on $\mathbf{X}$, at time t. Since $(\mathbf{X}, \mathbf{Y}_\mathbf{G})$ are both r.v. under $\mathbb{P}_t$, this can be achieved by maximizing their mutual information (MI) utility in $\mathbf{G}$ [14]:

$$U_{\text{MI}}(\mathbf{G}, \pi_t) := I_{\mathbb{P}_t}(\mathbf{X}; \mathbf{Y}_\mathbf{G}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}|\mathbf{X}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - \sum_{\mathbf{x}} \pi_t(\mathbf{x}) H_\mathbb{P}(\mathbf{Y}_\mathbf{G}|\mathbf{X} = \mathbf{x})\,,$$

where, for any r.v. $\mathbf{Z}$ with distribution $\mathbb{P}_\mathbf{Z}$ and pmf $\eta(\mathbf{z})$, $H_{\mathbb{P}_\mathbf{Z}}(\mathbf{Z}) = -\mathbb{E}_{\mathbb{P}_\mathbf{Z}}[\log \eta(\mathbf{Z})]$ is the entropy. The MI is a standard utility function in Bayesian experimental design [25, 9, 19]. In our particular setting, U_{MI} can be evaluated thanks to this lemma (proof in §A.1):

Lemma 1. *For a group* $\mathbf{g}$*, define* $f_{\pi_t}(\mathbf{g}) := \sum_\mathbf{x} \pi_t(\mathbf{x})\,[\mathbf{g}, \mathbf{x}]$*. For* $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_k) \in \mathcal{G}^k$*, one has*

$$I_{\mathbb{P}_t}(\mathbf{X}\,;\, \mathbf{Y}_\mathbf{G}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - \sum_{i=1}^{k} \left(h_{\sigma_{g_i}} + \gamma_{g_i} f_{\pi_t}(\mathbf{g}_i)\right)\ , \tag{6}$$

where $h(u) = -u\log u - (1-u)\log(1-u)$ *is the binary entropy, and for any group size g,* $h_{\sigma_g} = h(\sigma_g)$*,* $h_{s_g} = h(s_g)$*, and* $\gamma_g = h_{s_g} - h_{\sigma_g}$*. In the case of a single group* $\mathbf{g} \in \mathcal{G}$*, this reduces to*

$$I_{\mathbb{P}_t}(\mathbf{X}; Y_{\mathbf{g}}) = h\left(\rho_g\, f_{\pi_t}(\mathbf{g}) + 1 - \sigma_g\right) - \gamma_g f_{\pi_t}(\mathbf{g}) - h_{\sigma_g}\,. \tag{7}$$

When selecting one and only one group, choosing $\mathbf{g}$ boils down to maximizing (7). The MI utility of a group is directly evaluated from $f_{\pi_t}(\mathbf{g})$, the expected value of its negative/positive status. Therefore, to maximize its MI utility, $f_{\pi_t}(\mathbf{g})$ should be close of the real argmax of $z \mapsto h(\rho_g z + 1 - \sigma_g) - \gamma_g z - h_{\sigma_g}$. If $\sigma_g = s_g = 1$, that map reduces to $h(z)$, and is maximized at 1/2. Therefore, in a noiseless setting, a group $\mathbf{g}$ is deemed useful, from a MI viewpoint, if its test r.v. $Y_{\mathbf{g}}$ is almost an unbiased coin flip.

Other utilities. Instead of relying on information theoretic quantities, we may want to handle more specific criteria. Because the posterior π_t defines what test results $\mathbf{Y_G}$ are likely to be at time t, we can define a random probability for $\mathbf{X}$, by conditioning on a test outcome $\mathbf{Y_G}$ for $\mathbf{G}$:

$$\pi_t^{\mathbf{G}}(\mathbf{x}) := \mathbb{P}_t(\mathbf{X} = \mathbf{x} | \mathbf{Y_G})\,. \tag{8}$$

$\pi_t^{\mathbf{G}}$ can be interpreted as a family of 2^k hypothetical posteriors at time $t+1$ for $\mathbf{X}$, one for each possible outcome for $\mathbf{Y_G}$. In BOED [9] a guiding principle is to score each of these hypothetical posteriors using a functional Φ, and to choose groups that maximize that score in expectation:

$$\mathbf{G}^{t+1} \in \mathrm{argmax}_{\mathbf{G}\in\mathcal{G}^k}\, U_\Phi\left(\mathbf{G}, \pi_t\right) := \mathbb{E}_{\mathbb{P}_t} \Phi(\pi_t^{\mathbf{G}})\,. \tag{9}$$

If we set Φ to be the negative entropy, choosing groups $\mathbf{G}$ that decrease maximally the expected conditional entropy of $\mathbf{X}$ at $t+1$, we recover the MI criterion (see §A.2). Although we expect lower entropy to correlate with improved testing performance, we use the flexibility of the BOED framework to optimize directly more relevant utilities. We propose to maximize the expected area under the ROC curve (AUC) of the marginal decoder: For a pmf π on $\{0,1\}^n$, the marginal decoder $m(\pi) \in [0,1]^n$ is the vector of marginal probabilities that each individual is infected under π. Given an infection status $\mathbf{x} \in \{0,1\}^n$, where $\mathrm{Pos}(\mathbf{x}) = \sum_{i=1}^n x_i$ and $\mathrm{Neg}(\mathbf{x}) = \sum_{i=1}^n (1-x_i)$ are the total number of infected and non-infected, we write $\psi_{\mathrm{AUC}}(\mathbf{s}, \mathbf{x})$ the AUC of a predictor $\mathbf{s} \in \mathbb{R}^n$:

$$\psi_{\mathrm{AUC}}(\mathbf{s}, \mathbf{x}) = \frac{\sum_{i,j=1}^n x_i(1-x_j)\left(\mathbf{1}(s_i > s_j) + \frac{1}{2}\mathbf{1}(s_i = s_j)\right)}{\mathrm{Pos}(\mathbf{x})\,\mathrm{Neg}(\mathbf{x})}\,. \tag{10}$$

where, if either Pos or Neg is 0, the AUC is discarded from our computations. The expected AUC of the marginal decoder is therefore $\Phi(\pi) = \sum_{\mathbf{x}} \pi(\mathbf{x})\psi_{\mathrm{AUC}}(m(\pi), \mathbf{x})$, which we plug directly in (9).

4 Algorithms

In order to implement the sequential BOED procedure described in §3, we now describe in more details the algorithmic components needed at each stage to *(i)* maintain a computationally tractable description of the posterior distribution (5) after each stage, *(ii)* compute the utility of a batch of groups (r.h.s. of 9), *(iii)* find a batch that solves (9), and *(iv)* compute individual infection probabilities.

Algorithm to store and update the posterior. At every stage t, we need the posterior (5) in order to evaluate and optimize utilities to select groups. This posterior is then updated by observing the results from tests carried out at stage t, before moving to stage $t+1$. One way to proceed would be to store the posterior as a 2^n-dimensional vector, keeping track of the probability of each possible population infection status vector, and update it using Bayes' rule, given test results $\mathbf{y}^t \in \{0,1\}^k$ for group $\mathbf{G}^t$:

$$\pi_t(\mathbf{x}) \propto \pi_{t-1}(\mathbf{x})\mathbb{P}(\mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t \,|\, \mathbf{X} = \mathbf{x}) \propto \pi_0(\mathbf{x}) \prod_{i=1}^t \mathbb{P}(\mathbf{Y}_{\mathbf{G}^i} = \mathbf{y}^i \,|\, \mathbf{X} = \mathbf{x})\,. \tag{11}$$

While this approach is tractable for up to $n \approx 25$ (leading to $2^{25} \approx 33$ million probabilities to store), it does not scale further. We instead use a SMC sampler [16] to approximate π_t. This provides an approximation by a cloud of weighted particles of the form $\hat{\pi}_t = \sum_{i=1}^N \omega_i \delta_{\mathbf{x}_i}$, where $N \ll 2^n$. We follow closely the algorithmic approach outlined by Schäfer and Chopin [40, Algo. 2], with two modifications: we sample initially from π_0 rather than the uniform prior, and as t grows, use $\hat{\pi}_t$ to produce $\hat{\pi}_{t+1}$; we consider a few variants for the MCMC kernel used within SMC [40, Proc. 4], including theirs, which all appear to provide similar results. We pick the modified Gibbs kernel for discrete spaces introduced by Liu [26], where we loop on the n coordinates of all particles, 4 times by default at each kernel application (see §A.5 for algorithmic details and comparisons).

Algorithms to Evaluate Utilities. Given a functional Φ, Algo. 1 shows how to estimate the utility U_Φ (9) for a group $\mathbf{G}$ and posterior pmf $\hat{\pi}_t = \sum_{i=1}^N \omega_i \delta_{\mathbf{x}_i}$ obtained at any time t through a SMC sampler. The space complexity of Algo. 1 is $O(N \times \max(2^k, n))$, where n is the number of individuals, k is the number of groups allowed per stage, and N is the size of the support of pmf $\hat{\pi}_t$, e.g. the number of particles. The time complexity is dictated by line 6, where we repeat 2^k times a call to the utility function Φ where $\hat{\pi}_t$ has a support of size N in $\{0,1\}^n$. If this operation has complexity $C(N,n)$, then the time complexity of Algo. 1 is $O(2^k C(N,n))$. For example, for utilities based on marginals such as AUC, we need to compute the marginal first in $O(Nn)$; then sort the marginal itself in $O(n\ln(n))$; then compute the AUC on each particle in $O(Nn)$ in total, resulting in $C(N,n) = O(n \max(N, \ln(n)))$. If we set Φ to be the neg-entropy, this results in $C(N,n) = O(N)$, but in that case we can use the equivalent formulation of entropy minimization as MI maximization to derive a specific $O(2^k + N)$ algorithm (instead of $O(2^k N)$ with Algo. 1), as detailed in Algo. 3.

Algorithm 1: Compute utility of a set of groups given posterior $U_\Phi(\mathbf{G}, \hat{\pi}_t)$

```
Input: $\hat{\pi}_t(\mathbf{x}) = \sum_{i=1}^N \omega_i \delta_{\mathbf{x}_i}(\mathbf{x}) = \hat{\mathbb{P}}_t(\mathbf{X} = \mathbf{x})$; $\mathbf{G} = (\mathbf{g}_1, \dots, \mathbf{g}_k) \in \{0,1\}^{n \times k}$ a set of groups;
    $\sigma, s \in [0,1]^k$ the specificities and sensitivities of the test for each group in $\mathbf{G}$;
    $\Phi : [0,1]^N \times \{0,1\}^{n \times N}$ a utility function evaluated on a weighted cloud of $N$ particles.
Output: The utility $U_\Phi(\mathbf{G}, \hat{\pi}_t)$
1 $A_{ij} \leftarrow 1 - \sigma_i + (\sigma_i + s_i - 1)[\mathbf{g}_i, \mathbf{x}_j]$ for $(i,j) \in [\![k]\!] \times [\![N]\!]$    // $\mathbb{P}(Y_{\mathbf{g}_i} = 1 \,|\, \mathbf{X} = \mathbf{x}_j)$
2 $B_{ij} \leftarrow \prod_{t=1}^k A_{tj}^{b_{it}} (1 - A_{tj})^{1 - b_{it}}$ for $(i,j) \in [\![2^k]\!] \times [\![N]\!]$, where $b_{it}$ is the $t$-th bit from the right in
    the binary expansion of $i$    // $\mathbb{P}(\mathbf{Y}_\mathbf{G} = i \,|\, \mathbf{X} = \mathbf{x}_j)$
3 $C_{ij} \leftarrow B_{ij} \times \omega_j$ for $(i,j) \in [\![2^k]\!] \times [\![N]\!]$    // $\hat{\mathbb{P}}_t(\mathbf{Y}_\mathbf{G} = i \,,\, \mathbf{X} = \mathbf{x}_j)$
4 $D_i \leftarrow \sum_{j=1}^N C_{ij}$ for $i \in [\![2^k]\!]$    // $\hat{\mathbb{P}}_t(\mathbf{Y}_\mathbf{G} = i)$
5 $E_{ij} \leftarrow C_{ij} / D_i$ for $(i,j) \in [\![2^k]\!] \times [\![N]\!]$    // $\hat{\mathbb{P}}_t(\mathbf{X} = \mathbf{x}_j \,|\, \mathbf{Y}_\mathbf{G} = i)$
6 $F_i \leftarrow \Phi(E_{i\cdot}, \mathbf{X})$ for $i \in [\![2^k]\!]$    // $\Phi(\hat{\mathbb{P}}_t(\mathbf{X} \,|\, \mathbf{Y}_\mathbf{G} = i))$
7 $U \leftarrow \sum_{i=1}^{2^k} D_i F_i$    // $\mathbb{E}_{Y_\mathbf{G}} \Phi(\hat{\pi}_t^\mathbf{G})$
8 return $U$
```

Algorithms to Maximize Utility. Taking for granted that we can evaluate $U_\Phi(\mathbf{G}, \hat{\pi}_t)$ given a candidate batch $\mathbf{G}$ and posterior pmf $\hat{\pi}_t$ using Algo. 3, the question of finding a batch $\mathbf{G}$ that maximizes U_Φ in (9) is a difficult discrete optimization problem, with costly evaluations. Any standard algorithm for discrete optimization can in principle be used to find suboptimal solutions, such as greedy forward/backward optimization, simulated annealing or genetic algorithms. We implemented a greedy approach that adds incrementally groups one by one, to build up a batch. Each group itself grows in a greedy manner, using standard forward/backward steps: we first greedily add the best F individuals one by one (forward step), and then delete the B worst ones (backward step) until we either stop improving utility, or reach the maximal group size $n_{\max}$. In the case of MI maximization, we detail a way to carry out such greedy optimization more efficiently than by applying repeatedly Algo. 3 (in the appendix).

