PAPRIKA: Private Online False Discovery Rate Control

Wanrong Zhang[1] **Gautam Kamath**[*2] **Rachel Cummings**[*3]

Abstract

In hypothesis testing, a *false discovery* occurs when a hypothesis is incorrectly rejected due to noise in the sample. When adaptively testing multiple hypotheses, the probability of a false discovery increases as more tests are performed. Thus the problem of *False Discovery Rate (FDR) control* is to find a procedure for testing multiple hypotheses that accounts for this effect in determining the set of hypotheses to reject. The goal is to minimize the number (or fraction) of false discoveries, while maintaining a high true positive rate (i.e., correct discoveries). In this work, we study False Discovery Rate (FDR) control in multiple hypothesis testing under the constraint of differential privacy for the sample. Unlike previous work in this direction, we focus on the *online setting*, meaning that a decision about each hypothesis must be made immediately after the test is performed, rather than waiting for the output of all tests as in the offline setting. We provide new private algorithms based on state-of-the-art results in non-private online FDR control. Our algorithms have strong provable guarantees for privacy and statistical performance as measured by FDR and power. We also provide experimental results to demonstrate the efficacy of our algorithms in a variety of data environments.

1. Introduction

In the modern era of big data, data analyses play an important role in decision-making in healthcare, information technology, and government agencies. The growing availability of large-scale datasets and ease of data analysis, while beneficial to society, has created a severe crisis of reproducibility in science. In 2011, Bayer HealthCare reviewed 67 in-house projects and found that they could replicate fewer than 25 percent, and found that over two-thirds of the projects had major inconsistencies (National Academies, 2019). One major reason is that random noise in the data can often be mistaken for interesting signals, which does not lead to valid and reproducible results. This problem is particularly relevant when testing multiple hypotheses, when there is an increased chance of false discoveries based on noise in the data. For example, an analyst may conduct 250 hypothesis tests and find that 11 are significant at the 5% level. This may be exciting to the researcher who publishes a paper based on these findings, but elementary statistics suggests that (in expectation) 12.5 of those tests should be significant at that level purely by chance, even if the null hypotheses were all true. To avoid such problems, statisticians have developed tools for controlling overall error rates when performing multiple hypothesis tests.

In hypothesis testing, the *null hypothesis* of no interesting scientific discovery (e.g., a drug has no effect), is tested against the alternative hypothesis of a particular scientific theory being true (e.g., a drug has a particular effect). The significance of each test is measured by a p*-value*, which is the probability of the observed data occurring under the null hypothesis, and a hypothesis is *rejected* if the corresponding p-value is below some (fixed) significance level. Each rejection is called a *discovery*, and a rejected hypothesis is a *false discovery* if the null hypothesis is actually true. When testing multiple hypotheses, the probability of a false discovery increases as more tests are performed. The problem of *false discovery rate (FDR) control* is to find a procedure for testing multiple hypotheses that takes in the p-values of each test, and outputs a set of hypotheses to reject. The goal is to minimize the number of false discoveries, while maintaining high true positive rate (i.e., true discoveries).

In many applications, the dataset may contain sensitive personal information, and the analysis must be conducted in a privacy-preserving way. For example, in genome-wide association studies (GWAS), a large number of single-nucleotide polymorphisms (SNPs) are tested for an association with a disease simultaneously or adaptively. Prior work has shown that the statistical analysis of these datasets can lead to

[*] Equal contribution as last author. [1] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA [2] Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada [3] Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA. Correspondence to: Wanrong Zhang <wanrongz@gatech.edu>, Gautam Kamath <g@csail.mit.edu>, Rachel Cummings <rac2239@columbia.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

privacy concerns, and it is possible to identify an individual's genotype when only minor allele frequencies are revealed (Homer et al., 2008). The field of *differential privacy* (Dwork et al., 2006) offers data analysis tools that provide powerful worst-case privacy guarantees, and has become a de facto gold standard in private data analysis. Informally, an algorithm that is ε-differentially private ensures that any particular output of the algorithm is at most e^{ε} more likely when a single data point is changed. This parameterization allows for a smooth tradeoff between accurate analysis and privacy to the individuals who have contributed data. In the past decade, researchers have developed a wide variety of differentially private algorithms for many statistical tasks; these tools have been implemented in practice at major organizations including Google (Erlingsson et al., 2014), Apple (Differential Privacy Team, Apple, 2017), Microsoft (Ding et al., 2017), and the U.S. Census Bureau (Dajani et al., 2017).

Related Work. The only prior work on differentially private FDR control (Dwork et al., 2018) considers the classic offline multiple testing problem, where an analyst has all the hypotheses and corresponding p-values upfront. Their private method repeatedly applies REPORTNOISYMIN (Dwork & Roth, 2014) to the celebrated Benjamini-Hochberg (BH) procedure (Benjamini & Hochberg, 1995) in offline multiple testing to privately pre-screen the p-values, and then applies the BH procedure again to select the significant p-values. The (non-private) BH procedure first sorts all p-values, and then sequentially compares them to an increasing threshold, where all p-values below their (ranked and sequential) threshold are rejected. The REPORTNOISYMIN mechanism privatizes this procedure by repeatedly (and privately) finding the hypothesis with the lowest p-value.

Although the work of (Dwork et al., 2018) showed that it was possible to integrate differential privacy with FDR control in multiple hypothesis testing, the assumption of having all hypotheses and p-values upfront is not reasonable in many practical settings. For example, a hospital may conduct multi-phase clinical trials where more patients join over time, or a marketing company may perform A/B testings sequentially. In this work, we focus on the more practical *online hypothesis testing problem*, where a stream of hypotheses arrive sequentially, and decisions to reject hypotheses must be made based on current and previous results before the next hypothesis arrives. This sequence of the hypotheses could be independent or adaptively chosen. Due to the fundamental difference between the offline and online FDR procedures, the method of (Dwork et al., 2018) based on REPORTNOISYMIN cannot be applied to the online setting. Instead, we use SPARSEVECTOR, described in Section 2.1, as a starting point. Discussion of non-private online multiple hypothesis testing appears in Section 2.2.

Our Results. We develop a differentially private online FDR control procedure for multiple hypothesis testing, which takes a stream of p-values and a target FDR level and privacy parameter ε, and outputs discoveries that can control the FDR at a certain level at any time point. Such a procedure provides unconditional differential privacy guarantees (to ensure that privacy will be protected even in the worst case) and satisfy the theoretical guarantees dictated by the FDR control problem.

Our algorithm, Private Alpha-investing P-value Rejecting Iterative sparse veKtor Algorithm (PAPRIKA, Algorithm 1), is presented in Section 3. Its privacy and accuracy guarantees are stated in Theorem 3 and 4, respectively. While the full proofs appear in the appendix, we describe the main ideas behind the algorithms and proofs in the surrounding prose. In Section 4, we provide a thorough empirical investigation of PAPRIKA, with additional empirical results in Appendix C.

