

PAPRIKA: Private Online False Discovery Rate Control

Wanrong Zhang*

Gautam Kamath^{†‡}

Rachel Cummings^{*‡}

February 28, 2020

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 [oSEM19]. 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 problems, 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

*H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. {wanrongz, rachelc}@gatech.edu. 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. R.C. supported in part by a Mozilla Research Grant, a Google Research Fellowship, and NSF grant CNS-1850187. This work was initiated while both authors were visiting the Simons Institute for the Theory of Computing.

[†]Cheriton School of Computer Science, University of Waterloo. g@csail.mit.edu. Supported by a University of Waterloo startup grant. This work was initiated while the author was a Microsoft Research Fellow at the Simons Institute for the Theory of Computing at UC Berkeley.

[‡]Indicates equal contribution as last author.

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 called a *false discovery* if the null hypothesis is actually true. When testing multiple hypotheses, the probability of a false discovery due to noise increases as more tests are performed. The problem of *false discovery rate 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 (or fraction) of false discoveries, while maintaining high true positive rate (i.e., correct discoveries).

In many applications, the dataset may contain sensitive personal information, and the hypothesis testing procedure 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. Previous work has shown that the statistical analysis of these datasets can lead to privacy concerns, and it is possible to identify an individual’s genotype when only minor allele frequencies are revealed [HSR⁺08]. The field of *differential privacy* [DMNS06] offers data analysis tools that provide powerful worst-case privacy guarantees, and has become a de facto gold standard in privacy-preserving 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 parameterized privacy notion 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 [EPK14], Apple [Dif17], Microsoft [DKY17], and the U.S. Census Bureau [DLS⁺17].

1.1 Related Work

The only prior work on differentially private false discovery rate control (FDR) [DSZ18] considers the traditional offline multiple testing problem, where an analyst has all the hypotheses and corresponding *p-values* upfront. Their private procedure repeatedly applies the REPORTNOISYMIN mechanism [DR14] to the celebrated Benjamini-Hochberg (BH) procedure [BH95] 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 [DSZ18] 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 [DSZ18] cannot be applied to the online setting.

The online multiple hypothesis testing problem was first investigated by [FS08], who proposed a framework known as *online alpha-investing procedure* that models the hypothesis testing problem as an investment problem. Some extensions based on this framework include generalized alpha-investing (GAI) [AR14], Level based On Recent Discovery (LORD) [JM15, JM18], and the current state-of-the-art for online FDR control, SAFFRON [RZWJ18] and ADDIS [TR19]. Further discussion of these approaches appears in Section 2.2.

1.2 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 3), is presented in Section 3. Its privacy and accuracy guarantees are stated in Theorem 4 and 5, 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.

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 [DMNS06]). *An algorithm $\mathcal{M} : \mathcal{D}^n \rightarrow \mathcal{R}$ is (ϵ, δ) -differentially private if for every pair of neighboring databases $D, D' \in \mathcal{D}^n$, and for every subset of possible outputs $\mathcal{S} \subseteq \mathcal{R}$, $\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta$. If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.*

The *additive sensitivity* of a real-valued query $f : \mathcal{D}^n \rightarrow \mathbb{R}$ 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,

$$\Delta f = \max_{D, D' \text{ neighbors}} |f(D) - f(D')|.$$

If f is a vector-valued query, the expression above can be modified with the appropriate norm in place of the absolute value. Differential privacy guarantees are often achieved by adding *Laplace noise* at various places in the computation, where the noise scales with $\Delta f/\epsilon$. A Laplace random variable with parameter b is denoted $\text{Lap}(b)$, and has probability density function,

$$p_{\text{Lap}(b)}(x) = \frac{1}{2b} \exp\left(\frac{-|x|}{b}\right) \quad \forall x \in \mathbb{R}.$$

We may sometimes abuse notation and also use $\text{Lap}(b)$ to denote the realization of a random variable with this distribution.

The SPARSEVECTOR algorithm, first introduced by [DNPR10] and refined to its current form by [DR14], 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. We use this algorithm 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.

Algorithm 1 Sparse Vector: SPARSEVECTOR($D, \Delta, \{f_1, f_2, \dots\}, T, c, \varepsilon$)

Input: database D , stream of queries $\{f_1, f_2, \dots\}$ each with sensitivity Δ , threshold T , a cutoff point c , privacy parameter ε
Let $\hat{T}_0 = T + \text{Lap}(\frac{2\Delta c}{\varepsilon})$
Let count = 0
for each query i **do**
 Let $Z_i \sim \text{Lap}(\frac{4\Delta c}{\varepsilon})$
 if $f_i(X) + Z_i > \hat{T}$ **then**
 Output $a_i = \top$
 Let count = count + 1
 Let $\hat{T}_{\text{count}} = T + \text{Lap}(\frac{2\Delta c}{\varepsilon})$
 else
 Output $a_i = \perp$
 end if
 if count $\geq c$ **then**
 Halt.
 end if
end for

Theorem 1 ([DNPR10]). SPARSEVECTOR is $(\varepsilon, 0)$ -differentially private.

Theorem 2 ([DNPR10]). For any sequence of k queries f_1, \dots, f_k with sensitivity Δ such that $|\{i : f_i(D) \geq T - \alpha\}| \leq c$, SPARSEVECTOR outputs with probability at least $1 - \beta$ a stream of $a_1, \dots, a_k \in \{\top, \perp\}$ such that $a_i = \perp$ 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, [DSZ18] 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. Using this alternative sensitivity notion means that preserving privacy for these p -values only requires a small amount of noise.

Definition 2 (Multiplicative Sensitivity [DSZ18]). 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

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

Specifically, when μ is sufficiently small, then we can treat the logarithm of the p -values as having additive sensitivity η , and we only need to add noise that scales with η/ε , which may be much smaller than the noise required under the standard additive sensitivity notion.

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:

$$\text{FDR} = \mathbb{E}[\text{FDP}] = \mathbb{E}\left[\frac{|\mathcal{H}^0 \cap \mathcal{R}|}{|\mathcal{R}|}\right],$$

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 [FS08], 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 analysts 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 $\text{mFDR}(t) = \frac{\mathbb{E}[|\mathcal{H}^0 \cap \mathcal{R}|]}{\mathbb{E}[|\mathcal{R}|]}$. This approach was later extended to a class of generalized alpha-investing (GAI) rules [AR14]. One subclass of GAI rules, the Level based On Recent Discovery (LORD), was shown to have consistently good performance in practice [JM15, JM18]. The SAFFRON procedure, proposed by [RZWJ18], further improves the LORD procedures by adaptively estimating the proportion of true nulls. The SAFFRON procedure 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

$$\widehat{\text{FDP}}_{\text{SAFFRON}}(t) = \frac{\sum_{j \leq t} \alpha_j \frac{I(p_j > \lambda_j)}{1 - \lambda_j}}{|\mathcal{R}(t)|}, \quad (1)$$

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 is a much better estimator than LORD's naive estimator $\sum_{j \leq t} \alpha_j / |\mathcal{R}(t)|$. The SAFFRON estimator is a fairly tight estimate of $\text{FDP}^*(t)$, since intuitively $I(p_j > \lambda_j) / (1 - \lambda_j)$ has unit expectation under null hypotheses and is stochastically smaller than uniform under non-null hypotheses.

The SAFFRON algorithm is given formally in Algorithm 2. SAFFRON starts off with an error budget $(1 - \lambda_1)W_0 < (1 - \lambda_1)\alpha$, which will be allocated to different tests over time. It never loses wealth when testing candidate p -values with $p_j < \lambda_j$, and it earns back wealth of $(1 - \lambda_j)\alpha$ on every rejection except for the first. By construction, the SAFFRON algorithm controls $\widehat{\text{FDP}}_{\text{SAFFRON}}(t)$ to be less than α at any time t . The function g_t for defining the sequence $\{\lambda_j\}_{j=1}^\infty$ can be any coordinatewise non-decreasing function. 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 $p_j \leq \alpha_j$. It also takes in a non-increasing sequence of decay factors γ_j which sum to one. These decay factors serve to depreciate past wealth and ensure that the sum of the wealth budget is always below the desired level α .