Algorithms for Marginal Inference and Decoding. At any point in the testing campaign, one may want to compute the marginal probability for each individual to be infected, a step known as *decoding* test results. This marginal may be used to make informed decisions at any stage of the campaign, but could also be used to design new groups to test (to illustrate this, we propose in our experiments to use the informative Dorfman (ID) procedure [30] after a first round of test has been carried out, see §B). To estimate the marginal, we implemented two approaches: *(i)* marginalizing the approximate posterior $\hat{\pi}_t$ maintained by the SMC sampler (§A.5), which consists simply in computing the "mean" particle in $\hat{\pi}_t$, or *(ii)* computing the marginal with a LBP algorithm [35], as detailed in §A.4. The LBP is a fast and popular decoding algorithm in group testing [42, 3], which is not, however, guaranteed to converge to the correct marginal. On the other hand, although the SMC-based estimator may be inaccurate if the particle approximation of the posterior is poor, it does not suffer from convergence issues. We show in experiments (see Fig. 1) that SMC outperforms LBP decoding. More problematically, we find that LBP can display oscillatory behavior, notably for certain group testing strategies that re-test several times the same individuals. However, because LBP is significantly faster, we propose a mixed approach, which tests whether LBP has converged within a maximal number of iterations, and, if not, switches to a SMC. This strategy seems to be almost as effective in our simulations to using a SMC by default, and we adopt it for *all* group testing strategies.

5 Simulations

```
Algorithm 2: Simulator to evaluate the performance of a group testing Policy
Input: horizon $T \geq 1$, maximum number of tests per cycle $k$, maximum group size $n_{\max}$.
Ground truth infection probability $\mathbb{P}_0$ of $n$ patients on state space $\{0,1\}^n$.
Ground truth noise parameters $\sigma, s \in [0,1]^{n_{\max}} \times [0,1]^{n_{\max}}$ depending on group size.
Prior on infection probability $\hat{\mathbb{P}}_0$ used by algorithms.
Prior on noise parameters $\hat{\sigma}, \hat{s} \in [0,1]^{n_{\max}} \times [0,1]^{n_{\max}}$ used by algorithms.
Test$(\mathbf{G}, \sigma, s)$ returns col$(\mathbf{G})$ noisy tests using $\sigma, s$ by pooling samples according to $\mathbf{G}$.
Policy$(t, d, n_{\max}, \mathbf{y}^t, \mathbf{G}^t, \hat{\sigma}, \hat{s}, \hat{\pi}_{t-1}$ or $\bar{\mathbf{x}}_{t-1})$ calls $t$-th selector to produce up to $d$ new groups of
size at most $n_{\max}$. May use: past test results $\mathbf{y}^t, \mathbf{G}^t$; priors $\hat{\sigma}, \hat{s}$; posterior or marginal approx.
Sampler$(\mathbf{y}^t, \mathbf{G}^t, \hat{\sigma}, \hat{s}, \hat{\pi}_{t-1})$ produces $N$ approx. weighted samples from $\mathbb{P}_t$ using new tests.
MarginalSampler$(\mathbf{y}^t, \mathbf{G}^t, \hat{\sigma}, \hat{s}, \hat{\pi}_{t-1})$ produces only approx. marginal distribution.
Output: ground-truth vector, $T$ marginal predictions of infection.
1  $\mathbf{x}_{\text{truth}} \sim \mathbb{P}_0$                      // sample the ground truth status
2  $\hat{\pi}_0 \overset{N \text{i.i.d.}}{\sim} \hat{\mathbb{P}}_0$                 // sample $N$ i.i.d particles from prior
3  $\mathbf{G}^{\text{totest}} \leftarrow \mathbf{0}_{n \times 0}$                        // initialize groups
4  for $t \leftarrow 1$ to $T$ do
5      if col$(\mathbf{G}^{\text{totest}}) < k$ then                 // produce new groups if needed
6          $\mathbf{G}^{\text{add}} \leftarrow$ Policy$(t, k - \text{col}(\mathbf{G}^{\text{totest}}), \mathbf{y}^{t-1}, \mathbf{G}^{t-1}, \hat{\sigma}, \hat{s}, \hat{\pi}_{t-1}$ or $\bar{\mathbf{x}}_{t-1})$.
7          $\mathbf{G}^{\text{totest}} \leftarrow [\mathbf{G}^{\text{totest}}, \mathbf{G}^{\text{add}}]$                          // add groups to stack
8      $r \leftarrow \min(k, \text{col}(\mathbf{G}^{\text{totest}})), \mathbf{G}^t \leftarrow \mathbf{G}^{\text{totest}}_{:r}, \mathbf{G}^{\text{totest}} \leftarrow \mathbf{G}^{\text{totest}}_{r:}$       // set new tests
9      $\mathbf{y}^t \leftarrow$ Test $(\mathbf{G}^t, \sigma, s)$                    // receive lab tests
10     $\hat{\pi}_t \leftarrow$ Sampler$(\mathbf{y}^t, \mathbf{G}^t, \hat{\pi}_{t-1})$       // sample particles using test results
11     $\bar{\mathbf{x}}_t \leftarrow$ MarginalSampler$(\mathbf{y}^t, \mathbf{G}^t, \hat{\pi}_{t-1})$      // compute marginal using tests
12 return $\mathbf{x}_{\text{truth}}, (\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T)$          // ground truths + marginal predictions
```

Policies: Ours and baselines. We call a *selector* any algorithm, adaptive or not, that is able to choose groups at any stage, using possibly the knowledge of past tests. A group testing *policy* is a *sequence* of group selectors to be used at each stage. In the group testing literature, it is common that a policy sticks to a single selector throughout all stages. We propose here several new baseline policies: some use a single selector throughout, some use different selectors. For instance, we consider policies that may start with a non-adaptive selector in the first stage, followed next by an adaptive selector.

Our BOED selectors maximize, using greedy forward-3/backward-2 selection, either the mutual information (**G-MIMAX**) or the expected AUC utility (**G-AUCMAX**). We consider them as single-selector policies, and compare them to the following baseline policies. On the one hand, we consider the standard 2-stage **Dorfman** policy [17], which first splits the population in groups of size $\approx \min(n_{\max}, 1 + \lceil 1/\sqrt{q} \rceil)$, and then tests all individuals in positive groups, and the multi-stage **Binary Dorfman** policy that implements Hwang's hierarchical binary splitting approach [21] instead of the second stage of Dorfman's strategy. On the other hand, we test two non-adaptive selectors where groups are either uniformly **Random** (composed of g patients, where g is chosen so that the probability of a test being positive is close to $1/2$, which is asymptotically optimal in the absence of noise [32]), or fixed using the predefined `Origami M3` (**OM3**) assay matrix [23] containing 22 groups of maximal size 10 for 70 individuals, which was optimized to deal with an infection rate of $\approx 5\%$. We consider the **Random** selector as a policy in itself, and consider 3 mixed policies: *(i)* **Random-ID**, where a first batch of **Random** groups are formed, which is used to form a first guess for the marginal distribution, which can be used in the second stage by a variant of **Dorfmann**'s splitting known as **Informative Dorfman (ID)** [30], where the first uniform split of groups in the **Dorfman** strategy is replaced by an optimized strategy; *(ii)* **Origami-Random**, which first tests

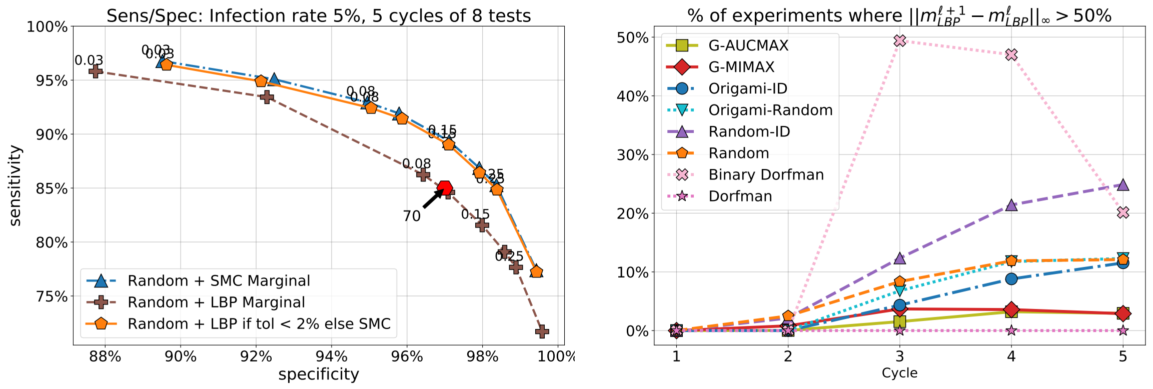


Figure 1: Issues with LBP as a decoder are highlighted in these figures. The left plot reveals that using LBP significantly degrades the performance of the **Random** policy, as measured by its average sensitivity/specificity after 40 tests. In the right plot, we count the proportion of simulations in which, at each given cycle (each cycle corresponds to $k = 8$ tests), the LBP marginal oscillates significantly, in the sense that even after $\ell = 1000$ iterations, the difference between two iterates is bigger, in at least one coordinate, by more than 0.5, which is a significant contradiction for at least one individual.

the 22 groups of **(OM3)**, and then switches to **Random** groups; (iii) **Origami-ID**, which switches instead to an **ID** strategy, using the posterior marginal computed from observing tests from **(OM3)**. These strategies are described in more detail in §B.

Group testing simulations: algorithm and parameters. Each simulation runs for a predefined T test cycles, during which we can carry out up to k tests simultaneously. We consider settings where $Tk < n$. The testing simulator is described in Alg. 2 using the following notations: $\texttt{col}(\mathbf{A})$ is the number of columns of a matrix $\mathbf{A}$; $\mathbf{A}_{:i}$, the first i columns of a matrix; $\mathbf{A}_{i:}$, the matrix $\mathbf{A}$ stripped of those i first columns. We use the following parameters in our simulations:

- population size $n = 70$; infection rate of $q = 2\%$ or 5% (see §C.1 for 10%); constant specificity $\sigma = 97\%$ and sensitivity $s = 85\%$ (see §C.5 for results with varying sensitivity);
- $k = 8$ tests per cycle, horizon of $T = 5$ cycles (total 40 tests), maximal group size: $n_{\max} = 10$.
- 5,000 simulation runs for each policy.

Decoder and discussion on LBP's convergence. To define the `MarginalSampler` referred to in Algo. 2, we considered two choices: LBP (§A.4) and the marginal of a posterior sample produced from SMC (§4,§A.5). We compare their performance in Fig. 1, using the setup of Fig. 2 & 3, to decode 40 tests generated with the **Random** policy. Fig. 1 (left) reveals that using the SMC marginal as a decoder, rather than LBP, significantly improves performances (we observed similar results for all other policies). However, because LBP is orders of magnitude faster than SMC, we propose a practical compromise, using a hybrid approach: we run LBP and check whether its iterates have stabilized after at most 1000 iterations. If the marginals produced on the two final successive iterations differ by more than 2% on any coordinate, we conclude that LBP has not stabilized and is possibly oscillating; in that case we run an SMC, and use its marginal instead. The performance of that approach is comparable to that of SMC. Notice, in Fig. 1 (right), that the number of times LBP is *significantly* unstable is far from negligible. We believe that LBP failures arise because of its inability to handle contradictory tests due to noise, notably for small groups, as can for instance happen in the Binary-Dorfman approach.

Performance in terms of sensitivity/specificity. In Fig. 2 we apply the *same* threshold on the marginal sampler's output $\bar{\mathbf{x}}_t$ (see Algo.2) at all steps, of all simulations, of all policies, to decide which individuals are classified as positive (marginal above threshold) or negative (below). We record the resulting sensitivity/specificity by comparing it to the corresponding $\mathbf{x}_{\texttt{truth}}$. We then obtain 5,000 pairs policy and at each cycle on 5,000 simulations. For those simulations with entirely negative ground-truth state vector, i.e. $\mathbf{x}_{\texttt{truth}} = \mathbf{0}$, which happens regularly when $q = 2\%$, the sensitivity cannot be evaluated, and those simulations are therefore only used to record specificity. Although

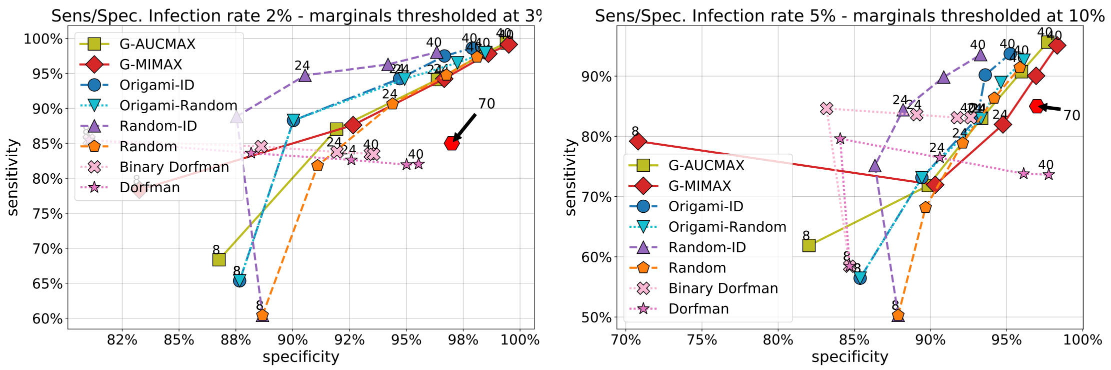


Figure 2: We provide in this plot average specificity/sensitivity for each policy, for two infection rates. For each policy, and at each step t, we use its marginal approximation $\bar{\mathbf{x}}_t$ from Algo.2 and threshold its coordinates at levels 3% and 10% respectively for 2% and 5% base infection rates, to make a binary decision. Comparing it to the corresponding $\mathbf{x}_{\text{truth}}$, we compute that simulation's specificity/sensitivity, and average them over 5000 simulations. The specificity/sensitivity of individual tests (requiring 70 tests) is plotted in red, and is significantly outperformed with our approaches.