2. Preliminaries

2.1. Background on Differential Privacy

Differential Privacy bounds the maximal amount that one data entry can change the output of the computation. Databases belong to the space $\mathcal{D}^n$ and contain n entries–one for each individual–where each entry belongs to data universe $\mathcal{D}$. We say that $D, D' \in \mathcal{D}^n$ are *neighboring databases* if they differ in at most one data entry.

Definition 1 (Differential Privacy (Dwork et al., 2006))**.** *An algorithm $\mathcal{M} : \mathcal{D}^n \to \mathcal{R}$ is* (ε, δ)-differentially private *if for every pair of neighboring databases $D, D' \in \mathbb{R}^n$, and for every subset of possible outputs $\mathcal{S} \subseteq \mathcal{R}$, $\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta$. If $\delta = 0$, we say that $\mathcal{M}$ is* ε-differentially private.

The *additive sensitivity* of a real-valued query $f : \mathcal{D}^n \to \mathbb{R}$ is denoted Δf, and is defined to be the maximum change in the function's value that can be caused by changing a single entry. That is, $\Delta f = \max_{D,D' \text{ neighbors}} |f(D) - f(D')|$. Differential privacy guarantees are often achieved by adding *Laplace noise* at various places in the computation, where the noise scales with $\Delta f / \varepsilon$. A Laplace random variable with parameter b is denoted Lap(b), and has probability density function, $p_{\text{Lap}(b)}(x) = \frac{1}{2b} \exp\left(\frac{-|x|}{b}\right) \quad \forall x \in \mathbb{R}$.

The SPARSEVECTOR algorithm, first introduced by (Dwork et al., 2010) and refined to its current form by (Dwork & Roth, 2014), privately reports the outcomes of a potentially very large number of computations, provided that only a few are "significant." It takes in a stream of queries and releases a bit vector indicating whether or not each noisy query answer is above the fixed noisy threshold. Pseudocode appears in Appendix A. We build off this algorithm, using it as a framework for our online private false discovery rate control algorithm as new hypotheses arrive online, and we only care about those "significant" hypotheses when the p-value is below a certain threshold. We note that the standard presentation below checks for queries with values above a threshold, but by simply changing signs this framework can be used to check for values *below* a threshold, as we will do with the p-values.

Theorem 1 ((Dwork et al., 2010))**.** *For any sequence of k queries $f_1, \dots, f_k$ with sensitivity Δ such that $|\{i : f_i(D) \geq T - \alpha_{SV}\}| \leq c$,* SPARSEVECTOR *outputs with probability at least $1 - \beta$ a stream of $a_1, \dots, a_k \in \{\top, \bot\}$ such that $a_i = \bot$ for every $i \in [m]$ with $f(i) < T - \alpha_{SV}$ and $a_i = \top$ for every $i \in [m]$ with $f(i) > T + \alpha_{SV}$ as long as $\alpha_{SV} \geq \frac{8\Delta c \log(2kc/\beta)}{\varepsilon}$.*

Unlike the conventional use of additive sensitivity, (Dwork et al., 2018) defined the notion of *multiplicative sensitivity* specifically for p-values. It is motivated by the observation that, although the additive sensitivity of a p-value may be large, the relative change of the p-value on two neighboring datasets is stable unless the p-value is very small. This notion allows us to treat the logarithm of the p-values as having additive sensitivity η, substantially reducing the scale of noise required to preserve privacy.

Definition 2 (Multiplicative Sensitivity (Dwork et al., 2018))**.** *A p-value function p is said to be (η, μ)-multiplicative sensitive if for all neighboring databases D and D', either both $p(D), p(D') \leq \mu$ or*

$$exp(-\eta)p(D) \leq p(D') \leq \exp(\eta)p(D).$$

2.2. Background on Online False Discovery Rate Control

In the online false discovery rate (FDR) control problem, a data analyst receives a stream of hypotheses on the database D, or equivalently, a stream of p-values $p_1, p_2, \dots$. The analyst must pick a threshold α_t at each time t to reject the hypothesis when $p_t \leq \alpha_t$; this threshold can depend on previous hypotheses and discoveries, and rejection must be decided before the next hypothesis arrives.

The error metric is the false discovery rate, formally defined as: $\text{FDR} = \mathbb{E}\left[\text{FDP}\right] = \mathbb{E}\left[\frac{|\mathcal{H}^0 \cap \mathcal{R}|}{|\mathcal{R}|}\right]$, where $\mathcal{H}^0$ is the (unknown to the analyst) set of hypotheses where the null hypothesis is true, and $\mathcal{R}$ is the set of rejected hypotheses. We will also write these terms as a function of time t to indicate their values after the first t hypotheses: $\text{FDR}(t)$, $\text{FDP}(t)$, $\mathcal{H}^0(t)$, $\mathcal{R}(t)$. The goal of FDR control is to guarantee that for any time t, the FDR up to time t is less than a pre-determined quantity $\alpha \in (0, 1)$.

Such a problem was first investigated by (Foster & Stine, 2008), who proposed a framework known as *online alpha-investing* that models the hypothesis testing problem as an investment problem. The analyst is endowed with an initial budget, can test hypotheses at a unit cost, and receives an additional reward for each discovery. The alpha-investing procedure ensures that the analyst always maintains an α-fraction of their wealth, and can therefore continue testing future hypotheses indefinitely. Unfortunately, this approach only controls a slightly relaxed version of FDR, known as *mFDR*, which is given by $\text{mFDR}(t) = \frac{\mathbb{E}\left[|\mathcal{H}^0 \cap \mathcal{R}|\right]}{\mathbb{E}[|\mathcal{R}|]}$. This approach was later extended to a class of generalized alpha-investing (GAI) rules (Aharoni & Rosset, 2014).

A generalized alpha-investing procedure starts with an initial wealth $W(0) = \alpha$, where α is the testing level. It uses a GAI rule $\mathcal{I}_{W(0)}$ that takes in past rejections to determine three quantities at each time t: the level of the test α_t, the amount φ_t subtracted from the wealth, and the reward ψ_t received for each discovery. $(\alpha_t, \varphi_t, \psi_t) = \mathcal{I}_{W(0)}(\{R_1, R_2, \dots, R_{t-1}\})$. The wealth updating rule is $W(t) = W(t-1) - \varphi_t + R_t\psi_t$. A GAI rule maintains nonnegative wealth $W(t) \geq 0$ for any t, and the following ineuqality holds:

$$0 \leq \psi_t \leq \min(\frac{\varphi_t}{\rho_t} + \alpha, \frac{\varphi_t}{\rho_t} + \alpha - 1), \tag{1}$$

where ρ_t is the best power of the t-th test.