Algorithm 2 SAFFRON : SAFFRON($\alpha, W_0, \{\gamma_j\}_{j=0}^\infty$)

Input: stream of p -values $\{p_1, p_2, \dots\}$, target FDR level α , initial wealth $W_0 < \alpha$, positive non-increasing sequence $\{\gamma_j\}_{j=0}^\infty$ of summing to one.

for each p -value p_t **do**

Set $\lambda_t = g_t(R_{1:t-1}, C_{1:t-1})$

Set the indicator for candidacy $C_t = I(p_t < \lambda_t)$. Set the candidates after the j -th rejection as $C_{j+} = \sum_{i=\tau_j+1}^{t-1} C_i$

if $t=1$ **then**

Set $\alpha_1 = (1 - \lambda_1)\gamma_1 W_0$

else

Compute $\alpha_t = (1 - \lambda_t)(W_0\gamma_{t-C_{0+}} + (\alpha - W_0)\gamma_{t-\tau_1-C_{1+}} + \sum_{j \geq 2} \alpha\gamma_{t-\tau_j-C_{j+}})$

end if

Output $R_t = I(p_t \leq \alpha_t)$

end for

The SAFFRON algorithm requires that the input sequence of p -values are not too correlated under the null hypothesis. This condition is formalized through a *filtration* on the sequence of candidacy and rejection decisions. Intuitively, this means that the sequence of hypotheses cannot be too adaptively chosen, otherwise the p -values may become overly correlated and violate this condition. Denote by $R_j := I(p_j \leq \alpha_j)$ the indicator for rejection, and let $C_j := I(p_j \leq \lambda_j)$ be the indicator for candidacy. Define the filtration formed by the sequences of σ -fields $\mathcal{F}^t := \sigma(R_1, \dots, R_t, C_1, \dots, C_t)$, and let $\alpha_t := f_t(R_1, \dots, R_{t-1}, C_1, \dots, C_{t-1})$, where f_t is an arbitrary function of the first $t-1$ indicators for rejections and candidacy. We say that the null p -values are conditionally super-uniformly distributed with respect to the filtration \mathcal{F} if:

$$\text{If null hypothesis } H_i \text{ is true, then } \Pr(p_t \leq \alpha_t | \mathcal{F}^{t-1}) \leq \alpha_t. \quad (2)$$

We note that independent p -values is a special case of the conditional super-uniformity condition of (2). When p -values are independent, they satisfy the following condition:

$$\text{If the null hypothesis } H_i \text{ is true, then } \Pr(p_t \leq u) \leq u \text{ for all } u \in [0, 1].$$

SAFFRON provides the following accuracy guarantees, where the first two conditions apply if p -values are conditionally super-uniformly distributed, and the last two conditions apply if the p -values are additionally independent under the null.

Theorem 3 ([RZWJ18]). *If the null p -values are conditionally super-uniformly distributed, then we have:*

- (a) $\mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > \lambda_j)}{1 - \lambda_j} \right] \geq \mathbb{E} [|\mathcal{H}^0 \cap \mathcal{R}(t)|];$
- (b) *The condition $\widehat{FDP}_{SAFFRON}(t)$ 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 $R_1, \dots, R_{t-1}, C_1, \dots, C_{t-1}$, then

- (c) $\mathbb{E} [\widehat{FDP}_{SAFFRON}(t)] \geq \mathbb{E} [FDP(t)] := FDR(t)$ for all $t \in \mathbb{N}$;
- (d) *The condition $\widehat{FDP}_{SAFFRON}(t)$ 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, Private Alpha-investing P-value Rejecting Iterative sparse vector Algorithm (PAPRIKA), given formally in Algorithm 3. It is a differentially private version of SAFFRON, where we use SPARSEVECTOR to ensure privacy of our rejection set. However, the combination of these tools is far from immediate for several reasons. Although the complete

proofs of our privacy and accuracy results appear in the appendix, we elaborate here on the algorithmic details and modifications needed to ensure privacy and FDR control.

Specifically, the SAFFRON algorithm decides 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. Our privacy analysis of the algorithm accounts for this change and shows that dynamic thresholds do not affect the privacy guarantees of SPARSEVECTOR.

Similar to prior work on private offline FDR control [DSZ18], we use the notion of *multiplicative sensitivity* described in Definition 2, because p -values may have high sensitivity and therefore require unacceptably large noise to be added to preserve privacy. We assume that our input stream of p -values p_1, p_2, \dots , 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.

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 to satisfy (ϵ, δ) -differential privacy, but the choice can be seen as inspired by the α_{SV} -accuracy term of SPARSEVECTOR as given in Theorem 2. 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 4 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 bounds of Theorem 4 correspond to $s = 4$, but in many scenarios choosing a smaller value of $s = 1$ or 2 will lead to better performance while still satisfy the privacy guarantee. Further analysis of how to choose this shift parameter is given in Section 4.3.

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 pre-processing 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 $\widehat{\text{FDP}}_{\text{SAFFRON}}(t)$ as defined in Equation (1), which serves as a conservative overestimate of $\text{FDP}(t)$. (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 $(\epsilon, 0)$ -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 differentially private.

Since we cannot add noise to the candidacy condition, in PAPRIKA we instead make our candidacy condition even weaker, to be $\log p_t < \log 2\lambda_t$. 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 such that $\log p_t < \log 2\lambda_t < \log p'_t$. 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 $\eta = \text{poly}^{-1}(n)$.¹ Our PAPRIKA algorithm is thus (ε, δ) -differentially private, and we account for this failure probability in our (exponentially small) δ parameter, as stated in Theorem 4.

Our algorithm also controls at each time t ,

$$\widehat{\text{FDP}}_{\text{PAPRIKA}}(t) \leq \frac{\sum_{j \leq t} \alpha_t \frac{I(p_j > 2\lambda_j)}{1-2\lambda_j}}{|\mathcal{R}(t)|}. \quad (3)$$

We note that this is equivalent to $\widehat{\text{FDP}}_{\text{SAFFRON}}(t)$ in Equation (1) 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 5.

Algorithm 3 Private Alpha-investing P-value Rejecting Iterative sparse veKtor Algorithm (PAPRIKA) :
PAPRIKA($\alpha, \lambda, W_0, \{\gamma_j\}_{j=0}^\infty, c, \varepsilon, \delta, s$)

Input: stream of p -values $\{p_1, p_2, \dots\}$ with multiplicative sensitivity (η, μ) , target FDR level α , 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 ε, δ , threshold shift magnitude s .

Let $Z_\alpha^0 \sim \text{Lap}(2\eta c/\varepsilon)$, $A = \frac{sc\eta}{\varepsilon} \log \frac{2}{3 \min\{\delta, 1 - ((1-\delta)/\exp(\varepsilon))^{1/k}\}}$, count = 0

for each p -value p_t **do**

 Let $Z_t \sim \text{Lap}(4\eta c/\varepsilon)$

 Set $\lambda_t = g_t(R_{1:t-1}, C_{1:t-1})$

 Set the indicator for candidacy $C_t = I(\log p_t < \log 2\lambda_t)$. Set the candidates after the j -th rejection as

$C_{j+} = \sum_{i=\tau_j+1}^{t-1} C_i$

if $t = 1$ **then**

 Set $\alpha_1 = (1 - 2\lambda_1)\gamma_1 W_0$

else

 Compute $\alpha_t = (1 - 2\lambda_t)(W_0\gamma_{t-C_{0+}} + (\alpha - W_0)\gamma_{t-\tau_1-C_{1+}} + \sum_{j \geq 2} \alpha\gamma_{t-\tau_j-C_{j+}})$

end if

if $C_t = 1$ and $\log p_t + Z_t \leq \log \alpha_t - A + Z_\alpha^{\text{count}}$ **then**

 Output $R_t = 1$

 count = count + 1

$Z_\alpha^{\text{count}} \sim \text{Lap}(2\eta c/\varepsilon)$

else

 Output $R_t = 0$

end if

if count $\geq c$ **then**

 Halt

end if

end for

Theorem 4. For any stream of p -values $\{p_1, p_2, \dots\}$, PAPRIKA($\alpha, \lambda, W_0, \{\gamma_j\}_{j=0}^\infty, c, \varepsilon, \delta, s$) 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 4 appears in Appendix A.

¹Such 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.

Next we describe the theoretical guarantees of FDR control for our private algorithm PAPRIKA which is an analog of Theorem 3. We need to slightly modify the conditional super-uniformity assumption given in (2) to incorporate the added Laplace noise. 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 $\alpha_t := f_t(R_1, \dots, R_{t-1}, C_1, \dots, C_{t-1})$, 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 $\mathcal{F}'^t := \sigma(R_1, \dots, R_t, C_1, \dots, C_t, Z_1, \dots, Z_t, Z_\alpha)$. We say that the null p -values are conditionally super-uniformly distributed with respect to the filtration \mathcal{F}' if:

$$\text{If the null hypothesis } H_i \text{ is true, then } \Pr(p_t \leq \alpha_t | \mathcal{F}'^{t-1}) \leq \alpha_t. \quad (4)$$

Our FDR control guarantees for PAPRIKA mirror those of SAFFRON (Theorem 3). The first two conditions apply if p -values are conditionally super-uniformly distributed, and the last two conditions apply if the p -values are additionally independent under the null.