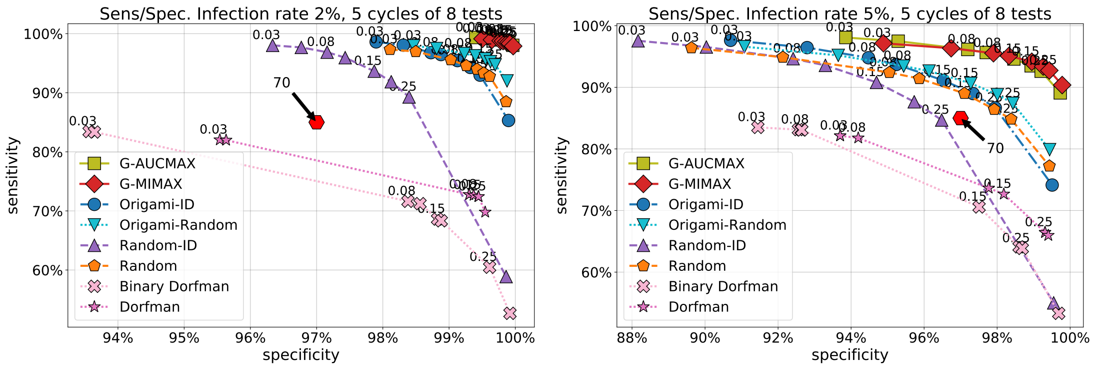


Figure 3: As in Figure 2 we report average specificity/sensitivity for various policies, but we focus on the final cycle ($t = 5$) and vary the threshold used to make decisions, which produces a global specificity/sensitivity curve for all 5,000 experiments.

we have considered previously the AUC of the marginal in an earlier version of this paper[1], we argue that computing average specificity/sensitivity for a fixed threshold results in a more realistic performance assessment: If these policies were to be deployed, one would need to "ship" them set with a threshold set beforehand. We report the dynamic progress of average specificity/sensitivity as a function of t, here labelled next to markers as total tests carried out. In Figure 3, we plot the average specificity/sensitivity of each policy, obtained this time by varying the threshold (labelled next to markers) after 5 cycles of 8 tests(*i.e.* 40 in total). This recovers a "frontier" curve of average sensitivity/specificity levels using all experiments (more plots in §C.2 at earlier cycles).

Conclusion. Our goal in this work was to maximize the efficiency of group testing in a noisy adaptive setting. We proposed a general framework to do so using Bayesian optimal sequential experimental design. By relying on a particle representation of the posterior, we formulate the problem of designing groups as a combinatorial maximization problem, solved with a greedy forward-backward approach. We have benchmarked our proposals against several baselines (some of our own design), and have shown a substantial improvement in performance. Results obtained with our G-MIMAX and G-AUCMAX approaches beat all other approaches by a wide margin. This

[1]https://arxiv.org/abs/2004.12508v1

work suggests several directions for improvement: quality of posterior sampling, alternative utility functions Φ, improvement of the combinatorial solver tasked to produce groups out of posterior samples. Since our method currently scales exponentially with the number k of requested groups (which we equate in this work with the number of tests available per cycle, for a pool of n patients), an extension of our work that carries out resampling at each group optimization iteration might be required for larger k.

References

[1] A. Agarwal, S. Jaggi, and A. Mazumdar. Novel impossibility results for group-testing. In *IEEE International Symposium on Information Theory*, pages 2579–2583, 2018.

[2] M. Aldridge. The capacity of Bernoulli nonadaptive group testing. *IEEE Trans. Inf. Theory*, 63 (11):7142–7148, 2017.

[3] M. Aldridge, O. Johnson, and J. Scarlett. Group testing: an information theory perspective. *Foundations and Trends® in Communications and Information Theory*, 15(3-4):196–392, 2019.

[4] G. K. Atia and V. Saligrama. Boolean compressed sensing and noisy group testing. *IEEE Trans. Inf. Theory*, 58(3):1880–1901, 2012.

[5] L. Baldassini, O. Johnson, and M. Aldridge. The capacity of adaptive group testing. In *IEEE International Symposium on Information Theory*, pages 2676–2680. IEEE, 2013.

[6] D. J. Balding, W. J. Bruno, D. C. Torney, and E. Knill. A comparative survey of non-adaptive pooling designs. In *Genetic Mapping and DNA Sequencing*, pages 133–154. Springer, 1996.

[7] T. Berger, N. Mehravari, D. Towsley, and J. Wolf. Random multiple-access communication and group testing. *IEEE Trans. Commun.*, 32(7):769–779, 1984.

[8] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi. Grotesque: Noisy group testing (quick and efficient). In *51st Annual Allerton Conf. on Communication, Control, and Computing*, pages 1234–1241, 2013.

[9] K. Chaloner and I. Verdinelli. Bayesian experimental design: a review. *Stat. Sci.*, 10(3):273–304, 1995.

[10] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In *49th Annual Allerton Conf. on Communication, Control, and Computing*, pages 1832–1839, 2011.

[11] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri. Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Trans. Inf. Theory*, 60(5):3019–3035, 2014.

[12] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild. Pattern matching with don't cares and few errors. *J. Comput. Syst. Sci.*, 76(2):115–124, 2010.

[13] G. Cormode and S. Muthukrishnan. What's hot and what's not: tracking most frequent items dynamically. *ACM Trans. Database Sys.*, 30(1):249–278, 2005.

[14] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, New-York, 1990.

[15] A. De Bonis, L. Gasieniec, and U. Vaccaro. Optimal two-stage algorithms for group testing problems. *SIAM J. Comput.*, 34(5):1253–1270, 2005.

[16] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. Roy. Stat. Soc. Ser. B*, 68(3):411–436, 2006.

[17] R. Dorfman. The detection of defective members of large populations. *Ann. Math. Statist.*, 14 (4):436–440, 1943.

[18] D. Du, F. K. Hwang, and F. Hwang. *Combinatorial Group Testing and Its Applications*, volume 12. World Scientific, 2000.

[19] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. In *Advances in Neural Information Processing Systems*, pages 14036–14047, 2019.

[20] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[21] F. K. Hwang. A method for detecting all defective members in a population by group testing. *J. Am. Stat. Assoc.*, 67(339):605–608, 1972.

[22] FK Hwang. A generalized binomial group testing problem. *Journal of the American Statistical Association*, 70(352):923–926, 1975.

[23] R. M. Kainkaryam and P. J. Woolf. poolhits: A shifted transversal design based pooling strategy for high-throughput drug screening. *BMC Bioinformatics*, 9(1):256, 2008.

[24] T. Laarhoven. Asymptotics of fingerprinting and group testing: Tight bounds from channel capacities. *IEEE Trans. Inf. Forensics Security*, 10(9):1967–1980, 2015.

[25] D. V. Lindley. On a measure of the information provided by an experiment. *Ann. Math. Statist.*, 27(4):986–1005, 1956.

[26] J.S. Liu. Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83(3), 1996.

[27] A. J. Macula and L. J. Popyack. A group testing method for finding patterns in data. *Discrete Appl. Math.*, 144(1-2):149–157, 2004.

[28] M. B. Malyutov. The separating property of random matrices. *Mathematical Notes of the Academy of Sciences of the USSR*, 23(1):84–91, 1978.

[29] M. B. Malyutov and P. S. Mateev. Planning of screening experiments for a nonsymmetric response function. *Mathematical notes of the Academy of Sciences of the USSR*, 27(1):57–68, 1980.

[30] C. S. McMahan, J. M. Tebbs, and C. R. Bilder. Informative Dorfman screening. *Biometrics*, 68 (1):287–296, 2012.

[31] P. Meerwald and T. Furon. Group testing meets traitor tracing. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 4204–4207, 2011.

[32] M. Mézard and C. Toninelli. Group testing with random pools: Optimal two-stage algorithms. *IEEE Trans. Inf. Theory*, 57(3):1736–1745, 2011.

[33] T. J. Mitchell and D. S. Scott. A computer program for the design of group testing experiments. *Commun. Stat. Theory Methods*, 16(10):2943–2955, 1987.

[34] H. Q. Ngo and D.-Z. Du. A survey on combinatorial group testing algorithms with applications to dna library screening. *Discrete Math. Problems with Medical Appl.*, 55:171–182, 2000.

[35] J. Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, 1982.

[36] E.G. Ryan, C.C. Drovandi, J.M. McGree, and A.N. Pettitt. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

[37] J. Scarlett. Noisy adaptive group testing: Bounds and algorithms. *IEEE Trans. Inf. Theory*, 65 (6):3646–3661, 2019.

[38] J. Scarlett and V. Cevher. Phase transitions in group testing. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 40–53. SIAM, 2016.

[39] J. Scarlett and V. Cevher. Near-optimal noisy group testing via separate decoding of items. *IEEE J. Sel. Topics Signal Process.*, 12(5):902–915, 2018.

[40] C. Schäfer and N. Chopin. Sequential Monte Carlo on large binary sampling spaces. *Stat. Comput.*, 23(2):163–184, 2013.

[41] E. Seifried and S. Ciesek. Pool-Testen von SARS-CoV-2 Proben erhöht Testkapazität. 2020. URL `https://www.medica.de/de/News/Redaktionelle_News/Pool-Testen_von_SARS-CoV-2_Proben_erhoht_Testkapazitat`.

[42] D. Sejdinovic and O. Johnson. Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 998–1003. IEEE, 2010.

[43] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell Syst. Tech. J.*, 38(5):1179–1252, 1959.

[44] P. Wikramaratna, R.S. Paton, M. Ghafari, and J. Lourenco. Estimating false-negative detection rate of SARS-CoV-2 by RT-PCR. *medRxiv*, 2020.

[45] J. Wolf. Born again group testing: Multiaccess communications. *IEEE Trans. Inf. Theory*, 31 (2):185–191, 1985.

[46] I Yelin, N Aharony, Tamar E Shaer, A Argoetti, E Messer, D Berenbaum, E Shafran, A Kuzli, N Gandali, O Shkedi, et al. Evaluation of COVID-19 RT-qPCR test in multi-sample pools. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*, 2020.

[47] Y. Zhou, U. Porwal, C. Zhang, H. Q. Ngo, X. Nguyen, C. Ré, and V. Govindaraju. Parallel feature selection inspired by group testing. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3554–3562. Curran Associates, Inc., 2014.

A Proofs and Algorithms

We provide in this section more details on the mathematics of our paper. We start in §A.1 with a proof for Lemma 1. This in turn motivates the link we make between maximizing mutual information and maximizing the expected neg-entropy of the conditional distribution of $\mathbf{X}$ given $\mathbf{Y}_\mathbf{G}$, which can be evaluated more quickly than by applying directly Algo. 3 and even maximized more efficiently using a greedy F/B algorithm as presented in §A.3. We conclude this section by providing in §A.4 the details of the message passing algorithm run to compute the LBP decoder, as well as, in §A.5, details on the SMC implementation we have considered.

A.1 Proof of Lemma 1

Let us start with a single group $\mathbf{g} \in \mathcal{G}$. We use the fact that $I_\mathbb{P}(X; Y_\mathbf{g})$ can also be written as

$$I_\mathbb{P}(\mathbf{X}; Y_\mathbf{g}) := H_\mathbb{P}(Y_\mathbf{g}) - \mathbb{E}_{\mathbb{P}_\mathbf{X}}[H_\mathbb{P}(Y_\mathbf{g}|\mathbf{X})] \,, \tag{12}$$

and compute each term in turn. $H_\mathbb{P}(Y_\mathbf{g})$ can be computed easily from the law of $\mathbf{X}$ since, by (3),

$$\mathbb{P}(Y_\mathbf{g} = 1) = \mathbb{E}_\mathbf{X}\mathbb{P}(Y_\mathbf{g} = 1 \,|\, \mathbf{X}) = \mathbb{E}_\mathbf{X}\left(1 - \sigma_g + \rho_g[\mathbf{g}, \mathbf{X}]\right) = 1 - \sigma_g + \rho_g f_{\mathbb{P}_\mathbf{X}}(\mathbf{g}) \,,$$

from which we obtain

$$H_\mathbb{P}(Y_\mathbf{g}) = h\left(\rho_g f_{\mathbb{P}_\mathbf{X}}(\mathbf{g}) + 1 - \sigma_g\right) \,. \tag{13}$$

For the second term, we notice that, conditionally to $\mathbf{X} = \mathbf{x}$, $Y_\mathbf{g}$ is a Bernoulli random variable whose expectation only depends on $[\mathbf{g}, \mathbf{x}]$, which itself can only take two values 0 and 1. By (2) we deduce:

$$H_\mathbb{P}(Y_\mathbf{g} \,|\, \mathbf{X} = \mathbf{x}) = \begin{cases} h_{s_g} & \text{if } [\mathbf{g}, \mathbf{x}] = 1 \,, \\ h_{\sigma_g} & \text{if } [\mathbf{g}, \mathbf{x}] = 0 \,, \end{cases}$$

which we can summarize as

$$H_\mathbb{P}(Y_\mathbf{g} \,|\, \mathbf{X} = \mathbf{x}) = h_{\sigma_g} + \gamma_g[\mathbf{g}, \mathbf{x}] \,. \tag{14}$$

We deduce that

$$\mathbb{E}_{\mathbb{P}_\mathbf{X}}[H(Y_\mathbf{g}|X)] = \mathbb{E}_{\mathbb{P}_\mathbf{X}}\left(h_{\sigma_g} + \gamma_g[\mathbf{g}, \mathbf{X}]\right) = h_{\sigma_g} + \gamma_g f_{\mathbb{P}_\mathbf{X}}(\mathbf{g}) \,. \tag{15}$$

Plugging (13) and (15) into (12) gives (7). Moving now to the case of a batch $\mathbf{G} = (\mathbf{g}_1, \dots, \mathbf{g}_k) \in \mathcal{G}^*$, we use the fact that the entries of $Y_\mathbf{G}$ are independent from each other given $\mathbf{X}$ to write, for any $\mathbf{x} \in \{0, 1\}^n$, and using (14),

$$H_\mathbb{P}(Y_\mathbf{G} \,|\, \mathbf{X} = \mathbf{x}) = \sum_{j=1}^{k} H_\mathbb{P}(Y_{\mathbf{g}_j} \,|\, \mathbf{X} = \mathbf{x}) = \sum_{j=1}^{k} \left(h_{\sigma_{g_j}} + \gamma_{g_j}[\mathbf{g}_j, \mathbf{x}]\right) \,.$$

As a result,

$$\mathbb{E}_{\mathbb{P}_\mathbf{X}}[H(Y_\mathbf{G} \,|\, \mathbf{X})] = \sum_{j=1}^{k} \left(h_{\sigma_{g_j}} + \gamma_{g_j} f_{\mathbb{P}_\mathbf{X}}(\mathbf{g}_j)\right) \,.$$

which gives (6).