One subclass of GAI rules, the Level based On Recent Discovery (LORD), was shown to have consistently good performance in practice (Javanmard & Montanari, 2015; 2018). GAI++ in (Ramdas et al., 2017) improves the class of GAI, with LORD++ as an explicit example. The SAFFRON procedure, proposed by (Ramdas et al., 2018), further improves the LORD procedures by adaptively estimating the proportion of true nulls, and is the current state-of-the-art in online FDR control for multiple hypothesis testing.

To understand the main differences between the SAFFRON and the LORD procedures, we first introduce an oracle estimate of the FDP as $\text{FDP}^*(t) = \frac{\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j}{|\mathcal{R}(t)|}$. The numerator $\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j$ overestimates the number of false discoveries, so $\text{FDP}^*(t)$ overestimates the FDP. The oracle estimator $\text{FDP}^*(t)$ cannot be calculated since $\mathcal{H}^0$ is unknown. LORD's naive estimator $\sum_{j \leq t} \alpha_j / |\mathcal{R}(t)|$ is a natural overestimate of $\text{FDP}^*(t)$. The SAFFRON's threshold sequence is based on a novel estimate of FDP as $\widehat{\text{FDP}}_{\text{SAFFRON}}(t) = \frac{\sum_{j \leq t} \alpha_j \frac{I(p_j > \lambda_j)}{1 - \lambda_j}}{|\mathcal{R}(t)|}$, where $\{\lambda_j\}_{j=1}^{\infty}$ is a sequence of user-chosen parameters in the interval $(0, 1)$, which can be a constant or a deterministic function of the information up to time $t - 1$. This estimate provides the null-proportion adaptivity basis for SAFFRON.

Our private algorithm is built upon the LORD++ and the SAFFRON algorithms, which are given formally in Algorithm 3 and 4 in Appendix A. As a class of GAI, the

LORD++ and the SAFFRON both start off with an error budget, which will be allocated to different tests over time. The wealth budget decays as each hypothesis is tested, and it earns back wealth on every rejection except for the first. The decay factors γ_j that depreciate past wealth is a non-increasing sequence summing to one, which ensures that the sum of the wealth budget is always below the desired level α. SAFFRON involves an additional candidacy checking step to be null-proportion adaptive: it never loses wealth when testing candidate p-values with $p_j < \lambda_j$. The sequence $\{\lambda_j\}_{j=1}^{\infty}$ can be defined by any coordinatewise non-decreasing function g_t. For example, $\{\lambda_j\}_{j=1}^{\infty}$ can be a deterministic sequence of constants, or $\lambda_t = \alpha_t$, as in the case of alpha-investing. These λ_j values serve as a weak overestimate of α_j. The algorithm first checks if a p-value is below λ_j, and if so, adds it to the *candidate set* of hypotheses that may be rejected. It then computes the α_j threshold based on current wealth, current size of the candidate set, and the number of rejections so far, and decides to reject the hypothesis if $p_j \leq \alpha_j$.

Both LORD++ and SAFFRON require that the input sequence of p-values are still valid p-values given past information. which is formalized as *conditional super-uniformity* of null p-values, with respect to a filtration process on the sequence of rejection decisions $\{R_j\}$ and candidacy $\{C_j\}$ (for SAFFRON). This is stated formally in Appendix A. Intuitively, it means that the sequence of hypotheses cannot be overly adaptive. Independent p-values is a special case of conditional super-uniformity.

SAFFRON provides the following accuracy guarantees under this condition.

Theorem 2 ((Ramdas et al., 2018)). *If the null p-values are conditionally super-uniformly distributed, then we have:*
(a) $\mathbb{E}\left[\sum_{j\leq t, j\in\mathcal{H}^0} \alpha_j \frac{I(p_j>\lambda_j)}{1-\lambda_j}\right] \geq \mathbb{E}\left[|\mathcal{H}^0 \cap \mathcal{R}(t)|\right]$*;*
(b) The condition $\widehat{FDP}_{SAFFRON}(t) \leq \alpha$ *for all* $t \in \mathbb{N}$ *implies that* $mFDR(t) \leq \alpha$ *for all* $t \in \mathbb{N}$.
If the null p-values are independent of each other and of the non-null p-values, and $\{\alpha_t\}$ and $\{\lambda_t\}$ are coordinatewise non-decreasing functions of the vector $R_1, \ldots, R_{t-1}, C_1, \ldots, C_{t-1}$*, then*
(c) $\mathbb{E}\left[\widehat{FDP}_{SAFFRON}(t)\right] \geq \mathbb{E}\left[FDP(t)\right] := FDR(t)$ *for all* $t \in \mathbb{N}$*;*
(d) The condition $\widehat{FDP}_{SAFFRON}(t) \leq \alpha$ *for all t implies that* $FDR(t) \leq \alpha$ *for all* $t \in \mathbb{N}$.

3. Private online false discovery rate control

In this section, we provide our algorithm for private online false discovery rate control, PAPRIKA, given formally in Algorithm 1. It starts with SAFFRON, using SPARSEVECTOR to ensure privacy of the rejection set. However, the combination of these tools is far from immediate, and several algorithmic innovations are required, including: dynamic thresholds in SPARSEVECTOR to accommodate the alpha-investing rule, adding noise that scales with the multiplicative sensitivity of p-values to reduce the noise required for privacy, shifting the SparseVector threshold to accommodate FDR as a novel accuracy metric, and the candidacy indicator step which cannot be done privately and requires modifications to the wealth updates. We resolve this by using a similar wealth updating rule as in LORD++. We provide new analysis for both privacy and accuracy. Complete proofs of our privacy and accuracy results appear in the appendix; we elaborate here on the algorithmic details and why these modifications are needed to ensure privacy and FDR control.

The non-private online false discovery rate control algorithms decide to reject hypothesis t if the corresponding p-value p_t is less than the rejection threshold α_t; that is, if $p_t \leq \alpha_t$. We instantiate the SPARSEVECTOR framework in this setting, where p_t plays the role of the t^{th} query answer $f_t(X)$, and α_t plays the role of the threshold. Note that SPARSEVECTOR uses a single fixed threshold for all queries, while our algorithm PAPRIKA allows for a dynamic threshold that depends on the previous output. Our privacy analysis of the algorithm accounts for this change and shows that dynamic thresholds do not affect the privacy guarantees of SPARSEVECTOR. However, the algorithm would not be private if the dynamic thresholds also depend on the data. Note that SAFFRON never loses wealth when testing candidate p-values with $p_j \leq \lambda_j$, and the threshold α_j depends on the data since it is based on current wealth. We remove such dependence in PAPRIKA by losing wealth at every step regardless of whether we test a candidate p-values, similar to LORD++. This will result in stricter FDR control (and potentially weaker power) because our wealth decays faster.