Theorem 5. *If the null p -values are conditionally super-uniformly distributed, then we have:*

- (a) $\mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \right] \geq \mathbb{E} [|\mathcal{H}^0 \cap \mathcal{R}(t)|];$
- (b) *The condition $\widehat{FDP}_{\text{PAPRIKA}}(t) \leq \alpha$ for all $t \in \mathbb{N}$ implies that $\text{mFDR}(t) \leq \alpha + \delta t$ 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 $R_1, \dots, R_{t-1}, C_1, \dots, C_{t-1}$, then

- (c) $\mathbb{E} [\widehat{FDP}_{\text{PAPRIKA}}(t)] \geq \mathbb{E} [FDP(t)] := FDR(t)$ 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 $t \in \mathbb{N}$.*

This statement can be compared with the guarantees of SAFFRON (Theorem 3). Similar to before, the first two conditions provide a bound on mFDR, whereas the latter two conditions bound FDR when we have independence of the p -values. In contrast to the non-private guarantees, we have a slack term of δt in the FDR bound. However, since δ will generally be cryptographically small in most applications, this will be have a negligible effect on the FDR. The proof of Theorem 5 appears in Appendix B.

The following lemma is a key tool in the proof of Theorem 5. Though it is qualitatively similar to Lemma 2 in [RZWJ18], it is crucially modified to show an analogous statement holds under the addition of Laplace noise. Its proof appears in Section C.

Lemma 1. *Assume p_1, p_2, \dots are independent of each other and let $h : \{0, 1\}^k \rightarrow \mathbb{R}$ to 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

$$\mathbb{E} \left[\frac{\min\{\alpha_t \exp(Z_\alpha - Z_t - A), 1\}}{h(R_{1:k})} | \mathcal{F}'^{t-1} \right] \geq \mathbb{E} \left[\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(R_{1:k})} | \mathcal{F}'^{t-1} \right].$$

4 Experiments

In this section, we provide experimental results that compare the performance of variations of the PAPRIKA and SAFFRON procedures. In particular, we evaluate the FDR and the statistical power of each algorithm under two different sequences of $\{\gamma_j\}$: one uses a constant sequence $\gamma_j = 0.2$, and the other one sets $\gamma_j = \alpha_j$. We refer to the latter case with an AI suffix (PAPRIKA AI and SAFFRON AI, respectively) to indicate that the choice of γ_j corresponds to the Alpha Investing (AI) rule. We generally observe that, even under moderately stringent privacy restrictions, PAPRIKA's performance is comparable to that of the non-private alternatives. Throughout our experiments, we set $\delta = 0.2/800 = 2.5 \times 10^{-4}$, and all the results are averaged over 100 runs.

We investigate two settings: in Section 4.1, the observations come Bernoulli distributions, and in Section 4.2, the observations are generated from truncated exponential distributions. In Section 4.3, 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

In this setting, we assume that we have n individuals in a database D , and that each individual's data contains k independent entries. The i th entries are associated with n i.i.d. Bernoulli variables ξ_1^i, \dots, ξ_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 i th entries. 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,

$$p_i(D) = \sum_{k=t_i}^n \frac{1}{2^n} \binom{n}{k}.$$

[DSZ18] 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. We choose θ_i for our experiments as follows:

$$\theta_i = \begin{cases} 0.5 & \text{with probability } 1 - \pi_1 \\ 0.75 & \text{with probability } \pi_1, \end{cases}$$

where we vary the parameter π_1 , corresponding to the expected fraction of non-nulls.

In the following experiments, we sequentially test H_0^i versus H_1^i for $i = 1, \dots, k$. We use $n = 1000$ as the size of the database D , and $k = 800$ as the number of entries as well as the number of hypotheses. Our experiments are run under several different shifts A , but we only report results with $A = \frac{c\eta}{\varepsilon} \log \frac{2}{3 \min\{\delta, 1 - ((1-\delta)/\exp(\varepsilon))^{1/k}\}}$ (i.e., when $s = 1$). Further discussion on the choice of A is deferred to Section 4.3. The results are summarized in Figure 1, which plots the false discovery rate (FDR) against the expected fraction of non-nulls, π_1 . We evaluate the performance of PAPRIKA under two different sequences of $\{\gamma_j\}$: $\gamma_j = \alpha_j$ and $\gamma_j = 0.2$, denoted by PAPRIKA AI and PAPRIKA, respectively. The non-private baseline methods are LORD in [JM15, JM18], Alpha-investing in [AR14], and SAFFRON and SAFFRON AI in [RZWJ18]. In Figure 1(a) and (b), we compare our algorithm with the baseline methods with privacy parameter $\varepsilon = 5$. In Figure 1(c,d) and (e,f), we compare the performance of PAPRIKA AI and PAPRIKA, respectively, with varying privacy parameters $\varepsilon = 3, 5, 10$.

As expected, the performance of PAPRIKA generally diminishes as ε decreases. A notable exception is that FDR 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 gives a lower FDR as compared with other 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 setting $\gamma_j = \alpha_j$ is a natural choice to ensure good performance in practice.