A.2 Neg-Entropy and Mutual Information

We notice first that, using the identity that defines the mutual information in §3:

$$\begin{aligned} U_{\text{MI}}(\mathbf{G}, \pi_t) :=& I_{\mathbb{P}_t}(\mathbf{X}; \mathbf{Y}_\mathbf{G}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}|\mathbf{X}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - \sum_\mathbf{x} \pi_t(\mathbf{x}) H_\mathbb{P}(\mathbf{Y}_\mathbf{G}|\mathbf{X} = \mathbf{x}) \\ =& H_{\mathbb{P}_t}(\mathbf{X}) - H_{\mathbb{P}_t}(\mathbf{X}|\mathbf{Y}_\mathbf{G}) = H_{\mathbb{P}_t}(\mathbf{X}) - H_{\mathbb{P}_t}(\mathbf{X}|\mathbf{Y}_\mathbf{G}) \\ =& H_{\mathbb{P}_t}(\mathbf{X}) - \sum_\mathbf{y} \pi_t(\mathbf{y}) H_{\mathbb{P}_t}(\mathbf{X} = \mathbf{x}|\mathbf{Y}_\mathbf{G} = \mathbf{y}) = H_{\mathbb{P}_t}(\mathbf{X}) + \mathbb{E}_{\mathbb{P}_t}[\Phi_{\text{NegEnt}}(\pi_t^\mathbf{G})] \end{aligned}$$

where

$$\Phi_{\text{NegEnt}}(\hat{\pi}) = \sum_{i=1} \omega_i \log \omega_i \,, \text{ where } \hat{\pi} = \sum \omega_i \delta_{\mathbf{x}_i} \,,$$

is the negative entropy utility. Since $\mathbf{G}$ has no influence on $H_{\mathbb{P}_t}(\mathbf{X})$ in the r.h.s of the last line above, maximizing mutual information is equivalent to maximizing the expected neg-entropy of $\mathbf{X}$ when conditioned on hypothetical test results for group $\mathbf{G}$.

From the equality

$$I_{\mathbb{P}_t}(\mathbf{X};\mathbf{Y}_\mathbf{G}) = H_{\mathbb{P}_t}(\mathbf{Y}_\mathbf{G}) - \mathbb{E}_\mathbf{X}[H(\mathbf{Y}_\mathbf{G}|\mathbf{X})]\,,$$

we propose an algorithm that is able to directly evaluate the first term from the vector D and the second term from the matrix A, with notations from Algo.1. Since $\mathbf{Y}_\mathbf{G}$ is a product distribution conditioned to $\mathbf{X}$ we have

$$H(\mathbf{Y}_\mathbf{G}|\mathbf{X}=\mathbf{x}) = \sum_{i=1}^{k} H(Y_{\mathbf{g}_i}|\mathbf{X}=\mathbf{x}).$$

The resulting algorithm is shown in Algorithm 3. Compared to using Algorithm 1 with $\Phi = \Phi_{\text{NegEnt}}$, the computation of F in $O(N \times 2^k)$ operations to compute 2^k entropies over a space of cardinality N in Algorithm 1, line 6, is replaced by the computation of H_2 in $O(N \times k)$ (Algorithm 3, line 2) and of H_1 in $O(2^k)$ to compute a single entropy over a space of cardinality 2^k (Algorithm 3, line 7).

Algorithm 3: Compute MI utility of a set of groups

```
Input: $\hat{\pi}_t(\mathbf{x}) = \sum_{i=1}^N \omega_i \delta_{\mathbf{x}_i}(\mathbf{x}) = \hat{\mathbb{P}}_t(\mathbf{X}=\mathbf{x})$; $\mathbf{G} = (\mathbf{g}_1,\dots,\mathbf{g}_k) \in \{0,1\}^{n\times k}$ a set of groups;
    $\sigma, s \in [0,1]^k$ the specificities and sensitivities of the test for each group in $\mathbf{G}$.
Output: The MI utility of the groups $U(\mathbf{G}) = \mathrm{MI}(\mathbf{X};\mathbf{Y}_\mathbf{G})$.
1 $L_{ij} \leftarrow [g_i, \mathbf{x}_j]$ for $(i,j) \in [\![k]\!] \times [\![N]\!]$
2 $h_2 \leftarrow \sum_{i=1}^k \left[\left(\sum_{j=1}^N \omega_j L_{ij}\right)(h(s_i) - h(\sigma_i)) + h(\sigma_i)\right]$    // $\mathbb{E}_X[H(\mathbf{Y}_\mathbf{G}|\mathbf{X})]$
3 $A_{ij} \leftarrow 1 - \sigma_i + (\sigma_i + s_i - 1)L_{ij}$ for $(i,j) \in [\![k]\!] \times [\![N]\!]$    // $\mathbb{P}(Y_{\mathbf{g}_i} = 1 \,|\, \mathbf{X} = \mathbf{x}_j)$
4 $B_{ij} \leftarrow \prod_{t=1}^k A_{tj}^{b_{it}}(1 - A_{tj})^{1-b_{it}}$ for $(i,j) \in [\![2^k]\!] \times [\![N]\!]$, where $b_{it}$ is the $t$-th bit from the right in
    the binary expansion of $i$    // $\mathbb{P}(\mathbf{Y}_\mathbf{G} = i \,|\, \mathbf{X} = \mathbf{x}_j)$
5 $C_{ij} \leftarrow B_{ij} \times \omega_j$ for $(i,j) \in [\![2^k]\!] \times [\![N]\!]$    // $\mathbb{P}(\mathbf{Y}_\mathbf{G} = i\,,\, \mathbf{X} = \mathbf{x}_j)$
6 $D_i \leftarrow \sum_{j=1}^N C_{ij}$ for $i \in [\![2^k]\!]$    // $\mathbb{P}(\mathbf{Y}_\mathbf{G} = i)$
7 $h_1 \leftarrow -\sum_{i=1}^{2^k} D_j \log(D_j)$    // $H(\mathbf{Y}_\mathbf{G})$
8 return $h_1 - h_2$
```

A.3 Algorithm to Maximize MI

In this section we describe an algorithm to maximize the mutual information utility, subject to the constraint that each group should have at most $n_{\max}$ individuals, and that the batch should contain any number $m \le k$ of groups. Simply put, we greedily create groups one by one, until we have m groups. Once we have created groups $\mathbf{G}^j = (\mathbf{g}_1,\dots,\mathbf{g}_j)$, we create a new group $\mathbf{g}_{j+1}$ by starting from an empty group $\mathbf{g} = \emptyset$ (line 3) and growing iteratively the group by selecting the individual that adds the most mutual information

$$\mathbf{g} \leftarrow \mathbf{g} \cup \{i\} \text{ where } i \in \arg\max_u I_{\mathbb{P}}(\mathbf{X}\,;\, Y_{(\mathbf{G}^j, \mathbf{g}\cup\{u\})}),$$

until either we stop making progress in terms of mutual information, or when the group has already reached size $n_{\max}$. We consider additionally a variant in which we do not only consider greedy *addition* of individuals to form a group, but also *removal*, resulting in Forward-Backward iterations. Algorithm 4 describes an efficient way to carry out such forward passes more efficiently than by evaluating repeatedly Algorithm 3, because it leverages the fact that $\mathbf{G}$ is built sequentially, column by column. We omit the backward pass which only consists in changing line 5 (by setting $\mathbf{g}_\omega = 1$ instead) and removing (rather than adding) $\boldsymbol{\iota}_{u^*}$ from $\mathbf{g}$ in line 16. At each loop index in i (line 4), having a number F of forward passes and B of backward passes, with $F > B$, requires executing the body of the loop (lines 5 to 14) F times in forward mode, and B times in backward mode.

We use the following notations in Algorithm 4: small letters denote constants, small bold letters denote vectors, bold capital letters are matrices and bold greek letters are 3D tensors.

Algorithm 4: G-MIMAX: Optimize MI of m prospective group tests with greedy search

```
Input: $\hat{\pi}_t(\mathbf{x}) = \sum_{i=1}^N \omega_i \delta_{\mathbf{x}_i}(\mathbf{x}) = \hat{\mathbb{P}}_t(\mathbf{X}=\mathbf{x})$
Number of $m$ groups to add, $n_{\max}$ upperbound on group size
$\rho_i = \sigma_i + s_i - 1, \gamma_i = h_{s_i} - h_{\sigma_i},\ i \in [\![n_{\max}]\!]$.
Output: Approximate maximizer $\mathbf{G}$ of $U(\mathbf{G}) = \mathrm{MI}(\mathbf{X};\mathbf{Y}_\mathbf{G})$.
1  $\mathbf{G} \leftarrow \mathbf{0}_{n\times 0},\ \mathbf{P} \leftarrow \mathbf{1}_{n\times 1},\ h \leftarrow 0$
2  for $j \leftarrow 1$ to $m$ do
3      $\mathbf{g} \leftarrow \mathbf{0}_n,\ f_0 \leftarrow 0, \mathbf{p} = \mathbf{0}_N$   // initialize group, objective, positive in group
         across particles indicator
4      for $i \leftarrow 1$ to $n_{\max}$ do
5          $\boldsymbol{\iota} \leftarrow (w \in [\![n]\!] : \mathbf{g}_w = 0), r \leftarrow |\boldsymbol{\iota}|$   // indices that can be added
6          $T_{uv} \leftarrow \mathbf{x}_v[\boldsymbol{\iota}_u] \vee \mathbf{p}_v, (u,v) \in [\![r]\!] \times [\![N]\!]$   // detect positive in candidates
7          $\mathbf{h}^2_u \leftarrow h_{\sigma_i} + \gamma_i \sum_v T_{uv}\,\omega_v + h,\ u \in [\![r]\!]$   // conditional entropies
8          $\boldsymbol{\Gamma}_{u,v,0} \leftarrow 1 - \sigma_i + \rho T_{uv},\ \boldsymbol{\Gamma}_{u,v,1} \leftarrow \sigma_i - \rho T_{uv}, u, v),\ (u,v) \in [\![r]\!] \times [\![N]\!]$
            // probabilities of 2 possible test results, tensorized
9          $\boldsymbol{\Xi}_{u,v,b} \leftarrow \boldsymbol{\Gamma}_{u,v,0}\mathbf{P}_{v,b},\ \boldsymbol{\Pi}_{u,v,b+2^{j-1}} \leftarrow \boldsymbol{\Gamma}_{u,v,1}\mathbf{P}_{v,b},\ (u,v,b) \in [\![r]\!] \times [\![N]\!] \times [\![2^{j-1}]\!]$
            // probability tensor across all possible candidate groups $\times$
            particles $\times$ $2^j$ hypothetical test results across $j$ groups.
10         $Q_{u,b} \leftarrow \sum_v \boldsymbol{\Pi}_{u,v,b}\,\omega_v, (u,b) \in [\![r]\!] \times [\![2^{j-1}]\!]$   // marginalization / particles
11         $\mathbf{h}^1 \leftarrow -\sum_{u,b} Q_{u,b} \log(Q_{u,b})$   // unconditional entropy
12         $\mathbf{m} \leftarrow \mathbf{h}^1 - \mathbf{h}^2$   // MI objective function
13         $u^* \leftarrow \mathrm{argmax}_u\, \mathbf{m}_u, f_i \leftarrow \mathbf{m}_{u^*}, \mathbf{P}^{\text{new}}_{v,b} = \boldsymbol{\Pi}_{u^*,v,b}$   // greedy selection
14         $h^{\text{new}} \leftarrow \mathbf{h}^2_{u^*}$   // record conditional entropies of all tests so far
15         if $f_i > f_{i-1}$ then
16             $\mathbf{g} = \mathbf{g} \cup \{\boldsymbol{\iota}_{u^*}\}$   // incorporate candidate
17             $\mathbf{p} = T_{u^*,\cdot}$   // update vector of positive in group across particles
18         else
19             $\mathbf{G} = [\mathbf{G}, \mathbf{g}]$   // incorporate g
20             $\mathbf{P} = \mathbf{P}^{\text{new}},\ h = h^{\text{new}}$   // update probability & entropy after adding g
21             break
```

As an alternative to greedy approaches, we have also considered stochastic optimization approaches based on simulated annealing, with a constant temperature. These approaches can also be combined with our greedy algorithm (or any algorithm), by choosing its output as initialization, rather than a random set of groups. A simple implementation did not yield significant improvement in performance for a comparable running time.