Similar to prior work on private offline FDR control (Dwork et al., 2018), we use *multiplicative sensitivity* as described in Definition 2, as p-values may have high sensitivity and require unacceptably large noise to be added to preserve privacy. We assume that our input stream of p-values $p_1, p_2, \ldots,$ each has multiplicative sensitivity (η, μ). As long as μ is small enough (i.e., less than the rejection threshold), we can treat the logarithm of the p-values as the queries with additive sensitivity η. Because of this change, we must make rejection decisions based on the logarithm of the p-values, so our reject condition is $\log p_t + Z_t \leq \log \alpha_t + Z_\alpha$ for Laplace noise terms Z_t, Z_α drawn from the appropriate distributions.

Algorithm 1 PAPRIKA($\alpha, \lambda, W_0, \gamma, c, \varepsilon, \delta, s$)

```
Input: stream of $p$-values $\{p_1, p_2, \ldots\}$ with mutiplicative sensitivity ($\eta$,$\mu$), target FDR level $\alpha$, initial wealth $W_0 < \alpha$, positive non-increasing sequence $\{\gamma_j\}_{j=0}^{\infty}$ of summing to one, expected number of rejections $c$, privacy parameters $\varepsilon, \delta$, threshold shift magnitude $s$, maximum number of $p$-values $k$.
Let $Z_\alpha^0 \sim \text{Lap}(2\eta c/\varepsilon)$, count $= 0$,
$A = \frac{sc\eta}{\varepsilon} \log \frac{2}{3\min\{\delta, 1-((1-\delta)/\exp(\varepsilon))^{1/k}\}}$
for each p-value $p_t$ do
    if count $\geq c$ then Output $R_t = 0$
    else
    Sample $Z_t \sim \text{Lap}(4\eta c/\varepsilon)$. Set $\lambda_t = g_t(R_{1:t-1}, C_{1:t-1})$. Set the indicator for candidacy $C_t = I(\log p_t < \log 2\lambda_t)$.
    if $t = 1$
    then Set $\alpha_1 = (1 - 2\lambda_1)\gamma_1 W_0$
    else
    Compute $\alpha_t = (1 - 2\lambda_t)(W_0\gamma_t + (\alpha - W_0)\gamma_{t-\tau_1} + \sum_{j\geq 2} \alpha\gamma_{t-\tau_j})$
    if $C_t = 1$ and $\log p_t + Z_t \leq \log \alpha_t - A + Z_\alpha^{\text{count}}$
    then Output $R_t = 1$. Set count = count $+1$ and sample $Z_\alpha^{\text{count}} \sim \text{Lap}(2\eta c/\varepsilon)$
    else Output $R_t = 0$
end for
```

The accuracy guarantees of SparseVector ensure that if a value is reported to be below threshold, then with high probability it will not be more than α_{SV} above the threshold. However, to ensure that our algorithm satisfies the desired bound $FDR \leq \alpha$, we require that reports of "below threshold" truly do correspond to p-values that are below the desired threshold α_t. To accommodate this, we shift our rejection threshold $\log \alpha_t$ down by a parameter A. A is chosen such that the algorithm satisfies (ε, δ)-differential privacy, but the choice can be seen as inspired by the α_{SV}-accuracy term of SparseVector as given in Theorem 1. Therefore our final reject condition is $\log p_t + Z_t \leq \log \alpha_t - A + Z_\alpha$. This ensures that "below threshold" reports are below $(\log \alpha_t - A) + \alpha_{SV} \approx \log \alpha_t$ with high probability. Empirically, we see that the bound of A in Theorem 3 may be overly conservative and lead to no hypotheses being rejected, so we allow an additional scaling parameter s that will scale the magnitude of shift by a factor of s. The conservative bounds of Theorem 3 correspond to $s = 4$, but in many scenarios a smaller value of $s = 1$ or 2 will lead to better performance while still satisfying the privacy guarantee. Further guidance choosing this shift parameter is given in Appendix C.1.

Even with these modifications, a naive combination of SparseVector and SAFFRON would still not satisfy differential privacy. This is due to the *candidacy indicator* step of the algorithm. In the SAFFRON algorithm, a preprocessing candidacy step occurs before any rejection decisions. This step checks whether each p-value p_t is smaller than a loose upper bound λ_t on the eventual reject threshold α_t. The algorithm chooses α_t using an α-investing rule that depends on the number of candidate hypotheses seen so far, and ensures that $\alpha_t \leq \lambda_t$, so only hypotheses in this candidate set can be rejected. These λ values are used to control $\widehat{\text{FDP}}_{\text{SAFFRON}}(t)$, which serves as a conservative overestimate of FDP(t). (For a discussion of how to choose λ_t, see Lemma 1 or our experimental results in Section 4. Reasonable choices would be $\lambda_t = \alpha_t$ or a small constant such as 0.2.)

Without adding noise to the candidacy condition, there may be neighboring databases with p-values p_t, p_t' for some hypothesis such that $\log p_t < \log \lambda_t < \log p_t'$, and hence the hypothesis would have positive probability of being rejected under the first database and zero probability of rejection under the neighbor. This would violate the $(\varepsilon, 0)$-differential privacy guarantee intended under SparseVector. If we were to privatize the condition for candidacy using, for example, a parallel instantiation of SparseVector, then we would have to reuse the same realizations of the noise when computing the rejection threshold α_t to still control FDP, but this would no longer be private.

Since we cannot add noise to the candidacy condition, we weaken it in PAPRIKA to be $\log p_t < \log 2\lambda_t$[1] Then if a hypothesis has different candidacy results under neighboring databases and the multiplicative sensitivity η is small, then the hypothesis is still extremely unlikely to be rejected even under the database for which it was candidate. To see this, consider a pair of neighboring databases that induce p-values where $\log p_t < \log 2\lambda_t < \log p_t'$. Due to the multiplicative sensitivity constraint, we know that $\log p_t \geq \log 2\lambda_t - \eta$. Plugging this into the rejection condition $\log p_t + Z_t \leq \log \alpha_t - A + Z_\alpha$, we see that we would need the difference of the noise terms to satisfy $Z_t - Z_\alpha \leq \log \frac{1}{2} - A + \eta$, which by analysis of the Laplace distribution, will happen with exponentially small probability in n when $\eta = \text{poly}^{-1}(n)$.[2] Our PAPRIKA algorithm is thus (ε, δ)-differentially private, and we account for this

[1] We note that although this change is algorithmically equivalent to scaling up the parameter λ_t by a factor of 2, this slack is relevant for certain instantiations of PAPRIKA that set $\lambda_t = \alpha_t$, which we show perform well empirically. (See Section 4 for more details.) We write this step as a relaxation of the candidacy condition both for notational consistency with existing non-private alpha-investing-based FDR control methods, such as SAFFRON AI (Ramdas et al., 2018), that also choose $\lambda_t = \alpha_t$, and to emphasize that this slack in the candidacy condition is necessary in ensuring differential privacy of the overall algorithm.