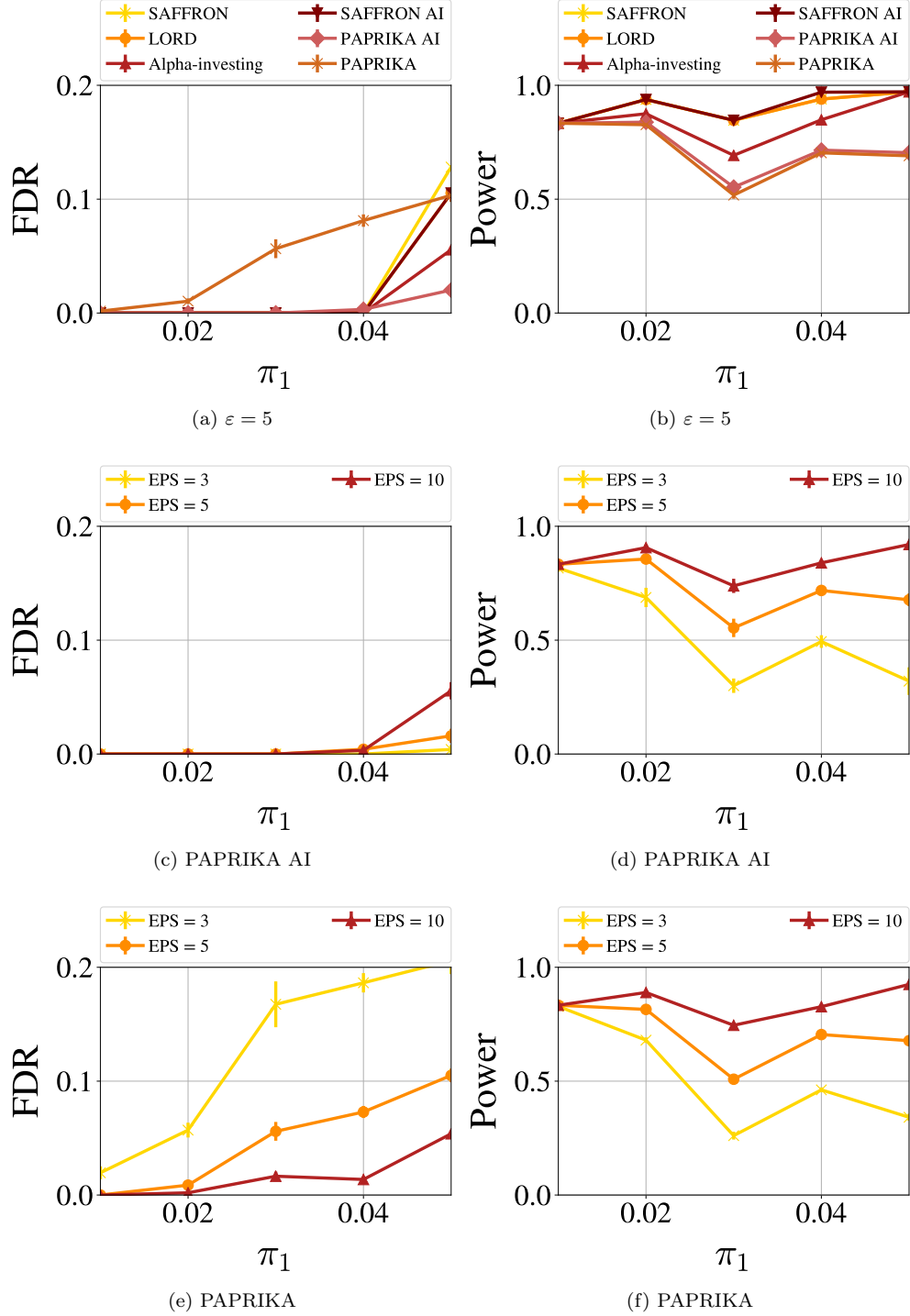


Figure 1: FDR and statistical power versus fraction of non-null hypotheses π_1 for PAPRIKA and non-private algorithms using two different sequences $\{\lambda_j\}$ when the database consists of Bernoulli observations. The first row plots the comparisons of FDR and power of PAPRIKA with privacy parameter $\varepsilon = 5$ with other non-private algorithms. The second row plots the FDR and power of PAPRIKA with $\lambda_j = \alpha_j$ (which we refer to as PAPRIKA AI) with privacy parameters $\varepsilon = 3, 5, 10$. The third row plots the FDR and power of PAPRIKA with $\lambda_j = 0.2$ with privacy parameters $\varepsilon = 3, 5, 10$.

4.2 Testing with Truncated Exponential Observations

In this section, we also assume that have n individuals in the database D . Each individual's data contains k independent entries. Here the i th entries are associated with n i.i.d. truncated exponential distributed variables ξ_1^i, \dots, ξ_n^i , each of which is sampled according to the density function,

$$f_i(x \mid \theta_i, b) = \frac{\theta_i \exp(-\theta_i x)}{1 - \exp(-b\theta_i)} I(0 \leq x \leq b),$$

for positive parameters b and θ_i . Let t_i be the realized sum of the i th entries, and let T_i denote the random variable of the sum i th entries, corresponding to the n truncated exponential distributed variables. 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,

$$p_i(D) = \Pr_{\theta_i=1}(T_i > t_i).$$

[DSZ18] 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 θ_i as follows:

$$\theta_i = \begin{cases} 1 & \text{with probability } 1 - \pi_1 \\ 1.95 & \text{with probability } \pi_1, \end{cases}$$

where we vary the parameter π_1 , corresponding to the expected fraction of non-nulls.

We sequentially test H_0^i versus H_1^i for $i = 1, \dots, k$. We use $n = 1000$ as the size of the database D , and $k = 800$ as the number of entries as well as the number of hypotheses. We note that there is no closed form to compute the p -values, however, 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

$$\begin{aligned} \mathbb{E}[\xi_j^i] &= \frac{1}{\theta_i} + \frac{1}{1 - \exp(-\theta_i)}, \text{ and} \\ \text{Var}[\xi_j^i] &= \frac{1}{\theta_i^2} - \frac{\exp(\theta_i)}{(\exp(\theta_i) - 1)^2}, \end{aligned}$$

respectively. Therefore, T_i is approximately distributed as $\mathcal{N}(n\mathbb{E}[\xi_j^i], n\text{Var}[\xi_j^i])$, and we compute the p -values accordingly. We run the experiments with shift $A = \frac{cn}{\epsilon} \log \frac{2}{3 \min\{\delta, 1 - ((1-\delta)/\exp(\epsilon))^{1/k}\}}$ (shift magnitude $s = 1$). The results are summarized in Figure 2.

All of the methods perform well in this setting. To illustrate the process, we also plot the rejection threshold α_t and wealth versus the hypothesis index in Figure 3. Each “jump” of the wealth corresponds to a rejection. We observe that the rejections of our private algorithms are consistent with the rejections of the non-private algorithms, another perspective which empirically confirms their accuracy.

One explanation for the good performance observed in Figure 2 could be that the signal between the null and alternative hypotheses as parameterized by θ_i could be very strong, which means the algorithms could easily discriminate between the true null and true non-null hypotheses based on the observed p -values. To measure this, we also varied the value of θ_i in the alternative hypotheses. These results are shown in Figure 4, which plots FDR and power of PAPRIKA and PAPRIKA AI with when the alternative hypotheses have parameter $\theta_i = 1.80, 1.90, 2.00$. As expected, the performance gets better as we increase the signal, and we observe that when the signal is too weak ($\theta_i = 1.80$), performance begins to decline.

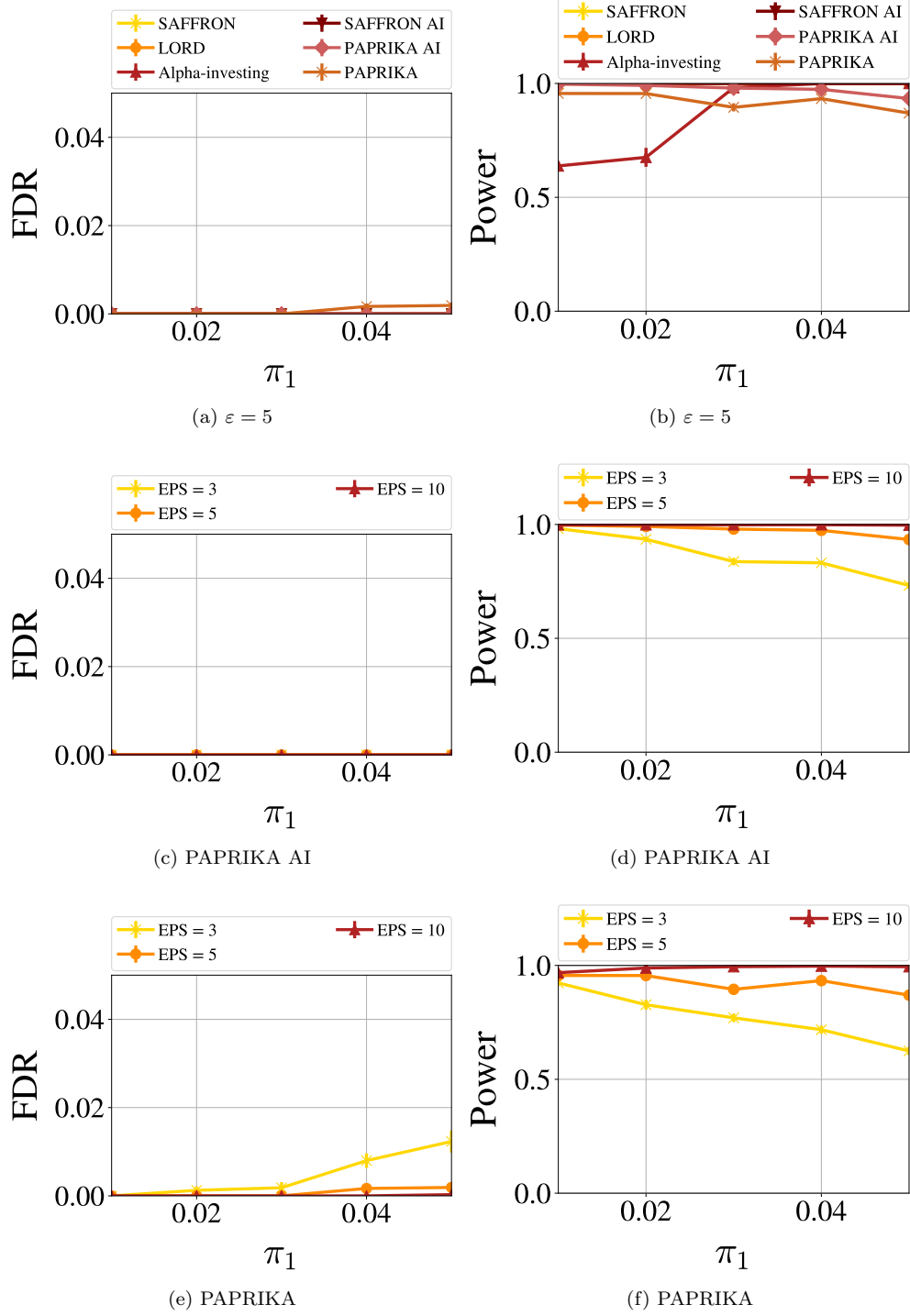


Figure 2: FDR and statistical power versus fraction of non-null hypotheses π_1 for PAPRIKA and non-private algorithms using two different sequences $\{\lambda_j\}$ when the database consists of truncated exponential observations. The first row plots the comparisons of FDR and power of PAPRIKA with privacy parameter $\varepsilon = 5$ with other non-private algorithms. The second row plots the FDR and power of PAPRIKA AI with $\lambda_j = \alpha_j$ under privacy parameters $\varepsilon = 3, 5, 10$. The third row plots the FDR and power of PAPRIKA with $\lambda_j = 0.2$ under privacy parameters $\varepsilon = 3, 5, 10$.