A.4 Approximate posterior estimation by loopy belief propagation

A standard way to compute an approximation of the posterior marginals is to run loopy belief propagation (LBP) until convergence. Here we detail the LBP equations for our setting. Given n individuals and m tests performed with groups $\mathbf{g}_i, \ldots, \mathbf{g}_m \in \mathcal{G}$, LBP alternates passing messages $\mu_{i\to j} = (\mu_{i\to j}(0), \mu_{i\to j}(1)) \in \mathbb{R}^2$ from individuals $i \in [\![n]\!]$ to groups $j \in [\![m]\!]$ with $i \in \mathbf{g}_j$, and $\tilde{\mu}_{j\to i} = (\tilde{\mu}_{j\to i}(0), \tilde{\mu}_{j\to i}(1)) \in \mathbb{R}^2$ from groups j with $i \in \mathbf{g}_j$ to individuals i, respectively.

Adding a superscript (t) to clarify the messages sent at the t-th iteration of LBP, the messages from an individual $i \in [\![n]\!]$ to a group $j \in [\![m]\!]$ with $i \in \mathbf{g}_j$ follow the standard equations:

$$\begin{cases} \mu^{(t+1)}_{i\to j}(0) & = (1-q_i) \displaystyle\prod_{j'\neq j\,:\, i \in \mathbf{g}_{j'}} \tilde{\mu}^{(t)}_{j'\to i}(0)\,, \\ \mu^{(t+1)}_{i\to j}(1) & = q_i \displaystyle\prod_{j'\neq j\,:\, i \in \mathbf{g}_{j'}} \tilde{\mu}^{(t)}_{j'\to i}(1)\,. \end{cases} \tag{16}$$

The messages from a group $j \in [\![m]\!]$ to an individual $i \in [\![n]\!]$ with $i \in \mathbf{g}_j$ depend on the result of the test $Y_{\mathbf{g}_j}$: if $Y_{\mathbf{g}_j} = 0$ (negative test), then

$$\begin{cases} \tilde{\mu}^{(t)}_{j\to i}(0) & = \sigma_{\mathbf{g}_j} \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0) + (1 - s_{\mathbf{g}_j}) \left(\prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1)) - \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0) \right) \\ & = (1 - s_{\mathbf{g}_j}) \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1)) + (\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1) \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0)\,, \\ \tilde{\mu}^{(t)}_{j\to i}(1) & = (1 - s_{\mathbf{g}_j}) \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1))\,, \end{cases} \tag{17}$$

while if $Y_{\mathbf{g}_j} = 1$ (positive test), then

$$\begin{cases} \tilde{\mu}^{(t)}_{j\to i}(0) & = (1 - \sigma_{\mathbf{g}_j}) \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0) + s_{\mathbf{g}_j} \left(\prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1)) - \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0) \right) \\ & = s_{\mathbf{g}_j} \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1)) - (\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1) \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \mu^{(t)}_{i'\to j}(0)\,, \\ \tilde{\mu}^{(t)}_{j\to i}(1) & = s_{\mathbf{g}_j} \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} (\mu^{(t)}_{i'\to j}(0) + \mu^{(t)}_{i'\to j}(1))\,. \end{cases} \tag{18}$$

To simplify these equations let us introduce some notations:

$$\begin{aligned} e^{-\mu_i} &= \frac{q_i}{1 - q_i} \text{ for } i \in [\![n]\!]\,, \\ e^{\gamma^0_j} &= \frac{\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1}{1 - s_{\mathbf{g}_j}} \text{ for } j \in [\![m]\!]\,, \\ e^{\gamma^1_j} &= \frac{\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1}{s_{\mathbf{g}_j}} \text{ for } j \in [\![m]\!]\,. \end{aligned}$$

Furthermore, let us make the change of variables, for any $(i, j, t) \in [\![n]\!] \times [\![m]\!] \times \mathbb{N}$,

$$\begin{aligned} \alpha^{(t)}_{ij} &= \ln \left(\frac{\mu^{(t)}_{i\to j}(0)}{\mu^{(t)}_{i\to j}(0) + \mu^{(t)}_{i\to j}(1)} \right)\,, \\ \beta^{(t)}_{ij} &= \ln \left(\frac{\tilde{\mu}^{(t)}_{j\to i}(0)}{\tilde{\mu}^{(t)}_{j\to i}(1)} \right)\,. \end{aligned} \tag{19}$$

Then (16) can be rewritten as:

$$\begin{aligned} \alpha^{(t)}_{ij} &= -\ln \left(1 + \frac{q_i}{1 - q_i} \prod_{j' \neq j \,:\, i \in \mathbf{g}_{j'}} \frac{\tilde{\mu}^{(t)}_{j'\to i}(1)}{\tilde{\mu}^{(t)}_{j'\to i}(0)} \right) \\ &= -\ln \left(1 + e^{-\mu_i - \sum_{j' \neq j \,:\, i \in \mathbf{g}_{j'}} \beta^{(t)}_{ij'}} \right) \\ &= -\ln \left(1 + e^{-\mu_i - \bar{\beta}^{(t)}_i + \beta^{(t)}_{ij}} \right)\,, \end{aligned} \tag{20}$$

where

$$\bar{\beta}^{(t)}_i = \sum_{j \,:\, i \in \mathbf{g}_j} \beta^{(t)}_{ij}\,.$$

Similarly, denoting

$$\bar{\alpha}^{(t)}_j = \sum_{i \,:\, i \in \mathbf{g}_j} \alpha^{(t)}_{ij}\,,$$

we can rewrite (17) and (18) as follows: if $Y_{\mathbf{g}_j} = 0$,

$$\begin{aligned}\beta_{ij}^{(t)} &= \ln\left(1 + \frac{\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1}{1 - s_{\mathbf{g}_j}} \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \frac{\mu_{i' \to j}^{(t)}(0)}{\mu_{i' \to j}^{(t)}(0) + \mu_{i' \to j}^{(t)}(1)}\right) \\ &= \ln\left(1 + e^{\gamma_j^0 + \sum_{i' \neq i \,:\, i' \in \mathbf{g}_j} \alpha_{i'j}^{(t)}}\right) \\ &= \ln\left(1 + e^{\gamma_j^0 + \bar{\alpha}_j^{(t)} - \alpha_{ij}^{(t)}}, \right) ,\end{aligned} \tag{21}$$

and if $Y_{\mathbf{g}_j} = 1$,

$$\begin{aligned}\beta_{ij}^{(t)} &= \ln\left(1 - \frac{\sigma_{\mathbf{g}_j} + s_{\mathbf{g}_j} - 1}{s_{\mathbf{g}_j}} \prod_{i' \neq i \,:\, i' \in \mathbf{g}_j} \frac{\mu_{i' \to j}^{(t)}(0)}{\mu_{i' \to j}^{(t)}(0) + \mu_{i' \to j}^{(t)}(1)}\right) \\ &= \ln\left(1 - e^{\gamma_j^1 + \sum_{i' \neq i \,:\, i' \in \mathbf{g}_j} \alpha_{i'j}^{(t)}}\right) \\ &= \ln\left(1 - e^{\gamma_j^1 + \bar{\alpha}_j^{(t)} - \alpha_{ij}^{(t)}}, \right) .\end{aligned} \tag{22}$$

After convergence of the messages (denoted as $t = \infty$), we estimate the posterior marginal of the i-th individual as

$$\begin{aligned}\ln \frac{P_{\text{LBP}}(D_i = 1 \,|\, Y_{\mathbf{g}_1}, \ldots, Y_{\mathbf{g}_m})}{P_{\text{LBP}}(D_i = 0 \,|\, Y_{\mathbf{g}_1}, \ldots, Y_{\mathbf{g}_m})} &= \ln \frac{q_i}{1 - q_i} \prod_{j \,:\, i \in \mathbf{g}_j} \frac{\tilde{\mu}_{j \to i}^{(\infty)}(1)}{\tilde{\mu}_{j \to i}^{(\infty)}(0)} \\ &= -\mu_i - \sum_{j \,:\, i \in \mathbf{g}_j} \beta_{ij}^{(\infty)} ,\end{aligned} \tag{23}$$

that is,

$$P_{\text{LBP}}(D_i = 1 \,|\, Y_{\mathbf{g}_1}, \ldots, Y_{\mathbf{g}_m}) = \frac{1}{1 + e^{\mu_i + \sum_{j \,:\, i \in \mathbf{g}_j} \beta_{ij}^{(\infty)}}} . \tag{24}$$

A.5 Approximate posterior estimation by sequential Monte Carlo sampler

We detail here the SMC sampler algorithm used to provide a Monte Carlo approximation

$$\hat{\pi}_t = \sum_{i=1}^{N} \omega_i^t \delta_{\mathbf{x}_i^t},$$

of π_t given an approximation $\hat{\pi}_{t-1} = \sum_{i=1}^{N} \omega_i^{t-1} \delta_{\mathbf{x}_i^{t-1}}$ of π_{t-1}. The main idea is to introduce intermediate pmfs $\pi_t^{\gamma_k}$ of the form

$$\pi_t^{\gamma}(\mathbf{x}) \propto \pi_{t-1}(\mathbf{x}) \{\mathbb{P}(\mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t \,|\, \mathbf{X} = \mathbf{x})\}^{\gamma}, \tag{25}$$

bridging smoothly π_{t-1} to π_t using a real sequence γ_k increasing from 0 to 1 so that $\pi_t^0 = \pi_{t-1}$ and $\pi_t^1 = \pi_t$. We then approximate sequentially these pmfs using a combination of importance sampling, resampling and MCMC steps [16, 40]. This method is detailed in Algorithm 5.

Practically given an approximation of $\hat{\pi}_t^{\gamma_{k-1}} = \sum_{i=1}^{N} \omega_i \delta_{\mathbf{x}_i}$ of $\pi_t^{\gamma_{k-1}}$, an importance sampling approximation of $\pi_t^{\gamma_k}$ is given by $\hat{\pi}_t^{\gamma_k} = \sum_{i=1}^{N} \omega_i' \delta_{\mathbf{x}_i}$ where

$$\omega_i' \propto \omega_i \, \{\mathbb{P}(\mathbf{Y}_{\mathbf{G}^t} = \mathbf{y}^t \,|\, \mathbf{X} = \mathbf{x}_i)\}^{\gamma_k - \gamma_{k-1}}, \quad \sum_{i=1}^{N} \omega_i' = 1, \tag{26}$$

and a proxy measuring the "quality" of this approximation is the Effective Sample Size (ESS):

$$\text{ESS} = \frac{1}{N \sum_{i=1}^{N} (\omega_i')^2} \in [1/N, 1]. \tag{27}$$

Simply put, the higher the ESS, the better the approximation. For equally weighted particles, one has ESS $= 1$. We select here γ_k such that the ESS is equal to a pre-specified value in $[1/N, 1)$ (set to 0.9 in our experiments) and, if this yields $\gamma_k > 1$, we set $\gamma_k = 1$. Practically, this is achieved using a bisection search as described in [40, Proc.2]. Once we have determined γ_k, we then compute the new importance weights using (26), and then use a resampling procedure to replicate particles with high weights and discard particles with low weights; i.e. we approximate $\hat{\pi}_t^{\gamma_k}$ by

$$\tilde{\pi}_t^{\gamma_k} = \frac{1}{N}\sum_{i=1}^{N} n_i \delta_{\mathbf{x}_i} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\mathbf{x}}_i}. \tag{28}$$

Each particle $\mathbf{x}_i$ is copied $n_i \in [\![N]\!]$ times with $\sum_{i=1}^N n_i = N$. This can be achieved by sampling N times from $\tilde{\mathbf{x}}_i \sim \hat{\pi}_t^{\gamma_k}$ so that $(n_1, ..., n_N)$ follow a multinomial distribution. However, we use here instead the systematic resampling scheme described in [40, Proc.3] which is faster to implement and enjoys better theoretical properties.

To improve the particle approximation (28) of $\pi_t^{\gamma_k}$, the particles $\tilde{\mathbf{x}}_i$ are then evolved according to a MCMC kernel of invariant pmf $\pi_t^{\gamma_k}$. The simplest scheme consists in using the Gibbs sampler [20] which cycles through the n components of $\mathbf{x} \in \{0,1\}^n$

$$\mathbb{P}_{\mathrm{G}}(\mathbf{X}' = \mathbf{x}'|\mathbf{X} = \mathbf{x}) = \prod_{j=1}^{n} \frac{\pi_t^{\gamma_k}(\mathbf{x}'_{:j-1}, x'_j, \mathbf{x}_{j+1:})}{\pi_t^{\gamma_k}(\mathbf{x}'_{:j-1}, 0, \mathbf{x}_{j+1:}) + \pi_t^{\gamma_k}(\mathbf{x}'_{:j-1}, 1, \mathbf{x}_{j+1:})}, \tag{29}$$

where $\mathbf{x}'_{:0} = \emptyset$, $\mathbf{x}'_{:j-1} = (x'_1, ..., x'_{j-1})$ for $j \geq 1$, $\mathbf{x}_{:k+1} = \emptyset$, $\mathbf{x}_{j+1:} = (x_{j+1}, ..., x_k)$ for $j < k$. We used in the paper a modified variant of that Gibbs sampler proposed in [26], i.e.

$$\mathbb{P}_{\mathrm{MG}}(\mathbf{X}' = \mathbf{x}'|\mathbf{X} = \mathbf{x}) = \prod_{j=1}^{n} \{\alpha_j(\mathbf{x}'_{:j-1}, \mathbf{x}_{j:})\delta_{\neg x_j}(x'_j) + (1 - \alpha_j(\mathbf{x}'_{:j-1}, \mathbf{x}_{j:}))\delta_{x_j}(x'_j)\}, \tag{30}$$

for

$$\alpha_j(\mathbf{x}'_{:j-1}, \mathbf{x}_{j:}) = \min\left(1, \frac{\pi_t^{\gamma_k}(\mathbf{x}'_{:j-1}, \neg x_j, \mathbf{x}_{j+1:})}{\pi_t^{\gamma_k}(\mathbf{x}'_{:j-1}, x_j, \mathbf{x}_{j+1:})}\right). \tag{31}$$

This boils down to proposing to flip sequentially each coordinate j, this flip being accepted with probability $\alpha_j(\mathbf{x}'_{:j-1}, \mathbf{x}_{j:})$.