[2] Such values of η are typical; see examples in Section 4 where $\eta = \frac{1}{\sqrt{n}}$. The shift term A also has dependence on η which contributes to the bound.

failure probability in our (exponentially small) δ parameter, as stated in Theorem 3.

One may wonder whether this candidacy step in necessary at all. Since we have removed the dependence of α_t on the size of the candidate set in PAPRIKA, the threshold α_t is no longer null-proportion sensitive. The advantage of being null-proportion adaptive in SAFFRON increases as the proportion of non-nulls increases, but we focus on the case where the non-nulls are sparse, and thus it has little impact in our setting. In Section 4, we empirically compare PAPRIKA to two private versions of LORD++, which we call PrivLORD and PrivLORD2. The former combines SPARSEVECTOR and LORD++, with the same threshold shifting as described earlier in this section. The latter adds the candidacy checking step on top of PrivLORD. We see in Section 4.2 that both methods provide poor FDR control relative to PAPRIKA, thus providing empirical evidence that the candidacy step in PAPRIKA plays a vital role in FDR control, even if α_t is not null-proportion sensitive. Further details about PrivLORD and PrivLORD2 are deferred to Appendix C.

Our PAPRIKA algorithm allows analysts to specify a maximum number of hypotheses tested k and rejections c. We require a bound on the maximum number of hypotheses tested because the accuracy guarantees of SPARSEVECTOR only allows exponentially (in the size of the database) many queries to be answered accurately. Once the total number of rejections reaches c, the algorithm will fail to reject all future hypotheses. We do not halt the algorithm as in SPARSEVECTOR and therefore, PAPRIKA does not have a stopping criterion, and we can safely talk about the FDR control at any fixed time, just like SAFFRON and LORD++.

Our algorithm also controls at each time t, $\widehat{\text{FDP}}_{\text{PAPRIKA}}(t) \leq \frac{\sum_{j\leq t} \alpha_t \frac{I(p_j > 2\lambda_j)}{1-2\lambda_j}}{|\mathcal{R}(t)|}$. We note that this is equivalent to $\widehat{\text{FDP}}_{\text{SAFFRON}}(t)$ by scaling down λ_j by a factor of 2. By analyzing and bounding this expression, we achieve FDR bounds for our PAPRIKA algorithm, as stated in Theorem 4.

Theorem 3. *For any stream of p-values* $\{p_1, p_2, \ldots\}$, PAPRIKA *is* (ε, δ)*-differentially private.*

As a starting point, our privacy comes from SPARSEVECTOR, but as discussed above, many crucial modifications are required. To briefly summarize the key considerations, we must handle different thresholds at different times, multiplicative rather than additive sensitivity, a modified notion of the candidate set, and introducing a small delta parameter to account for the new candidate set definition and the shift. The proof of Theorem 3 appears in Appendix D.

Next we describe the theoretical guarantees of FDR control for our private algorithm PAPRIKA which is an analog of Theorem 2. We modify the notation of the conditional super-uniformity assumption of SAFFRON to incorporate the added Laplace noise. The conditions are otherwise identical. (See (2) in Appendix A for comparison.) We note that independent p-values is a special case of conditional super-uniformity, but this requirement more generally allows for a broader class of dependencies among p-values. Let $R_j := I(p_j + Z_j \leq \alpha_j + Z_\alpha)$ be the rejection decisions, and let $C_j := I(p_j \leq 2\lambda_j)$ be the indicators for candidacy. We let $\alpha_t := f_t(R_1, \ldots, R_{t-1}, C_1, \ldots, C_{t-1})$, where f_t is an arbitrary function of the first $t-1$ indicators for rejections and candidacy. Define the filtration formed by the sequences of σ-fields ${\mathcal{F}'}^t := \sigma(R_1, \ldots, R_t, C_1, \ldots, C_t, Z_1, \ldots, Z_t, Z_\alpha)$. The null p-values are conditionally super-uniformly distributed with respect to the filtration $\mathcal{F}'$ if when the null hypothesis H_i is true, then $\Pr(p_t \leq \alpha_t | {\mathcal{F}'}^{t-1}) \leq \alpha_t$. We emphasize that this condition is only needed for FDR control, and that our privacy guarantee of Theorem 3 holds for arbitrary streams of p-values, even those which do not satisfy conditional super-uniformity.

Our FDR control guarantees for PAPRIKA mirror those of SAFFRON (Theorem 2). The first two statements apply if p-values are conditionally super-uniform, and the last two statements apply if the p-values are additionally independent under the null. The proof of Theorem 4 appears in Appendix E.

Theorem 4. *If the null p-values are conditionally super-uniformly distributed, then we have:*
(a) $\mathbb{E}\left[\sum_{j\leq t, j\in\mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1-2\lambda_j}\right] + \delta t \geq \mathbb{E}\left[|\mathcal{H}^0 \cap \mathcal{R}(t)|\right]$*;*
(b)The condition $\widehat{FDP}_{\text{PAPRIKA}}(t) \leq \alpha$ *for all* $t \in \mathbb{N}$ *implies that* $mFDR(t) \leq \alpha + \delta t$ *for all* $t \in \mathbb{N}$.
If the null p-values are independent of each other and of the non-null p-values, and $\{\alpha_t\}$ *and* $\{\lambda_t\}$ *are coordinate-wise non-decreasing functions of the vector* $R_1, \ldots, R_{t-1}, C_1, \ldots, C_{t-1}$*, then*
(c) $\mathbb{E}\left[\widehat{FDP}_{\text{PAPRIKA}}(t)\right] + \delta t \geq \mathbb{E}\left[FDP(t)\right] := FDR(t)$ *for all* $t \in \mathbb{N}$*;*
(d) The condition $\widehat{FDP}_{\text{PAPRIKA}}(t) \leq \alpha$ *for all* t *implies that* $FDR(t) \leq \alpha + \delta t$ *for all* $t \in \mathbb{N}$.

Relative to the non-private guarantees of Theorem 2, the FDR bounds provided by PAPRIKA are weaker by an additive of δt. In most differential privacy applications, δ is typically required to be cryptographically small (i.e., at most negligible in the size of the database) (Dwork & Roth, 2014), so this additional term should have a minuscule effect on the FDR.[3] We note that ε plays a role in the analysis of Theorem

[3] Alternatively, δ could be treated like a tunable parameter to balance the tradeoff between privacy and FDR control. If an analyst has an upper bound on the allowable slack in FDR, say 0.01, then she could set $\delta = 0.01/t$ to ensure her desired bound.

4, although it does not appear in FDR bounds. Equation (22) in the appendix shows that the additive slack term δt in Theorem 4 is in fact $\min\left\{\delta, 1 - ((1-\delta)/\exp(\varepsilon))^{\frac{1}{k}}\right\} t$, which is upper bounded by δt.

The following lemma is a key tool in the proof of Theorem 4. Though it is qualitatively similar to Lemma 2 in (Ramdas et al., 2018), it is crucially modified to show an analogous statement holds under the addition of Laplace noise. Its proof appears in Appendix F.