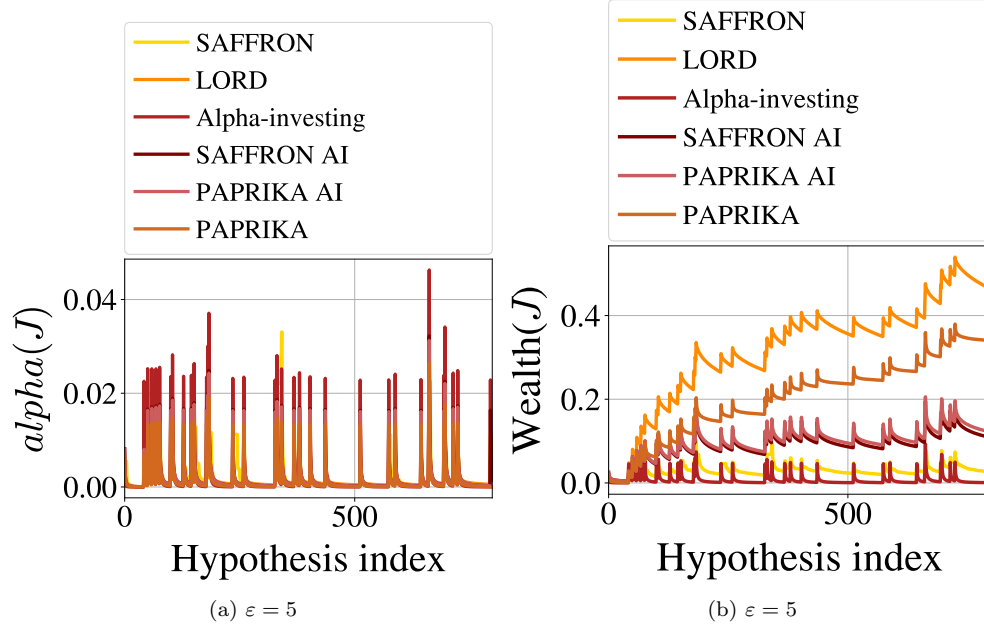


Figure 3: Wealth and rejection threshold α_t versus hypothesis index with privacy parameter $\varepsilon = 5$ when the database consists of truncated exponential observations. PAPRIKA AI and SAFFRON AI used $\lambda_j = \alpha_j$, PAPRIKA used $\lambda_j = 0.2$, and SAFFRON used $\lambda_j = 0.5$.

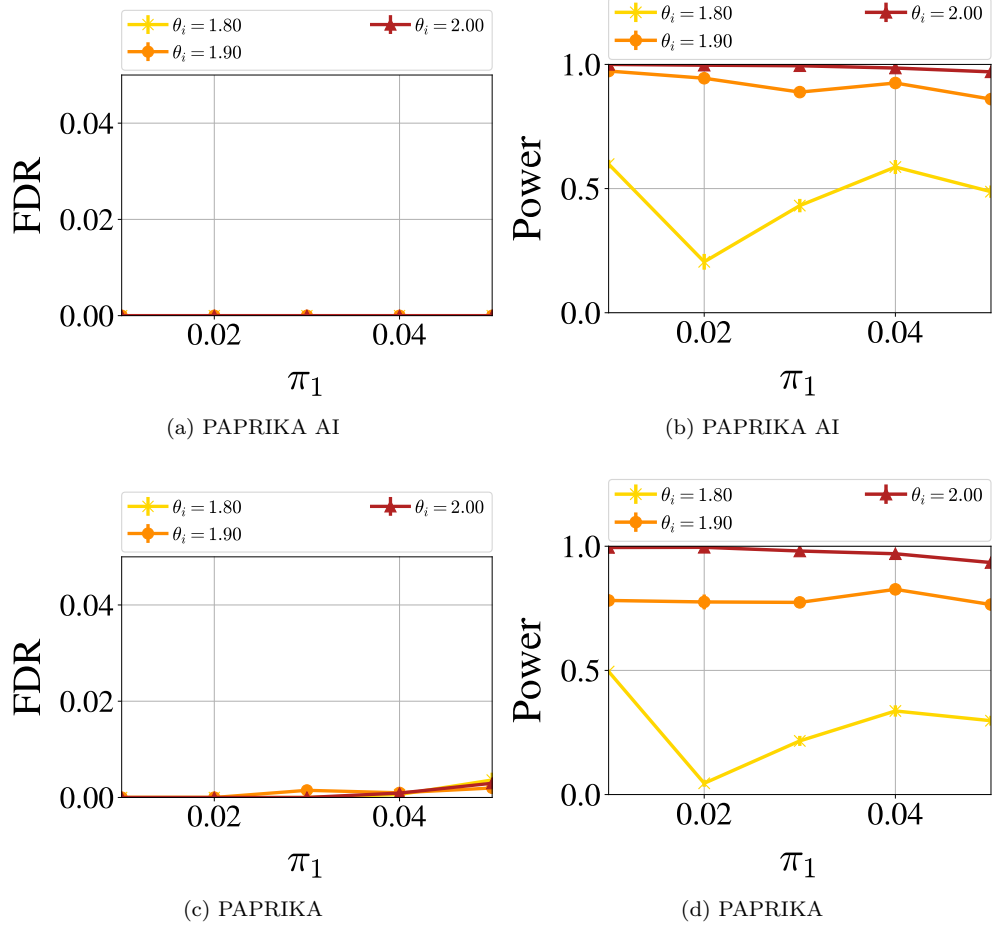


Figure 4: FDR and statistical power versus expected fraction of non-null hypotheses π_1 under various choices of signal $\theta_i = 1.80, 1.90, 2.00$ for alternative hypothesis parameters. The privacy parameter is $\varepsilon = 5$, and the database consists of truncated exponential observations. The first row shows performance of PAPRIKA AI where $\lambda_j = \alpha_j$, and the second row shows performance of PAPRIKA where $\lambda_j = 0.2$.

4.3 Choice of shift A

We now discuss how to choose the shift parameter A . Theorem 4 gives a theoretical lower bound for A in terms of the privacy parameter δ , but this bound may be overly conservative. Since the shift A is closely related to the performance of FDR and statistical power, we wish to pick a value of A that yields good performance in practice. In Theorem 5, we show that $\text{FDR}(t)$ is less than our desired bound α plus the privacy parameter δt , which naturally requires that the privacy loss parameter δ be small.

We use the Bernoulli example in Section 4.1 to investigate the performance under different choices of the shift A with privacy parameter $\varepsilon = 5$. The results are summarized in Figure 5, which plots the FDR and power versus the expected fraction of non-nulls when we vary the shift size with $s = 0.5, 1, 1.5, 2$.

Larger shifts (corresponding to larger values of s) will lower the rejection threshold, which causes fewer hypotheses to be rejected. This improves FDR of the algorithm, but harms Power, as the threshold may be too low to reject true nulls. Figure 5 shows that the shift size parameter s should be chosen by the analyst to balance the tradeoff between FDR and Power, as demanded by the application.