We also tried the independent Metropolis-Hastings sampler described in [40] that uses all the particles to build a proposal on $\{0,1\}^n$. We iterate these steps - schedule calculation, importance sampling, resampling and MCMC moves - until $\gamma_k = 1$.

Algorithm 5: `Sampler`$(\mathbf{y}^t, \mathbf{G}, \hat{\pi}_{t-1})$ returns N approximate samples from $\mathbb{P}_t$ given $\mathbf{y}^t$ and $\hat{\pi}_{t-1}$.

```
Input: Approximation $\hat{\pi}_{t-1} = \sum_{i=1}^N \omega_i^{t-1}\delta_{\mathbf{x}_i^{t-1}}$ of $\pi_{t-1}$
Output: Approximation $\hat{\pi}_t = \sum_{i=1}^N \omega_i^t\delta_{\mathbf{x}_i^t}$ of $\pi_t$
1  $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}_{t-1}, \mathbf{X} \leftarrow \mathbf{X}_{t-1}$.
2  $\gamma \leftarrow$ AdaptiveSchedule$(0, \boldsymbol{\omega}, \mathbf{X}, \mathbf{y}^t)$.            // determine first $\gamma$
3  $\boldsymbol{\omega} \leftarrow$ ImportanceWeights$(\gamma, \boldsymbol{\omega}, \mathbf{X})$.         // compute importance weights
4  while $\gamma < 1$ do
5      $\widetilde{\mathbf{X}} \leftarrow$ Resample$(\boldsymbol{\omega}, \mathbf{X})$.  // discard/multiply particles with low/high weights
6      $\boldsymbol{\omega} \leftarrow \mathbf{1}_N/N$.
7      $\mathbf{X} \leftarrow$ MCMC$(\gamma, \widetilde{\mathbf{X}}, \mathbf{y}^t)$.                 // MCMC moves targeting $\pi_t^\gamma$
8      $\gamma_{\text{old}} \leftarrow \gamma$.
9      $\gamma \leftarrow$ AdaptiveSchedule$(\gamma_{\text{old}}, \boldsymbol{\omega}, \mathbf{X}, \mathbf{y}^t)$.            // determine next $\gamma$
10     $\boldsymbol{\omega} \leftarrow$ ImportanceWeights$(\gamma - \gamma_{\text{old}}, \boldsymbol{\omega}, \mathbf{X})$.     // compute importance weights
11 $\boldsymbol{\omega}_t \leftarrow \boldsymbol{\omega}, \mathbf{X}_t \leftarrow \mathbf{X}$.
12 return $\hat{\pi}_t = \sum_{i=1}^N \omega_i^t\delta_{\mathbf{x}_i^t}$
```

B Policies and group selectors

We provide in this section more details on the various baselines we have considered in §5 in the main body of the paper.

B.1 Group Selectors

We start with selectors that require no knowledge other than the base infection rate; introduce the informative Dorfman procedure that builds on marginal information; and conclude with our selectors, G-MIMAX and G-AUCMAX. All selectors are constrained by a maximal size for groups $n_{\max}$.

• **Dorfman Splitting (D).** It splits all n patients into subgroups of size $\approx \min(n_{\max}, 1 + \lceil 1/\sqrt{q} \rceil)$.
• **Split only positives (SplitPos).** The second stage of Dorfman tests, focusing exclusively on those groups that tested positive after **(D)**. **(SplitPos:0)** tests individually all samples that have appeared in a positive group; **(SplitPos:2)** uses Hwang's hierarchical binary splitting approach [1972].
• **Mézard and Toninelli (MT).** It selects randomly groups of the same size g across all n possible patients. Given a prior q, the group is chosen to get an acceptance probability of $1/2$, yielding $g = \min(n_{\max}, \log((s_g - 1/2)/\rho_g)/\log(1-q))$. [32] proves that in the absence of noise this choice is asymptotically optimal.
• **Origami fixed design (OM3).** We also consider predefined groups, as enumerated in the `Origami M3` assay matrix [23] of size 70×22, with 22 groups whose size is equal to or smaller than 10. This matrix was proposed with a deterministic decoder that operates assuming an infection rate lower than $\approx 5\%$, in a noiseless setting. We therefore expect that assay to be the most useful when $q \leq 5\%$.
• **Informative Dorfman (ID).** Given results from the first exploitable wave of tests, we plug the marginal distribution produced by a sampler in the informative Dorfman rule [30], a generalization to a noisy setting of an approach proposed by [22]. The rule proceeds by sorting patients by increasing marginal infection probability, and group them with groups that are initially large (to clear large subsets of unlikely infected patients) to small (to test individuals likelier to be infected in smaller groups). More precisely, given a sorted list of individuals with increasing infection probability $\mathbf{p} = (p_1, \ldots, p_n)$, [30] propose in their pool specific optimal Dorfman (PSOD) algorithm to group together the first c^* individuals, where c^* is defined as

$$c^* = \operatorname{argmin}_c \frac{1}{c}\left(1 + \mathbf{1}_{c>1} c \left(s + (1 - s - \sigma)\prod_{u=1}^{c}(1 - p_u)\right)\right), \tag{32}$$

remove them from the queue and proceed until all individuals are grouped. We constrain c to be smaller than $n_{\max}$. In this work, because the infection prior is uniform, we first run a first wave of tests (using either **Random** or **Origami**) and use the resulting marginal as an estimate for $\mathbf{p}$. When appropriate, we also use group size specific sensitivites and specificities in (32).
• **Greedy Maximization of Mutual Information (G-MIMAX) and AUC (G-AUCMAX).** We optimize the MI and AUC utilities using k groups as described in Algorithm 1. The greedy approach for G-MIMAX is detailed in Algo. 4, G-AUCMAX uses calls to Algo. 1 along with a greedy forward/backward solver. We use 3 Forward steps and 2 Backward steps in all our experiments. We have experiemnted with a larger number of forward / backward steps, but we find that the resulting computational overhead is not worth the small variation in performance that is obtained.

B.2 Policies

We consider the following policies, all composed by one or at most two group selectors.

• **Dorfman**: starts with Dorfman splitting first **(D)** followed by **(SplitPos:0)**
• **Binary Dorfman**: starts with Dorfman splitting first **(D)** followed repeatedly by **(SplitPos:2)**.
• **Random** : generates random groups at each stage with the **(MT)** selector.
• **Random-ID** : starts with a random group **(MT)**, follows with **(ID)**.
• **Origami-ID**: uses **(OM3)** for 22 tests, and then switches to **(ID)**.
• **Origami-Random**: uses **(OM3)** for 22 tests, and then switches to **(ID)**.
• **G-MIMAX** and **G-AUCMAX**: optimize first utilities on a sample with $N = 10^4$ particles from the prior, and then from the posterior distribution using a SMC sampler with the same size N.

All policies are decoded using the same decoded, a hybrid rule that runs an LBP, checks if it has converged (tolerance of 2% for any coordinate) and, if not, runs a SMC (see discussion in §5).

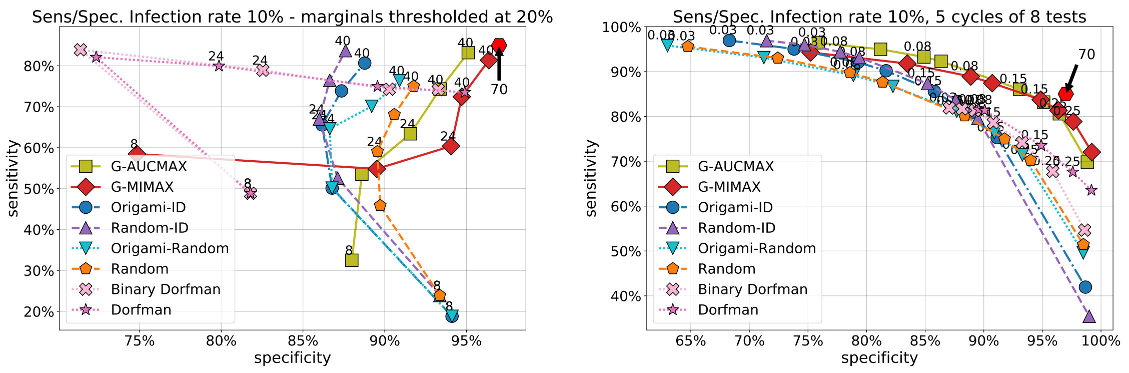


Figure 4: Using the same setup as in Fig. 3 and 2, we report results for an infection base rate of $q = 10\%$. The red hexagon depicts the sensitivity/specificity of 70 individual tests.

C Additional Experiments

We provide in this section many more results that validate further the good performance of our approaches and illustrate their robustness.

- **10% infection rate**: we list in §C.1 new plots, comparable to those already included in the main body of the paper, for a higher infection prevalence prior of $q = \hat{q} = 10\%$.
- **Dynamics of testing performance**: other plots are listed in §C.2 to provide a more detailed assessment of the performance of each policy as the number of tests that is revealed grows.
- **Robustness to Mis-specification**: we list plots in §C.3 in which we use different values for q, s (parameters used by the simulator) and $\hat{q}, \hat{s}$ (parameters used by the policies to produce groups and marginal decoders to interpret them). Obviously some policies are more sensitive to these gaps: For instance, Dorfman splitting does not consider sensitivities to form groups (but uses them to decode test results). Although one may have expected a substantial decrease in performance of our proposals G-AUCMAX and G-MIMAX, neither seems to materialize, as their performance stays clearly above that of all other baselines.
- $k = 1$ **and single test steps**: In §C.4 we report fine grained results for **G-MIMAX** and other baselines that show the evolution of the performance of this method in the most favorable setting on paper, that in which they are free to choose a new test based on the result of the previous test. While this setup would be unrealistic, because it would involve waiting hours required to output a single test result before choosing a new group, it showcases the speed at which our method reaches better performance than other baselines.
- **More challenging setup:** $n = 96$ **and varying** s_g: we end this section with a last setup that is more challenging ($n = 96$ is larger, and so one may worry about the ability of our posteriors to cover a space of size 2^96) and which also factors a degraded sensitivity as a function of group size. In that setting, we still observe a substantial gap between G-MIMAX and the other techniques. Additionally, we show that N (the number of particles) and F/B (number of forward/backward iterations) also seem to have a small impact on the performance of G-MIMAX.

C.1 $q = 10\%$ infection rates

We provide in Fig. 4 additional results for a base infection rate of 10%.

C.2 Specificity / Sensitivity curves as function of cycles

We provide in Fig. 5 additional plots that complement 3, displaying how the specificity/sensitivity frontier evolves as the number of tests grows, for each of the policies we considered. We propose a

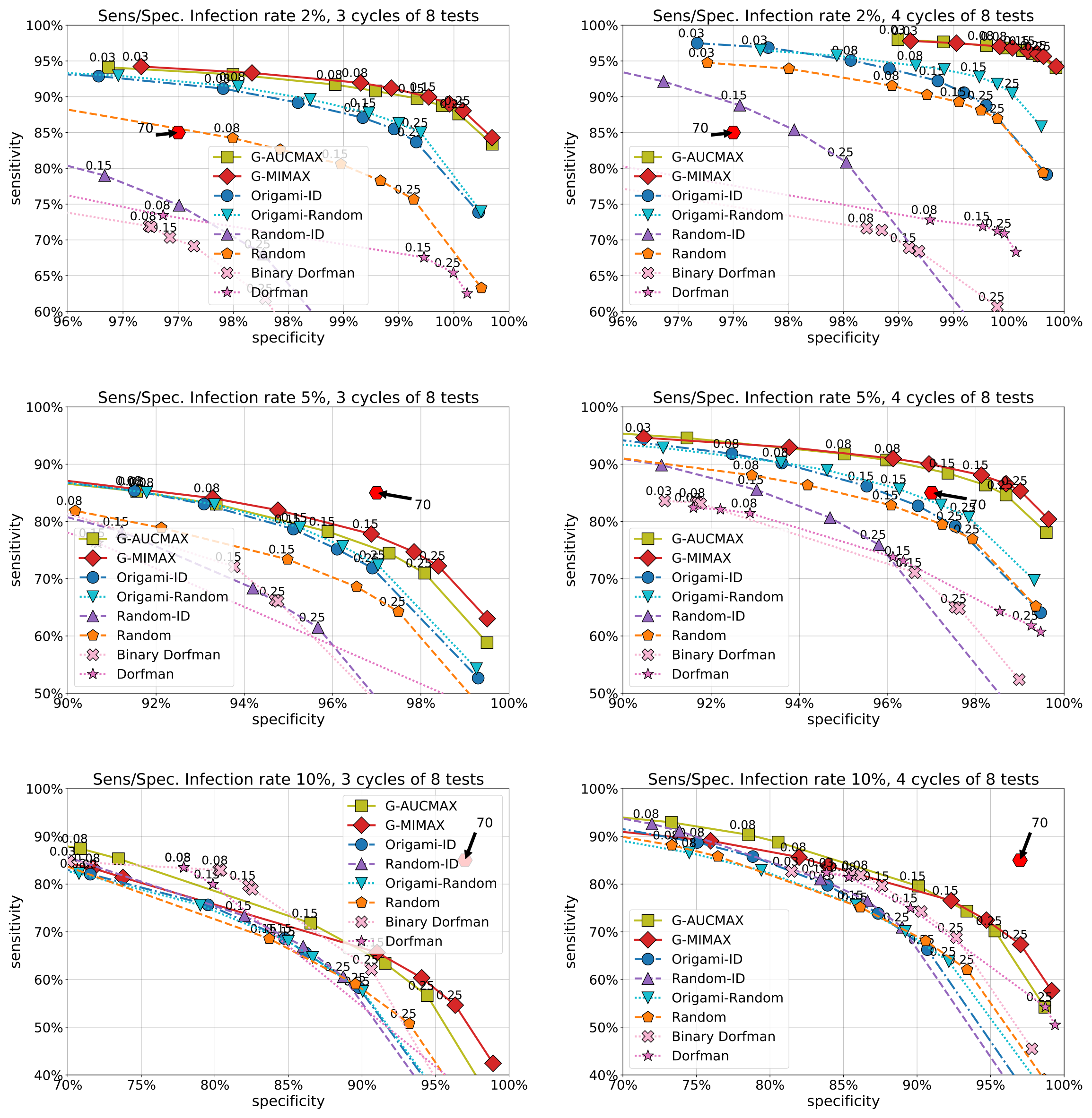


Figure 5: Using the same setup as in Fig. 3, we report results for 3 and 4 testing cycles (corresponding therefore to 24 and 32 tests carried out in total). Note that, since the Origami assay only considers 22 tests,the slight difference in performance between Origami-ID and Origami-Random that arises on the left plots is only due to 2 tests, carried out using ID or randomly.

more detailed view of that dynamic evolution for each policy taken individually for all three base infection rates, in Fig. 6, 7 and 8.