Lemma 1. *Assume $p_1, p_2, \ldots$ are all independent and let $h : \{0,1\}^k \to R$ be any coordinate-wise non-decreasing function. Assume f_t and g_t are coordinate-wise non-decreasing functions and that $\alpha_t = f_t(R_{1:t-1}, C_{1:t-1})$ and $\lambda_t = g_t(R_{1:t-1}, C_{1:t-1})$. Then for any $t \leq k$ such that $H_t \in \mathcal{H}^0$, we have $\mathbb{E}\left[\frac{\alpha_t I(p_t > 2\lambda_t)}{(1-2\lambda_t)h(R_{1:k})} | \mathcal{F}'^{t-1}\right] \geq \mathbb{E}\left[\frac{\alpha_t}{h(R_{1:k})} | \mathcal{F}'^{t-1}\right]$ and $\mathbb{E}\left[\frac{\min\{\alpha_t \exp(Z_\alpha - Z_t - A), 1\}}{h(R_{1:k})} | \mathcal{F}'^{t-1}\right] \geq \mathbb{E}\left[\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(R_{1:k})} | \mathcal{F}'^{t-1}\right]$.*

There are no known theoretical bounds on the statistical power of SAFFRON even in the non-private setting. Instead, we validate power empirically through the experimental results in Section 4.

4. Experiments

We experimentally compare the FDR and the statistical power of variations of the PAPRIKA and SAFFRON procedures, under different sequences of $\{\lambda_j\}$. Following the convention of (Ramdas et al., 2018), we define PAPRIKA-Alpha-Investing, or PAPRIKA AI, to be the instantiation of Algorithm 1 with the sequence $\lambda_j = \alpha_j$, where the rejection threshold matches the α-investing rule, and we use PAPRIKA to denote Algorithm 1 instantiated with a sequence of constant of λ_j, which in our experiments is $\lambda_j = 0.2$. We use $\lambda_j = 0.5$ in SAFFRON.[4] We generally observe that, even under moderately stringent privacy restrictions, PAPRIKA and its AI variant perform comparably to the non-private alternatives, with PAPRIKA AI typically outperforming PAPRIKA. This suggests that even though setting λ_j as a fixed constant may be easier for implementation, parameter optimization can lead to meaningful performance improvements. We chose the sequence $\{\gamma_j\}$ to be a constant $1/k$ up to time k. Note that the sequence can be decreasing such as of the form $\gamma_j \propto j^{-s}$ in (Ramdas et al., 2018), which controls the wealth to be more concentrated around small values of j. See (Ramdas et al., 2018) for more discussion on the choice of $\{\gamma_j\}$. In our experiments, we set the target FDR level $\alpha + \delta t = 0.2$, and thus our privacy parameter δ is set to be bounded by $0.2/800 = 2.5 \times 10^{-4}$. The maximum number of rejections $c = 40$. All the results are averaged over 100 runs. We investigate two settings: the observations come Bernoulli distributions in Section 4.1, and the observations are generated from truncated exponential distributions in Section 4.3. In Section 4.2, we compare our algorithm against other private algorithms. In Appendix C.1, we discuss our choice of the shift parameter A and give guidance on how to choose this parameter in practice. Code for PAPRIKA and our experiments is available at `https://github.com/wanrongz/PAPRIKA`.

4.1. Testing with Bernoulli Observations

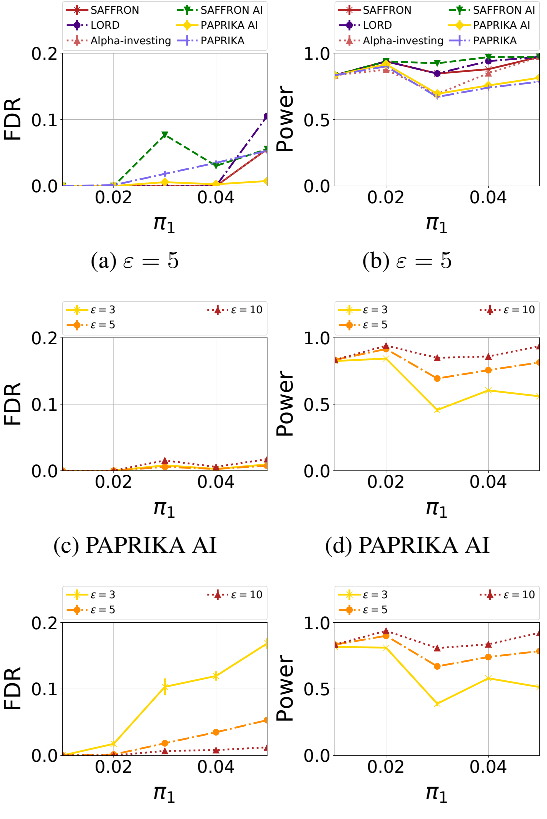
We assume that the database D contains n individuals with k independent features. The ith feature is associated with n i.i.d. Bernoulli variables $\xi_1^i, \ldots, \xi_n^i$, each of which takes the value 1 with probability θ_i, and takes the value 0 otherwise. Let t_i be the sum of the ith features. A p-value for testing null hypothesis $H_0^i : \theta_i \leq 1/2$ against $H_1^i : \theta_i > 1/2$ is given by $p_i(D) = \sum_{k=t_i}^{n} \frac{1}{2^n}\binom{n}{k}$. (Dwork et al., 2018) showed that p_i is (μ, η)-multiplicatively sensitive for $\mu = m^{-1-c}$ and $\eta \asymp \sqrt{\frac{\log n}{n}}$, where $m \leq \mathrm{poly}(n)$ and c is any small positive constant. We choose $\theta_i = 0.5$ with probability $1 - \pi_1$, and $\theta_i = 0.75$ with probability π_1, for varying values of π_1, which represents the expected fraction of non-null hypotheses. We consider relatively small values of π_1 as most practical applications of FDR control (such as GWAS studies) will have only a small fraction of true "discoveries" in the data.

In the following experiments, we sequentially test H_0^i versus H_1^i for $i = 1, \ldots, k$. We use $n = 1000$ as the size of the database D, and $k = 800$ as the number of features as well as the number of hypotheses. Our experiments are run under several different shifts A, but due to space constraints, we only report results in the main body with $A = \frac{c\eta}{\varepsilon} \log \frac{2}{3\min\{\delta, 1-((1-\delta)/\exp(\varepsilon))^{1/k}\}}$ (i.e., when $s = 1$), which still satisfies our privacy guarantee. Further discussion on the choice of A and additional results under other shift parameters s are deferred to Appendix C.1. The results are summarized in Figure 1, which plots the FDR and statistical power against the expected fraction of non-nulls, π_1. In Figure 1(a) and (b), we compare our algorithms with privacy parameter $\varepsilon = 5$ to the non-private baseline methods of LORD (Javanmard & Montanari, 2015; 2018), Alpha-investing (Aharoni & Rosset, 2014), and SAFFRON and SAFFRON AI from (Ramdas et al., 2018). In Figure 1(c,d) and (e,f), we compare the performance of PAPRIKA AI and PAPRIKA, respectively, with varying privacy parameters $\varepsilon = 3, 5, 10$. We also list these values in Table 1 in Appendix C.2.