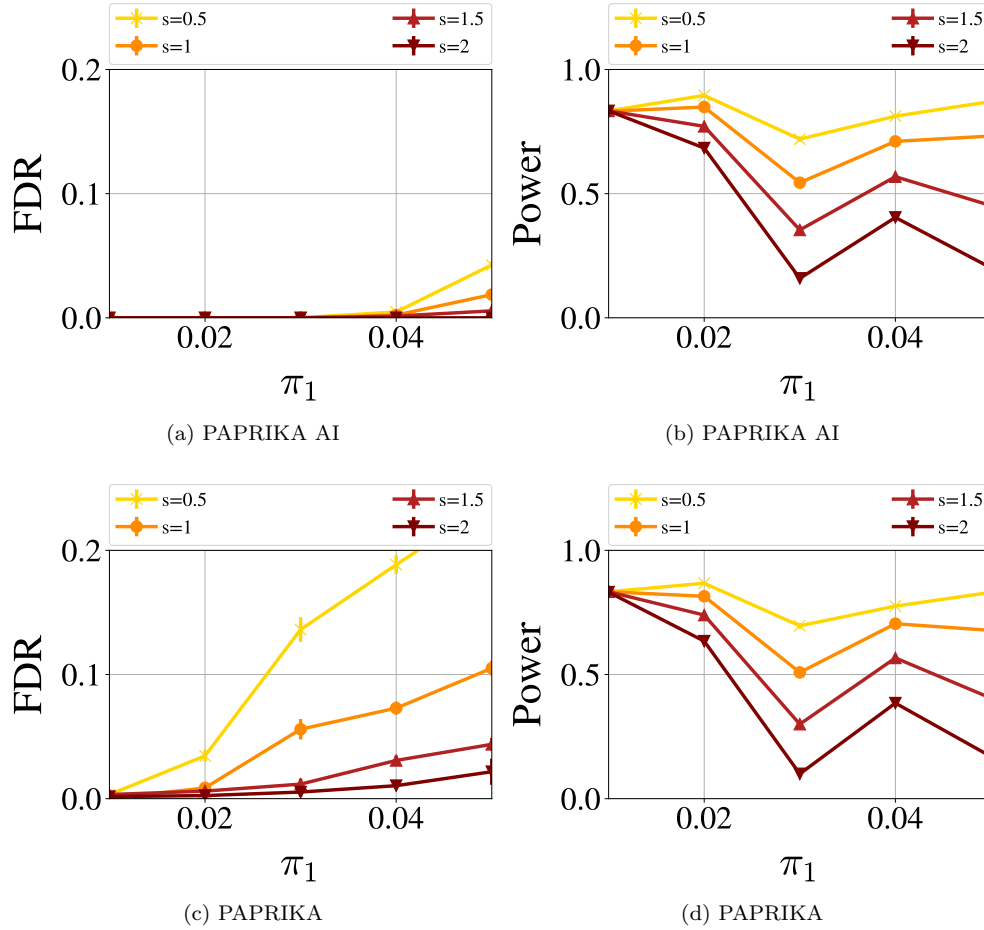


Figure 5: FDR and statistical power versus expected fraction of non-null hypotheses π_1 under various choices of shift magnitude s . The privacy parameter is $\varepsilon = 5$, and the database consists of Bernoulli observations. The first row shows performance of PAPRIKA AI where $\lambda_j = \alpha_j$, and the second row shows performance of PAPRIKA where $\lambda_j = 0.2$.

References

- [AR14] Ehud Aharoni and Saharon Rosset. 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.
- [BH95] Yoav Benjamini and Yosef Hochberg. 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.
- [Dif17] Differential Privacy Team, Apple. Learning with privacy at scale. <https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appliedifferentialprivacysystem.pdf>, December 2017.
- [DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, NIPS ’17, pages 3571–3580. Curran Associates, Inc., 2017.

- [DLS⁺17] Aref N. Dajani, Amy D. Lauger, Phyllis E. Singer, Daniel Kifer, Jerome P. Reiter, Ashwin Machanavajjhala, Simson L. Garfinkel, Scot A. Dahl, Matthew Graham, Vishesh Karwa, Hang Kim, Philip Lelerc, Ian M. Schmutte, William N. Sexton, Lars Vilhuber, and John M. Abowd. 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.
- [DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, 2006.
- [DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, 2010.
- [DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- [DSZ18] Cynthia Dwork, Weijie J Su, and Li Zhang. Differentially private false discovery rate control. *arXiv preprint arXiv:1807.04209*, 2018.
- [EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.
- [FS08] Dean P Foster and Robert A Stine. 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.
- [HSR⁺08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.
- [JM15] Adel Javanmard and Andrea Montanari. On online control of false discovery rate. *arXiv preprint arXiv:1502.06197*, 2015.
- [JM18] Adel Javanmard and Andrea Montanari. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of Statistics*, 46(2):526–554, 2018.
- [oSEM19] National Academies of Sciences Engineering and Medicine. Reproducibility and replicability in science. *Washington, DC: The National Academies Press*, 2019.
- [RZWJ18] Aaditya Ramdas, Tijana Zrnic, Martin Wainwright, and Michael Jordan. SAFFRON: an adaptive algorithm for online control of the false discovery rate. *arXiv preprint arXiv:1802.09098*, 2018.
- [TR19] Jinjin Tian and Aaditya Ramdas. ADDIS: An adaptive discarding algorithm for online FDR control with conservative nulls. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 9383–9391. Curran Associates, Inc., 2019.

A Proof of Theorem 4

Before proving Theorem 4, we will state and prove the following lemma, which will be useful in the proofs of Theorem 4 and Theorem 5.

Lemma 2. *If $Z_1 \sim \text{Lap}(2b)$, $Z_2 \sim \text{Lap}(b)$ and $C > 0$ is a constant, we have $\Pr(Z_1 \geq Z_2 - C) = 1 - \frac{2}{3} \exp(-\frac{C}{2b}) + \frac{1}{6} \exp(-C/b)$.*

Proof.

$$\begin{aligned}
\Pr(Z_1 \geq Z_2 - C) &= \int_{-\infty}^{\infty} \int_{x-C}^{\infty} \frac{1}{4b} \exp(-\frac{|y|}{2b}) \frac{1}{2b} \exp(-\frac{|x|}{b}) dy dx \\
&= \int_{-\infty}^C (1 - \frac{1}{2} \exp(-\frac{|x-C|}{2b})) \frac{1}{2b} \exp(-\frac{|x|}{b}) dx + \int_C^{\infty} \frac{1}{2} \exp(-\frac{|x-C|}{2b}) \frac{1}{2b} \exp(-\frac{|x|}{b}) dx \\
&= \int_{-\infty}^C \frac{1}{2b} \exp(-\frac{|x|}{b}) dx - \int_{-\infty}^0 \frac{1}{4b} \exp(-\frac{|3x-C|}{2b}) dx \\
&\quad - \int_0^C \frac{1}{4b} \exp(-\frac{C+x}{2b}) dx + \int_C^{\infty} \frac{1}{4b} \exp(-\frac{|3x-C|}{2b}) dx \\
&= 1 - \frac{1}{2} \exp(-\frac{C}{b}) - \frac{1}{6} \exp(-\frac{C}{2b}) - \frac{1}{2} \exp(-\frac{C}{2b}) + \frac{1}{2} \exp(-\frac{C}{b}) + \frac{1}{6} \exp(-\frac{C}{b}) \\
&= 1 - \frac{2}{3} \exp(-\frac{C}{2b}) + \frac{1}{6} \exp(-\frac{C}{b})
\end{aligned}$$

□

Theorem 4. For any stream of p -values $\{p_1, p_2, \dots\}$, $\text{PAPRIKA}(\alpha, \lambda, W_0, \{\gamma_j\}_{j=0}^{\infty}, c, \varepsilon, \delta, s)$ is (ε, δ) -differentially private.

Proof. Fix any two neighboring databases D and D' . Let R denote the random variable representing the output of $\text{PAPRIKA}(D, \alpha, \lambda, W_0, \{\gamma_j\}_{j=0}^{\infty}, c, \varepsilon, \delta, s)$ and let R' denote the random variable representing the output of $\text{PAPRIKA}(D', \alpha, \lambda, W_0, \{\gamma_j\}_{j=0}^{\infty}, c, \varepsilon, \delta, s)$. Let k denote the total number of hypotheses. When $\log p_t \geq \log 2\lambda$ and $\log p'_t \geq \log 2\lambda$ for all t , $\Pr(R = \{0, 0, \dots, 0\}) = 1 = \Pr(R' = \{0, 0, \dots, 0\})$. When $\log p_t < \log 2\lambda$ and $\log p'_t < \log 2\lambda$ for all t , privacy follows from the privacy of SPARSEVECTOR . For other cases, the worst case is that for all t , $\log p_t < \log 2\lambda$ and $\log p'_t \geq \log 2\lambda$. In this setting, we have

$$\Pr(R' = r) = \begin{cases} 1 & \text{if } r = \{0, 0, \dots, 0\} \\ 0 & \text{otherwise.} \end{cases}$$