C.3 Robustness to misspecification

We study in this section robustness to misspecification of the policies we considered. Since all policies rely on a marginal decoder, their specificity / sensitivity hinges on the fact that the infection rate q and testing device's noise parameters s, σ both match with those used by the marginal decoder. We quantify how that performance varies under misspecification by considering the following perturbations: we use the setup from the right plot in Fig. 3, namely $q = 5\%, s = 85\%, \sigma = 97\%$,

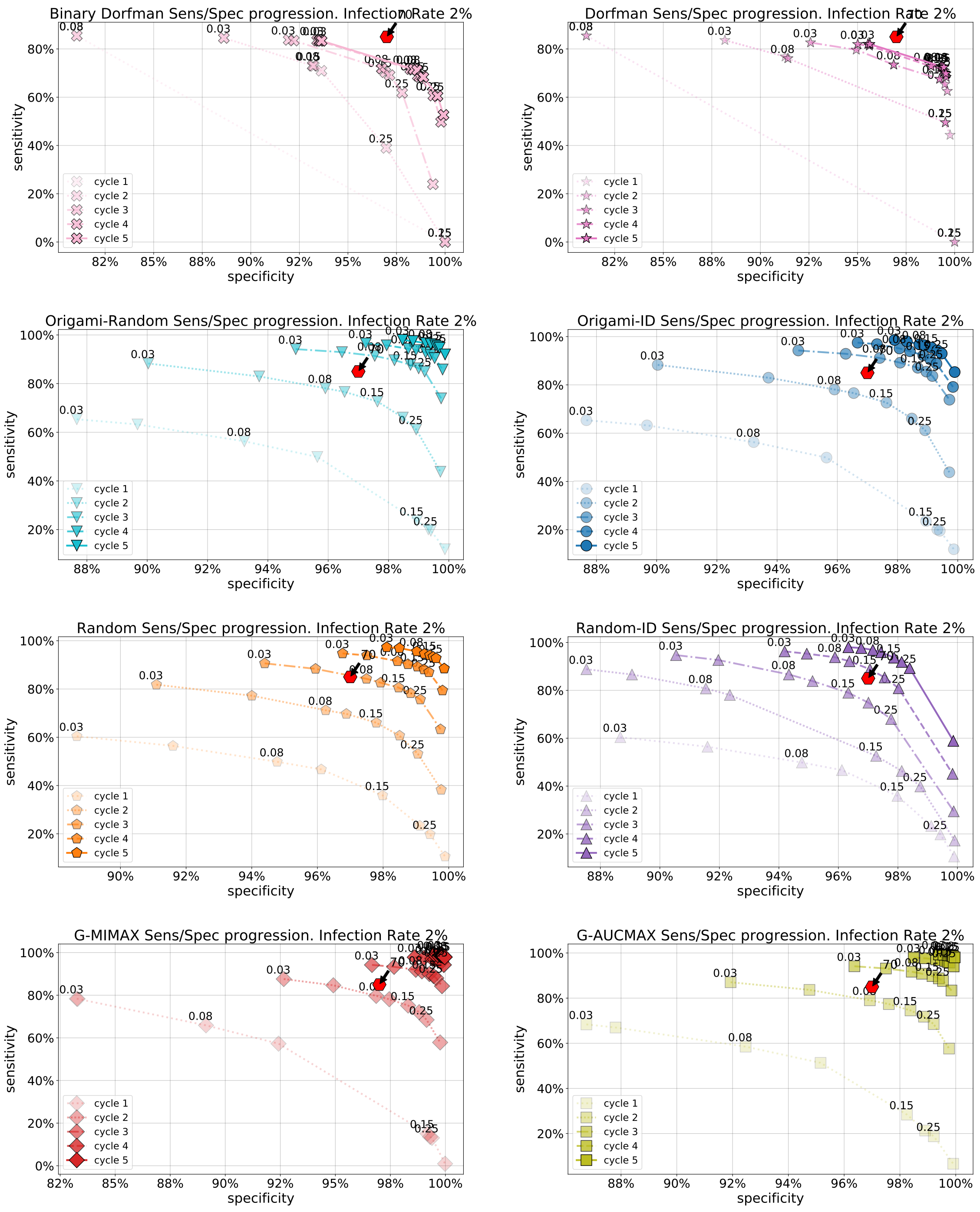


Figure 6: Using the same setup as in Fig. 3, we report dynamic results for each policy, as the number of tests increases, for $q = 2\%$. The number of tests is equal to k (here 8) times the cycle number.

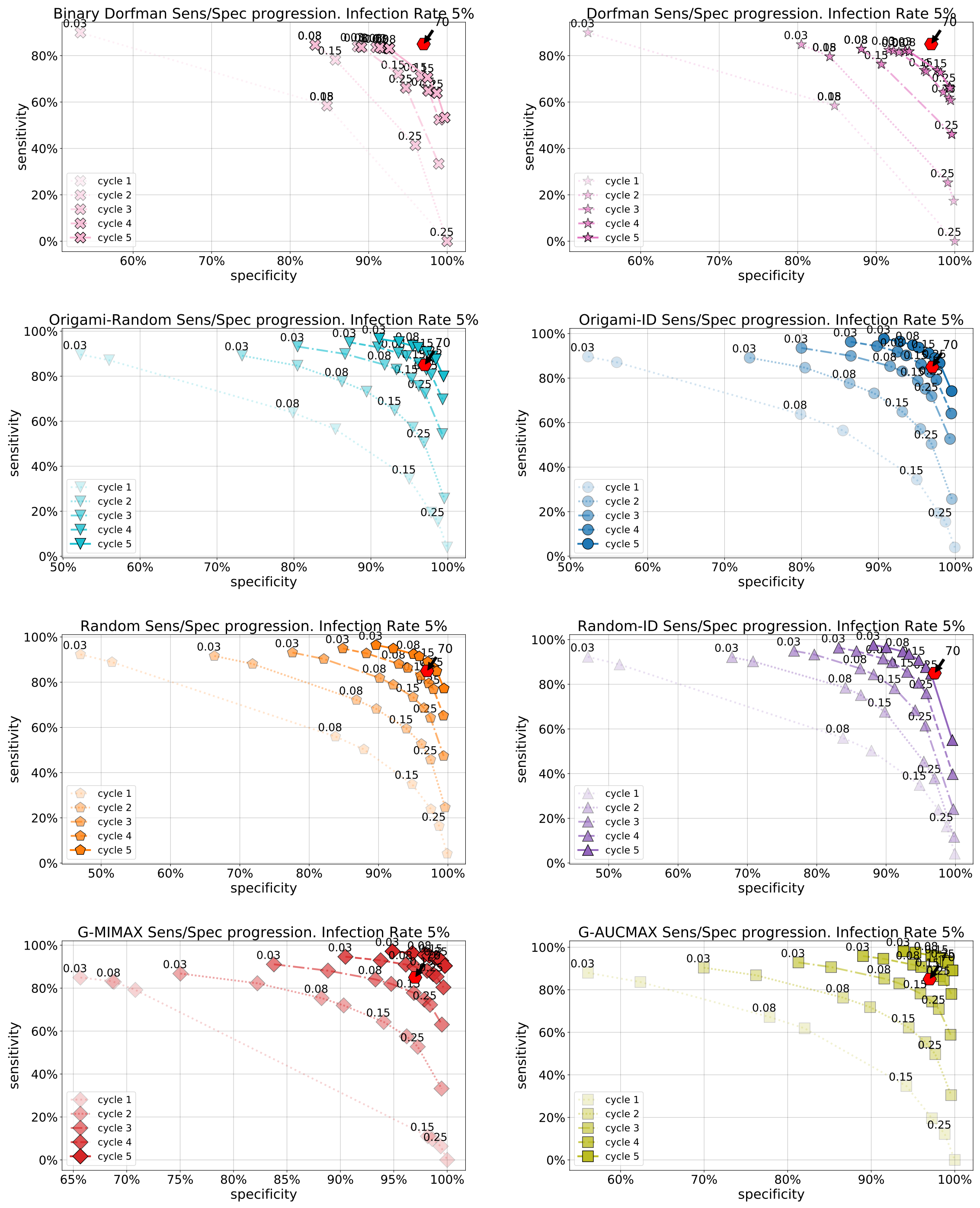


Figure 7: Using the same setup as in Fig. 3, we report dynamic results for each policy, as the number of tests increases, for $q = 5\%$. The number of tests is equal to k (here 8) times the cycle number.

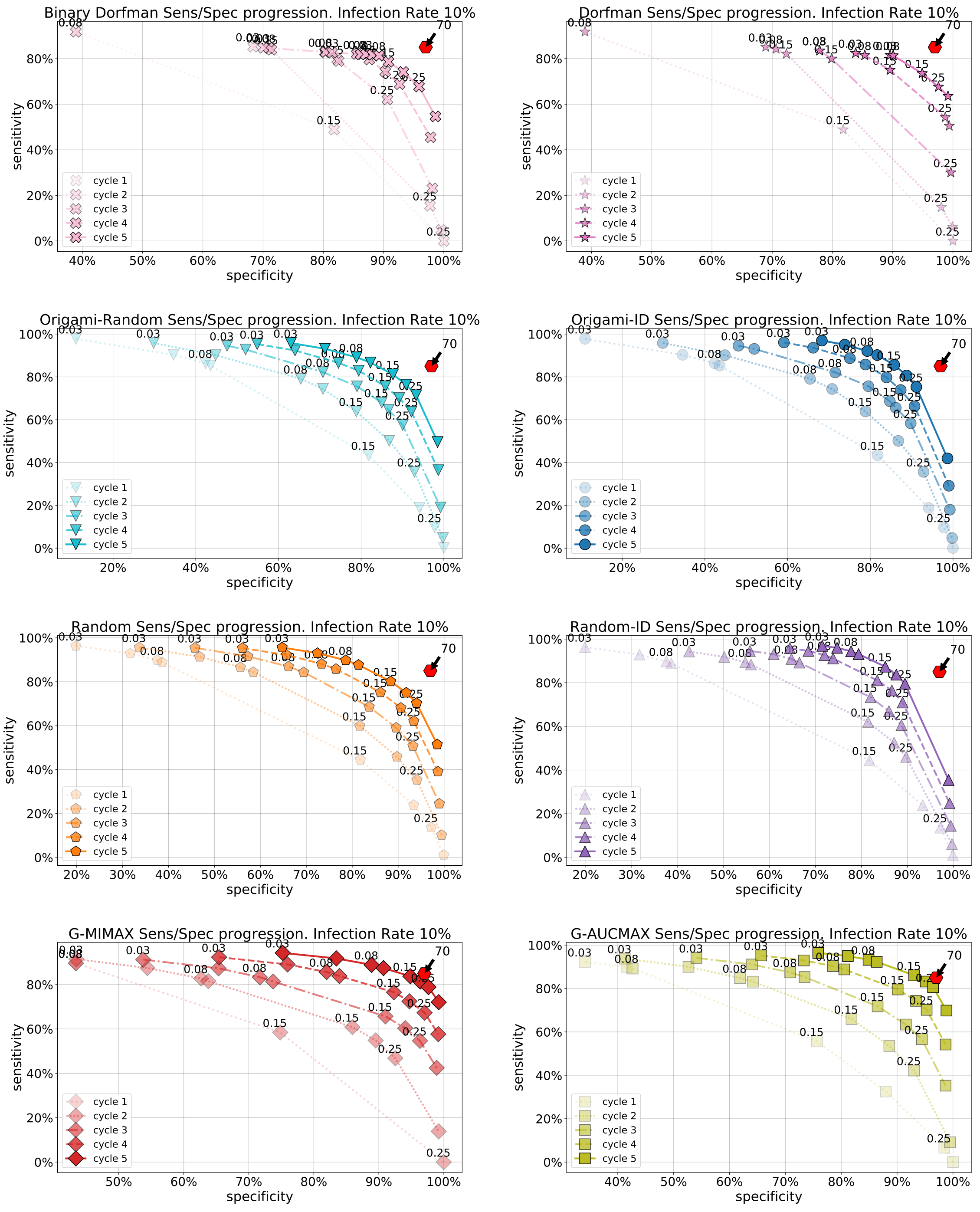


Figure 8: Using the same setup as in Fig. 3, we report dynamic results for each policy, as the number of tests increases, for $q = 10\%$. The number of tests is equal to k (here 8) times the cycle number.

to generate the ground truth in our simulations, as well as to execute tests. On the other hand, the policies (along with their decoders), will be tested under 8 additional scenarios: $\hat{q} \in \{3\%, 5\%, 8\%\}$ and $\hat{s} \in \{78\%, 85\%, 92\%\}$. Naturally, when $\hat{q} = 5\%$ and $\hat{s} = 85\%$ we fall back on the well-specified scenario. To facilitate comparison, our two proposals **(G-AUCMAX)** and **(G-MIMAX)** are displayed

C.4 Experiments with $k = 1$

We consider now in Fig. 10 a setup where $k = 1$. In that setting, we can have a fine grained picture of what each of the considered policies does when using the latest test result to produce a new group. The adaptiveness of **(G-MIMAX)** is showed case here, as we see the method maintain an acceptable specificity to highlight progressively positives while making few mistakes. This setting is particularly relevant to compare in an idealized setting our approach to the performance of Dorfman baselines.