As expected, the performance of PAPRIKA generally diminishes as ε decreases. A notable exception is that FDR

[4] Recall from Section 3 that our λ_j is equivalent to the λ_j in SAFFRON scaling down by a factor of 2.

also decreases in Figure 1(c). This phenomenon is because we set $\lambda_j = \alpha_j$, resulting in a smaller candidacy set and leading to insufficient rejections. Surprisingly, PAPRIKA AI also yields a lower FDR than many of the non-private algorithms (Figure 1(a)), since it tends to make fewer rejections. We also see that PAPRIKA AI performs dramatically better than PAPRIKA, suggesting that the choice of $\lambda_j = \alpha_j$ should be preferred to constant λ_j to ensure good performance in practice.



(a) $\varepsilon = 5$ (b) $\varepsilon = 5$

(c) PAPRIKA AI (d) PAPRIKA AI

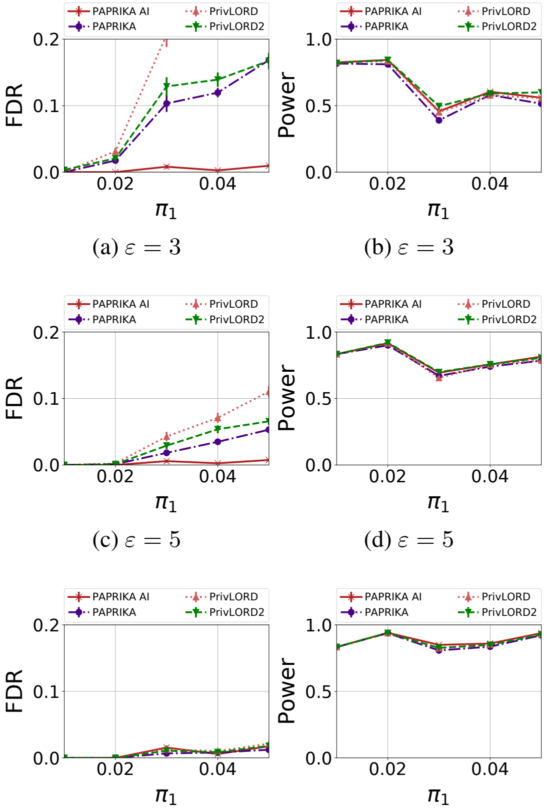
(e) PAPRIKA (f) PAPRIKA

Figure 1. FDR and statistical power versus fraction of non-null hypotheses π_1 for PAPRIKA (with $\lambda_j = 0.2$), PAPRIKA AI (with $\lambda_j = \alpha_j$), and non-private algorithms when the database consists of Bernoulli observations.

4.2. Comparison with Other Private Algorithms

As PAPRIKA is the first algorithm for private online FDR control, there is no private baseline for comparison. In Appendix C, we show that naïve Laplace privatization of SAFFRON is ineffective. This naïve approach applies the Laplace Mechanism (Dwork et al., 2006) to the p-values of each hypothesis, and then uses these noisy p-values as input to SAFFRON. We see that this baseline mechanism performs extremely poorly relative to PAPRIKA and PAPRIKA AI.

We also compare our PAPRIKA against PrivLORD and PrivLORD2 with Bernoulli observations in Figure 2 and truncated exponential observations in Figure 6 in Appendix 4.3. For comparison, we use the same shift A for the four algorithms, but we note that A should be larger in PrivLORD to control FDR at the level 0.2 as it lacking the candidate checking step, and a larger A leads to worse power, see Section C.1.



(a) $\varepsilon = 3$ (b) $\varepsilon = 3$

(c) $\varepsilon = 5$ (d) $\varepsilon = 5$

(e) $\varepsilon = 10$ (f) $\varepsilon = 10$

Figure 2. FDR and statistical power versus fraction of non-nulls π_1 for PAPRIKA (with $\lambda_j = 0.2$), PAPRIKA AI (with $\lambda_j = \alpha_j$), and PrivLORD and PrivLORD2 when the database consists of Bernoulli observations.

We make three key observations. First, PrivLORD makes significantly more false discoveries than the other three algorithms, suggesting that the candidacy checking step largely offsets against the added noise for private algorithms. The performance of PrivLORD gets closer to PAPRIKA and PAPRIKA AI when we add less noise as ε goes large. Second, PAPRIKA with constant λ_t has stricter FDR control and slightly weaker power compared to PrivLORD2 as expected, since the threshold α_t in PAPRIKA has an additional constant $(1 - 2\lambda_t)$ factor. Third, PAPRIKA AI provides dramatically better FDR and power tradeoffs—it controls FDR at a much lower level while maintaining power at a similar level as other methods (even the best in Figure 2(f) and 6(d)), suggesting PAPRIKA with a smart choice of the predictable sequence $\{\lambda_t\}$ is preferred.

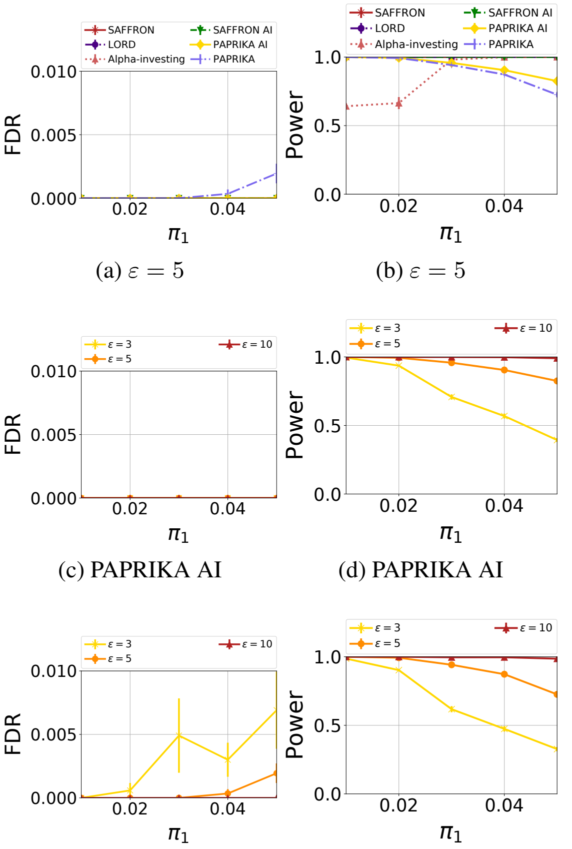
4.3. Testing with Truncated Exponential Observations