To satisfy (ε, δ) -differential privacy, we need to bound the probability of outputting r for database D . We first consider $r = \{0, 0, \dots, 0\}$. We wish to bound $\Pr(R' = \{0, 0, \dots, 0\}) \leq \exp(\varepsilon) \Pr(R = \{0, 0, \dots, 0\}) + \delta$ and $\Pr(R = \{0, 0, \dots, 0\}) \leq \exp(\varepsilon) \Pr(R' = \{0, 0, \dots, 0\}) + \delta$. The latter is trivial since $\exp(\varepsilon) \Pr(R' = \{0, 0, \dots, 0\}) + \delta = \exp(\varepsilon) + \delta$, which is greater than 1. It remains to satisfy $\Pr(R' = \{0, 0, \dots, 0\}) \leq \exp(\varepsilon) \Pr(R = \{0, 0, \dots, 0\}) + \delta$, which is equivalent to $1 - \delta \leq \exp(\varepsilon) \Pr(R = \{0, 0, \dots, 0\})$. We have

$$\begin{aligned}
\Pr(R = \{0, 0, \dots, 0\}) &= \Pr(R_1 = 0) \Pr(R_2 = 0 | R_1 = 0) \dots \Pr(R_k = 0 | R_{k-1} = 0) \\
&= \prod_{t=1}^k \Pr(\log p_t + Z_t \geq \log \alpha_t - A + Z_{\alpha}) \\
&> \prod_{t=1}^k \Pr(\log 2\lambda - \eta + Z_t \geq \log \alpha_t - A + Z_{\alpha}) \tag{5}
\end{aligned}$$

$$\begin{aligned}
&= \prod_{t=1}^k \Pr(Z_t \geq Z_{\alpha} + \log \alpha_t - \log 2\lambda + \eta - A) \\
&= \prod_{t=1}^k \left(1 - \frac{2}{3} \exp(-\frac{\varepsilon(A + \log(2\lambda/\alpha_t) - \eta)}{4\eta c}) + \frac{1}{6} \exp(-\frac{\varepsilon(A + \log(2\lambda/\alpha_t) - \eta)}{2\eta c}) \right) \tag{6}
\end{aligned}$$

$$\geq \left(1 - \frac{2}{3} \exp(-\frac{\varepsilon(A + \log 2 - \eta)}{4\eta c}) \right)^k, \tag{7}$$

where Inequality (5) is because the worst case happens when p_t is η below the candidacy threshold $\log 2\lambda$, Equation (6) applies Lemma 2, and Inequality (7) follows from the facts that $\alpha_t \leq \lambda$ for all t and that the third term in (6) is positive. Setting (7) to be larger than $(1 - \delta)/\exp(\varepsilon)$, we have,

$$\frac{2}{3} \exp\left(-\frac{\varepsilon(A + \log 2 - \eta)}{4\eta c}\right) \leq 1 - \left(\frac{1 - \delta}{\exp(\varepsilon)}\right)^{\frac{1}{k}}. \quad (8)$$

Next, we consider all other possible outputs r . Define the set $S := \{r \mid \text{there exists a } t \text{ such that } r_t = 1\}$. We wish to bound $\Pr(R \in S) \leq \exp(\varepsilon) \Pr(R' \in S) + \delta$ and $\Pr(R' \in S) \leq \exp(\varepsilon) \Pr(R \in S) + \delta$. The latter is trivial since $\Pr(R' \in S) = 0$. It remains to bound $\Pr(R \in S) \leq \delta$. For any t , we have

$$\begin{aligned} \Pr(R \in S) &\leq \Pr(R_t = 1) \\ &= \Pr(\log p_t + Z_t \leq \log \alpha_t - A + Z_\alpha) \\ &\leq \Pr(\log 2\lambda + Z_t \leq \log \alpha_t - A + Z_\alpha) \end{aligned} \quad (9)$$

$$\begin{aligned} &= \Pr(Z_t \leq Z_\alpha - (\log(2\lambda/\alpha_t) + A)) \\ &\leq \Pr(Z_t \leq Z_\alpha - (\log 2 + A)) \\ &= \frac{2}{3} \exp\left(-\frac{\varepsilon(A + \log 2)}{4\eta c}\right) - \frac{1}{6} \exp\left(-\frac{\varepsilon(A + \log 2)}{2\eta c}\right) \end{aligned} \quad (10)$$

$$\leq \frac{2}{3} \exp\left(-\frac{\varepsilon(A + \log 2)}{4\eta c}\right), \quad (11)$$

where Inequality (9) is because the worst case occurs when $\log p_t = \log 2\lambda$, Equality (10) applies Lemma 2, and Inequality (11) follows from the facts that $\alpha_t \leq \lambda$ for all t and that the second term in (10) is negative. Setting (11) to be less than δ , we have,

$$\frac{2}{3} \exp\left(-\frac{\varepsilon(A + \log 2)}{4\eta c}\right) \leq \delta. \quad (12)$$

Combining Equations (12) and (8), we have the condition that $\frac{2}{3} \exp\left(-\frac{\varepsilon(A + \log 2 - \eta)}{4\eta c}\right) \leq \min\{\delta, 1 - ((1 - \delta)/\exp(\varepsilon))^{1/k}\}$.

Rearranging this inequality for A gives

$$A \geq \frac{4\eta c}{\varepsilon} \left(\log \frac{2}{3 \min\{\delta, 1 - ((1 - \delta)/\exp(\varepsilon))^{1/k}\}} - \log 2 + \eta \right),$$

which is how the shift term A is set in PAPRIKA. □

B Proof of Theorem 5

Theorem 5. *If the null p -values are conditionally super-uniformly distributed, then we have:*

(a) $\mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \right] \geq \mathbb{E} [|\mathcal{H}^0 \cap \mathcal{R}(t)|];$

(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 $R_1, \dots, R_{t-1}, C_1, \dots, C_{t-1}$, then

(c) $\mathbb{E} [\widehat{FDP}_{\text{PAPRIKA}}(t)] \geq \mathbb{E} [FDP(t)] := FDR(t)$ 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 $t \in \mathbb{N}$.*

Proof. For any time $t > 0$, we bound the number of false rejections as follows:

$$\begin{aligned}
\mathbb{E}[|\mathcal{H}^0 \cap \mathcal{R}(t)|] &= \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E}[I(\log p_j + Z_j \leq \log \alpha_j - A + Z_\alpha)] \\
&\leq \sum_{j \leq t, j \in \mathcal{H}^0} \Pr(\log p_j \leq \log \alpha_j) + \Pr(Z_j \leq Z_\alpha - A) \\
&\leq \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E}[\alpha_j] + \Pr(Z_j \leq Z_\alpha - A)
\end{aligned} \tag{13}$$

where Inequality (13) follows from the conditional super-uniformity property. We bound each term in (13) separately. Using the law of iterated expectations by conditioning on \mathcal{F}^{t-1} , we can bound the first term of (13) as follows:

$$\begin{aligned}
\sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E}[\alpha_j] &\leq \mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \mathbb{E} \left[\frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \middle| \mathcal{F}^{t-1} \right] \right] \\
&= \mathbb{E} \left[\mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \middle| \mathcal{F}^{t-1} \right] \right] \\
&= \mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \right],
\end{aligned} \tag{14}$$

where Equation (14) applies the conditional super-uniformity. Since $\widehat{\text{FDP}}_{\text{PAPRIKA}}(t) \leq \alpha$, we have,

$$\mathbb{E} \left[\sum_{j \leq t, j \in \mathcal{H}^0} \alpha_j \frac{I(p_j > 2\lambda_j)}{1 - 2\lambda_j} \right] \leq \alpha \mathbb{E}[|\mathcal{R}(t)|].$$

Next, we bound the second term in (13) as follows:

$$\begin{aligned}
\sum_{j \leq t, j \in \mathcal{H}^0} \Pr(Z_j \leq Z_\alpha - A) &\leq \frac{2t}{3} \exp\left(-\frac{A\varepsilon}{4\eta c}\right) - \frac{t}{6} \exp\left(-\frac{A\varepsilon}{2\eta c}\right) \\
&\leq t \min \left\{ \delta, 1 - \left(\frac{1 - \delta}{\exp(\varepsilon)} \right)^{\frac{1}{k}} \right\}.
\end{aligned}$$

Combining this inequality with (14), we bound mFDR as

$$\begin{aligned}
mFDR &:= \frac{\mathbb{E}[|\mathcal{H}^0 \cap \mathcal{R}(t)|]}{\mathbb{E}[|\mathcal{R}(t)|]} \\
&\leq \alpha + \frac{1}{\mathbb{E}[|\mathcal{R}(t)|]} \sum_{j \leq t, j \in \mathcal{H}^0} \Pr(Z_j \leq Z_\alpha - A) \\
&\leq \alpha + \min \left\{ \delta, 1 - \left(\frac{1 - \delta}{\exp(\varepsilon)} \right)^{\frac{1}{k}} \right\} t \\
&\leq \alpha + \delta t.
\end{aligned}$$