C.5 Varying sensitivity

We explore an additional experimental setup that is a bit more ambitious in scale, since $n = 96$, $n_{\max} = 12$, $k = 10$ and $T = 4$. In that setting, we assume correct specification, with slightly different infection rates than those used before, $q \in \{2\%, 4\%, 7\%\}$ and a more reliable specificity $\sigma = 0.99$, but factor in a decreasing sensitivity as the group size increases, $s_g = (91 - g)\%$. We also consider in that setup various iterations for our greedy forward-backward approach, and $N \in \{10000, 20000\}$ total particles.

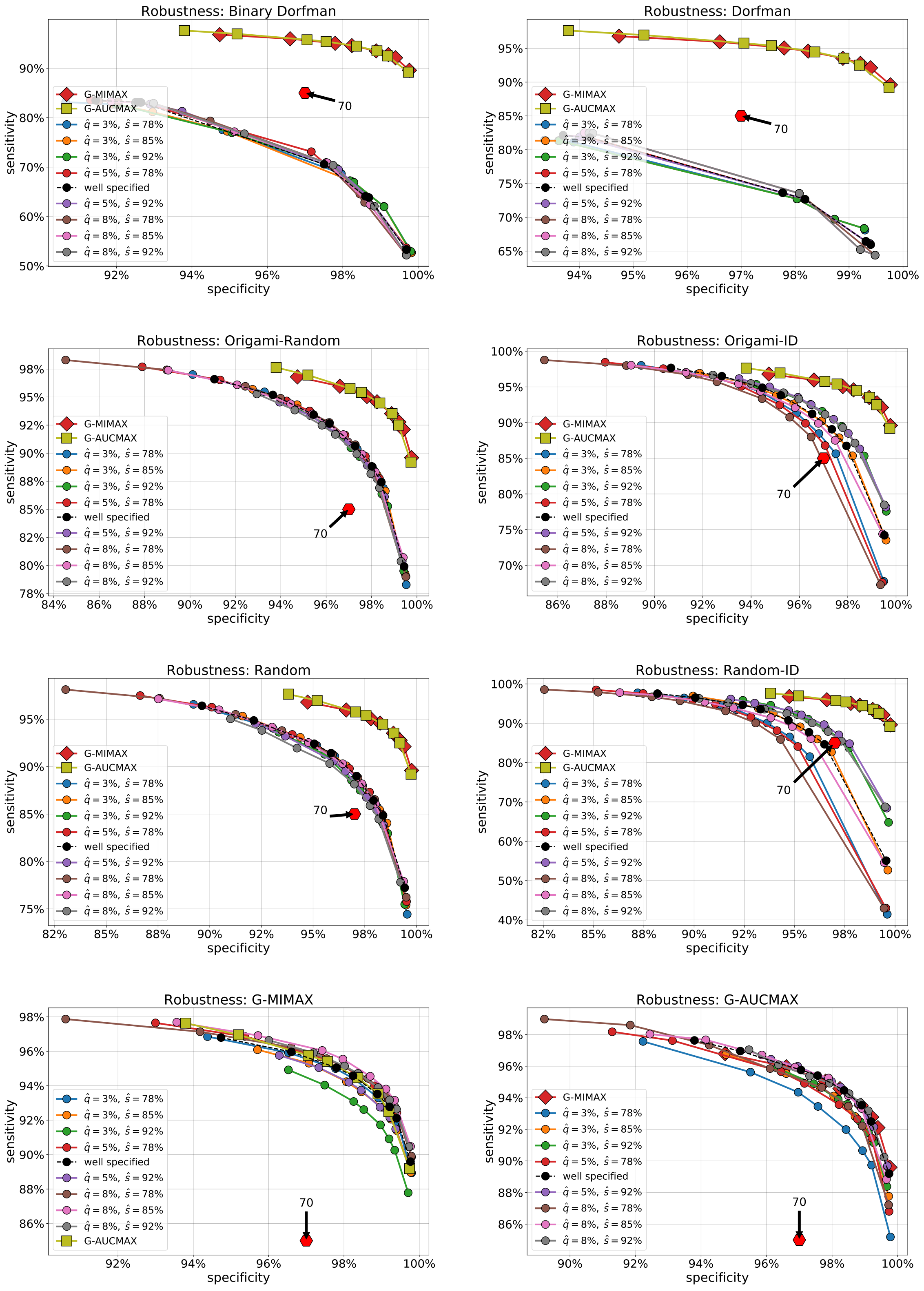
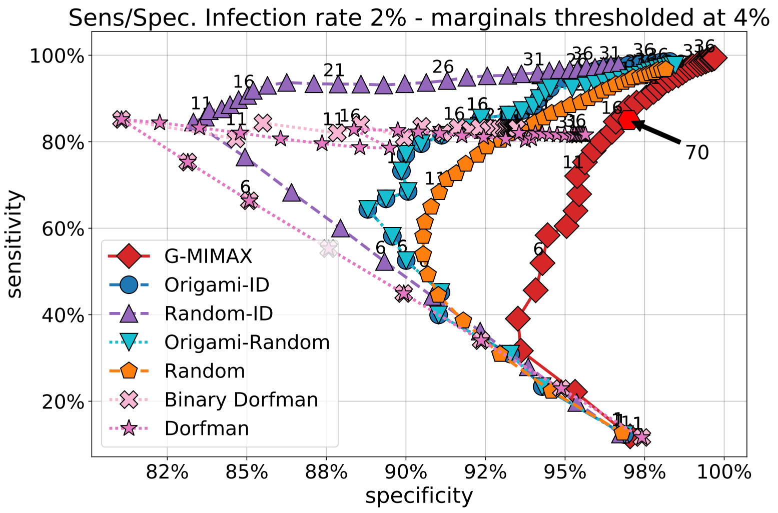
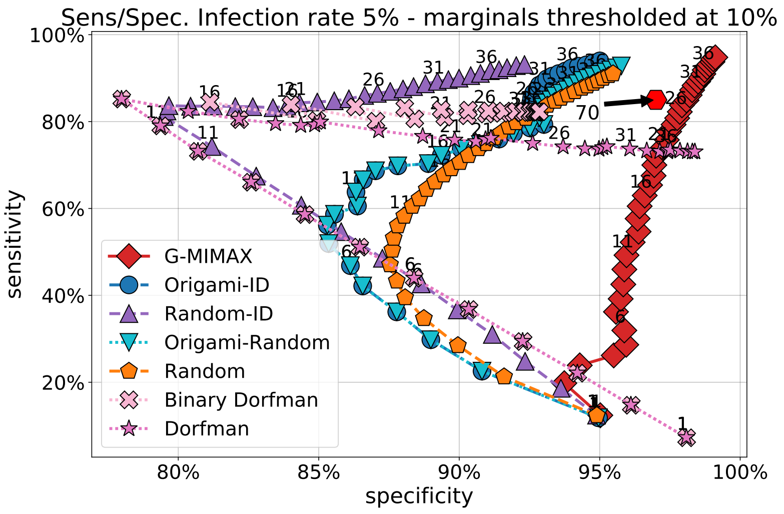


Figure 9: Robustness to misspecification of two crucial parameters: prior infection rates and sensitivity $\hat{q}$ and $\hat{s}$, compared to ground truth parameters used to generate ground truth and tests $q = 5\%$ and $s = 85\%$. Specificity is well specified in all experiments, i.e. $\sigma = \hat{\sigma} = 97\%$. Note that scales are relative to each plot, and highlight the robustness of our methods (bottom) to misspecification.





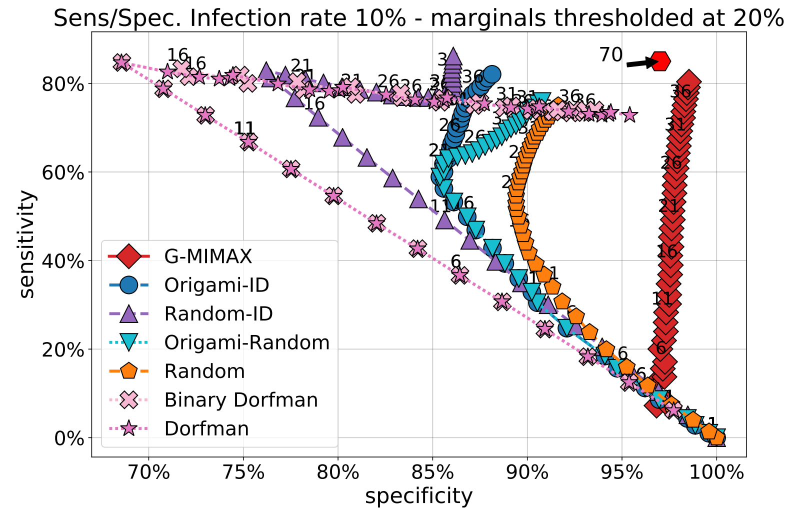


Figure 10: Experiments in which only one test is carried out at a time before recomputing the marginal and deciding on the next test. Here $F = 5, B = 4$ for G-MIMAX (with $N = 10000$) and all methods rely on a LBP decoder only.

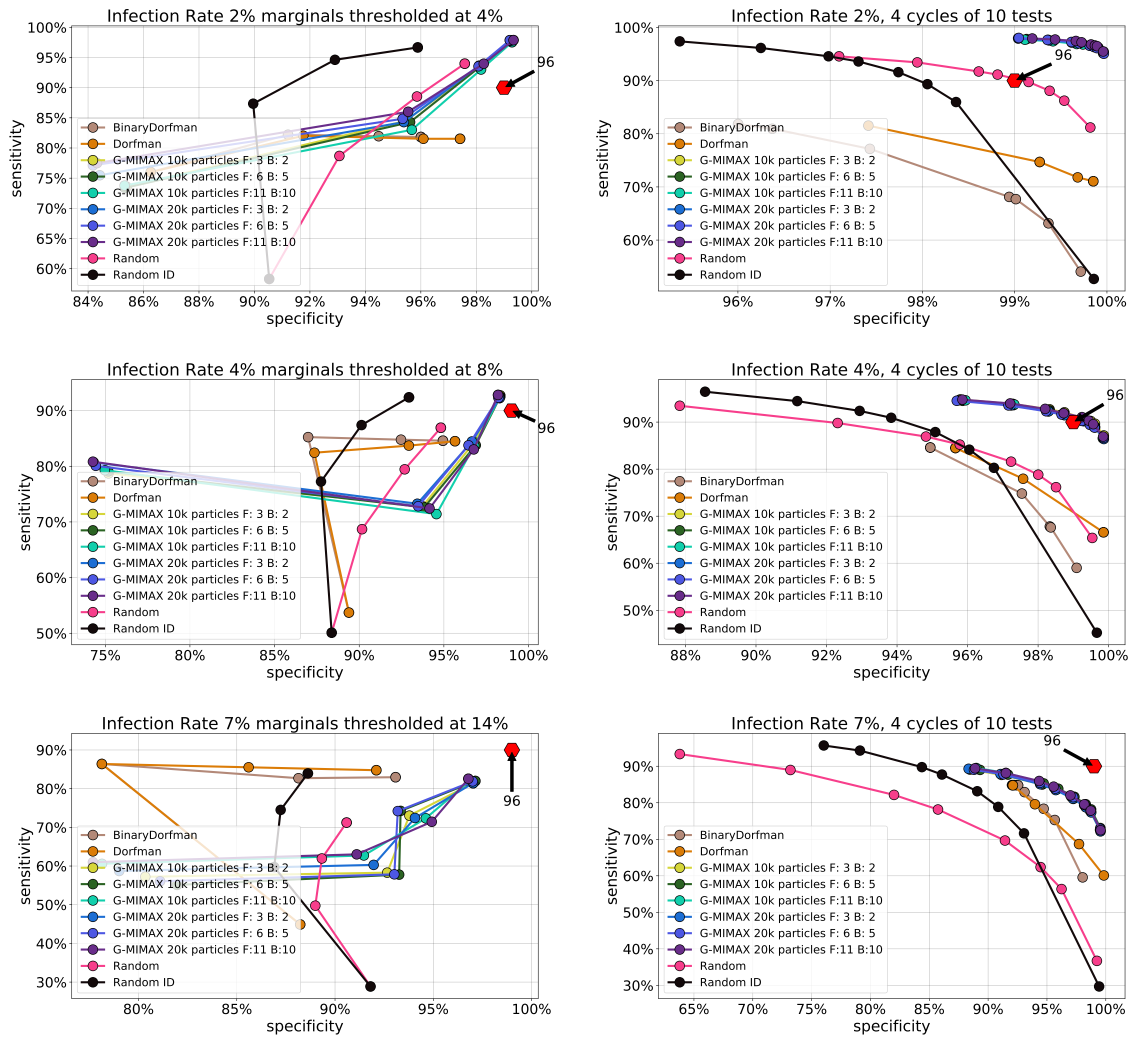


Figure 11: Experiments with a sensitivity that decreases with group size. Here we also show the relatively minor impact for the G-MIMAX strategy of choosing parameters such as N and F/B. The red hexagonal dot stands for the sensisitivy/specificity of a single test, knowing that the sensitivity for groups decreases by 1% every time the group size is increased by 1.