We again assume that the database D contains n individuals with k independent features. The ith feature is associated with n i.i.d. truncated exponential distributed variables $\xi_1^i, \ldots, \xi_n^i$, each of which is sampled according to density $f_i(x \mid \theta_i, b) = \frac{\theta_i \exp(-\theta_i x)}{1-\exp(-b\theta_i)} I(0 \leq x \leq b)$, for positive parameters b and θ_i. Let t_i be the realized sum of the ith features, and let T_i denote the random variable of the sum of the n truncated exponential distributed variables in the ith entry. A p-value for testing the null hypothesis $H_0^i : \theta_i = 1$ against the alternative hypothesis $H_1^i : \theta_i > 1$ is given by, $p_i(D) = \Pr_{\theta_i = 1}(T_i > t_i)$. (Dwork et al., 2018) showed that p_i is (μ, η)-multiplicatively sensitive for $\mu = m^{-1-c}$ and $\eta \asymp \sqrt{\frac{\log n}{n}}$, where $m \leq \text{poly}(n)$ and c is any small positive constant. In the following experiments, we generate our database using the exponential distribution model truncated at $b = 1$. We set $\theta_i = 1$ with probability $1 - \pi_1$, and $\theta_i = 1.95$ with probability π_1, again varying the value of π_1.

We sequentially test H_0^i versus H_1^i for $i = 1, \ldots, k$. We use $n = 1000$ as the size of the database D, and $k = 800$ as the number of features as well as the number of hypotheses. While there is no closed form to compute the p-values, the sum of $n = 1000$ i.i.d. samples is approximately normally distributed by the Central Limit Theorem. The expectation and the variance of ξ_j^i with $b = 1$ are $\mathbb{E}\left[\xi_j^i\right] = \frac{1}{\theta_i} + \frac{1}{1-\exp(\theta_i)}$ and $\text{Var}[\xi_j^i] = \frac{1}{\theta_i^2} - \frac{\exp(\theta_i)}{(\exp(\theta_i)-1)^2}$, respectively. Therefore, T_i is approximately distributed as $\mathcal{N}(n\mathbb{E}\left[\xi_j^i\right], n\text{Var}[\xi_j^i])$, and we compute the p-values accordingly. We run the experiments with shift $A = \frac{c\eta}{\varepsilon} \log \frac{2}{3\min\{\delta, 1-((1-\delta)/\exp(\varepsilon))^{1/k}\}}$ (shift magnitude $s = 1$). The results are shown in Figure 3, which plots the FDR and statistical power against the expected fraction of non-nulls, π_1.

As in the case with binomial data, we see that the performance of PAPRIKA generally diminishes as ε decreases, and that PAPRIKA AI outperforms PAPRIKA, again reinforcing the need for tuning the parameters λ_j based on the alpha-investing rule. All methods perform well in this setting, and the FDR of PAPRIKA AI is visually indistinguishable from 0 at all levels of ε and π_1 tested. Numerical values are listed in Table 2 in Appendix C for ease of comparison.

Additionally in Appendix C: we plot the rejection threshold α_t and wealth of all methods over time in Figure 4, and find that our private algorithms are consistent with the rejections of the non-private algorithms, another perspective which empirically confirms their accuracy. We also vary θ_i, which parameterizes the strength of the signal between the null and alternative hypotheses. We also vary the signal in the alternative hypotheses and Figure 5 shows that performance begins to decline with a weaker signal. Finally, in Appendix C.1 we discuss how the shift parameter A should be chosen to balance the tradeoff between FDR and statistical power, displaying results in Figure 7.



(a) $\varepsilon = 5$

(b) $\varepsilon = 5$

(c) PAPRIKA AI

(d) PAPRIKA AI

(e) PAPRIKA

(f) PAPRIKA

Figure 3. FDR and statistical power versus fraction of non-nulls π_1 for PAPRIKA (with $\lambda_j = 0.2$), PAPRIKA AI (with $\lambda_j = \alpha_j$), and non-private algorithms when the database consists of truncated exponential observations.

Acknowledgements

W.Z. supported in part by a Mozilla Research Grant, NSF grant CNS-1850187, and an ARC-TRIAD Fellowship from the Georgia Institute of Technology. G.K. supported by a University of Waterloo startup grant and a NSERC Discovery grant. R.C. supported in part by a Mozilla Research Grant, a Google Research Fellowship, NSF grants CNS-1850187 and CNS-1942772 (CAREER), and a JPMorgan Chase Faculty Research Award. Most of this work was completed while R.C. was at Georgia Institute of Technology. This work was initiated while all authors were visiting the Simons Institute for the Theory of Computing.

References

Aharoni, E. and Rosset, S. Generalized α-investing: definitions, optimality results and application to public databases. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):771–794, 2014.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Dajani, A. N., Lauger, A. D., Singer, P. E., Kifer, D., Reiter, J. P., Machanavajjhala, A., Garfinkel, S. L., Dahl, S. A., Graham, M., Karwa, V., Kim, H., Lelerc, P., Schmutte, I. M., Sexton, W. N., Vilhuber, L., and Abowd, J. M. The modernization of statistical disclosure limitation at the U.S. census bureau, 2017. Presented at the September 2017 meeting of the Census Scientific Advisory Committee.

Differential Privacy Team, Apple. Learning with privacy at scale. https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf, December 2017.

Ding, B., Kulkarni, J., and Yekhanin, S. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pp. 3571–3580. Curran Associates, Inc., 2017.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pp. 265–284, 2006.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pp. 715–724, 2010.

Dwork, C., Su, W. J., and Zhang, L. Differentially private false discovery rate control. *arXiv preprint arXiv:1807.04209*, 2018.

Erlingsson, Ú., Pihur, V., and Korolova, A. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pp. 1054–1067, New York, NY, USA, 2014. ACM.

Foster, D. P. and Stine, R. A. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):429–444, 2008.

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., and Craig, D. W. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

Javanmard, A. and Montanari, A. On online control of false discovery rate. *arXiv preprint arXiv:1502.06197*, 2015.

Javanmard, A. and Montanari, A. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of Statistics*, 46(2):526–554, 2018.

National Academies, S. Reproducibility and replicability in science. 2019.

Ramdas, A., Yang, F., Wainwright, M. J., and Jordan, M. I. Online control of the false discovery rate with decaying memory. In *Advances In Neural Information Processing Systems*, pp. 5650–5659, 2017.

Ramdas, A., Zrnic, T., Wainwright, M., and Jordan, M. SAFFRON: an adaptive algorithm for online control of the false discovery rate. In *International Conference on Machine Learning*, pp. 4286–4294, 2018.

Tian, J. and Ramdas, A. ADDIS: An adaptive discarding algorithm for online FDR control with conservative nulls. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pp. 9383–9391. Curran Associates, Inc., 2019.