If the null p -values are independent of each other and the non-nulls, and $\{\alpha_t\}$ is a coordinate-wise non-decreasing function of the vector R_1, \dots, R_{t-1} , then we have

$$\begin{aligned}
FDR(t) &= \mathbb{E} \left[\frac{|\mathcal{H}^0 \cap \mathcal{R}(t)|}{|\mathcal{R}(t)|} \right] \\
&= \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E} \left[\frac{I(\log p_j + Z_j \leq \log \alpha_j - A + Z_\alpha)}{|\mathcal{R}(t)|} \right] \\
&\leq \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E} \left[\frac{\min\{\alpha_j \exp(Z_\alpha - Z_j - A), 1\}}{|\mathcal{R}(t)|} \right] \tag{15}
\end{aligned}$$

$$\leq \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E} \left[\frac{\alpha_j}{|\mathcal{R}(t)|} \right] + \Pr(Z_j \leq Z_\alpha - A), \tag{16}$$

where Inequality (15) applies the law of iterated expectations by conditioning on \mathcal{F}^{t-1} and Lemma 1. Inequality (16) follows by a case analysis: if $Z_j > Z_\alpha - A$, then $\exp(Z_\alpha - Z_j - A) < 1$, and thus $\frac{\min\{\alpha_j \exp(Z_\alpha - Z_j - A), 1\}}{|\mathcal{R}(t)|}$ reduces to $\frac{\alpha_j}{|\mathcal{R}(t)|}$. On the other hand, if $Z_j \leq Z_\alpha - A$, then $\frac{\min\{\alpha_j \exp(Z_\alpha - Z_j - A), 1\}}{|\mathcal{R}(t)|} \leq \frac{1}{|\mathcal{R}(t)|} \leq 1$, allowing us to upper bound the expectation by the probability of this event.

We bound the first term in (16) as follows:

$$\sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E} \left[\frac{\alpha_j}{|\mathcal{R}(t)|} \right] \leq \sum_{j \leq t, j \in \mathcal{H}^0} \mathbb{E} \left[\frac{\alpha_j I(p_j > 2\lambda_j)}{(1 - 2\lambda_j)|\mathcal{R}(t)|} \right] \tag{17}$$

$$\begin{aligned}
&\leq \mathbb{E} \left[\frac{\sum_{j \leq t} \alpha_j I(p_j > 2\lambda_j)}{(1 - 2\lambda_j)|\mathcal{R}(t)|} \right] \\
&= \mathbb{E} \left[\widehat{\text{FDP}}_{\text{PAPRIKA}}(t) \right] \\
&\leq \alpha, \tag{18}
\end{aligned}$$

where Inequality (17) applies Lemma 1.

It remains to bound the second term in (16), which we do using Lemma 2 as follows:

$$\begin{aligned}
\sum_{j \leq t, j \in \mathcal{H}^0} \Pr(Z_j \leq Z_\alpha - A) &\leq \sum_{j \leq t} \Pr(Z_j \leq Z_\alpha - A) \\
&= \frac{2t}{3} \exp\left(-\frac{A\varepsilon}{4\eta c}\right) - \frac{t}{6} \exp\left(-\frac{A\varepsilon}{2\eta c}\right) \\
&\leq \min \left\{ \delta, 1 - \left(\frac{1 - \delta}{\exp(\varepsilon)} \right)^{\frac{1}{k}} \right\} t. \tag{19}
\end{aligned}$$

Combining (18) and (19), we reach the conclusion that $FDR(t) \leq \alpha + \min\{\delta, 1 - ((1 - \delta)/\exp(\varepsilon))^{1/k}\}t \leq \alpha + \delta t$. \square

C Proof of Lemma 1

Lemma 1. Assume p_1, p_2, \dots are independent of each other and let $h : \{0, 1\}^k \rightarrow \mathbb{R}$ to 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

$$\mathbb{E} \left[\frac{\alpha_t I(p_t > 2\lambda_t)}{(1 - 2\lambda_t)h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right] \geq \mathbb{E} \left[\frac{\alpha_t}{h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right],$$

and

$$\mathbb{E} \left[\frac{\min\{\alpha_t \exp(Z_\alpha - Z_t - A), 1\}}{h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right] \geq \mathbb{E} \left[\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right].$$

Proof. The proof is similar to the proof of Lemma 2 in [RZWJ18] with the addition of i.i.d. Laplace noise.

In a high level, we hallucinate a vector of p -values that are same as the original vector of p -values, except for the t -th index. This allows us to apply the conditional uniformity property, since now p_t is independent of the hallucinated rejections. We then connect the original rejections and the hallucinated rejections by the monotonicity of the rejections.

We perform our analysis using a hallucinated process: let $\tilde{p}_{1:k}^t$ be a copy of $p_{1:k}$ that is identical everywhere except for the t -th p -value which is set to be 1. That is,

$$\tilde{p}_i = \begin{cases} 1 & \text{if } i = t \\ p_i & \text{otherwise.} \end{cases}$$

Also let the hallucinated Laplace noises $\tilde{Z}_{1:k}^t$ be an identical copy of $Z_{1:k}$, and let \tilde{Z}_α be an identical copy of Z_α . The t -th value of $\tilde{Z}_{1:k}^t$ can be arbitrary since we have ensure the event $\{\tilde{p}_t > 2\lambda_t\}$, so it will fail to become a candidate and the values of \tilde{Z}_t will not be relevant. We denote $\tilde{C}_{1:k}$ and $\tilde{R}_{1:k}$ as the candidates and rejections made using $\tilde{p}_{1:k}^t$, $\tilde{Z}_{1:k}^t$, and \tilde{Z}_α .

By construction, we have $\tilde{R}_{1:t-1} = R_{1:t-1}$. On the event $\{p_t > 2\lambda_t\}$, we have $R_t = \tilde{R}_t = 0$ and $C_t = \tilde{C}_t = 0$ because $\tilde{p}_t = 1$, so both will fail to become candidates, and hence we have $\tilde{R}_{1:k} = R_{1:k}$ and the following equation holds:

$$\frac{\alpha_t I(p_t > 2\lambda_t)}{(1 - 2\lambda_t)h(R_{1:k})} = \frac{\alpha_t I(p_t > 2\lambda_t)}{(1 - 2\lambda_t)h(\tilde{R}_{1:k})}.$$

We note that when $p_t \leq 2\lambda_t$, the above equation still holds since both sides will be zero. Since $\tilde{R}_{1:k}^t$ is independent of p_t , we have

$$\begin{aligned} \mathbb{E} \left[\frac{\alpha_t I(p_t > 2\lambda_t)}{(1 - 2\lambda_t)h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right] &= \mathbb{E} \left[\frac{\alpha_t I(p_t > 2\lambda_t)}{(1 - 2\lambda_t)h(\tilde{R}_{1:k})} \middle| \mathcal{F}^{t-1} \right] \\ &\geq \mathbb{E} \left[\frac{\alpha_t}{h(\tilde{R}_{1:k})} \middle| \mathcal{F}^{t-1} \right] \end{aligned} \tag{20}$$

$$\geq \mathbb{E} \left[\frac{\alpha_t}{h(R_{1:k})} \middle| \mathcal{F}^{t-1} \right] \tag{21}$$

where Inequality (20) is obtained by taking the expectation only with respect to p_t by invoking the conditional super-uniformity property and independence of p_t and $h(\tilde{R}_{1:k})$, and Inequality (21) follows from the facts that $R_i \geq \tilde{R}_i$ for all i and that the function h is non-decreasing.

For the second inequality in the lemma statement, we hallucinate a vector of p -values $\bar{p}_{1:k}^t$ that equals $p_{1:k}$ everywhere except for the t -th p -value which is set to be 0. That is,

$$\bar{p}_i = \begin{cases} 0 & \text{if } i = t \\ p_i & \text{otherwise.} \end{cases}$$

Also let the hallucinated Laplace noises $\bar{Z}_{1:k}^t$ be an identical copy of $Z_{1:k}$, and let \bar{Z}_α be an identical copy of Z_α . We denote $\bar{C}_{1:k}$ and $\bar{R}_{1:k}$ as the candidates and rejections made using $\bar{p}_{1:k}^t$ and $\bar{Z}_{1:k}^t$. By construction, we have $\bar{R}_i = R_i$ and $\bar{C}_i = C_i$ for all $i < t$. On the event that $\{\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A\}$, since $\bar{p}_t = 0$ and we inject the same Laplace noise, we have $\bar{R}_t = R_t = 1$ and $\bar{C}_t = C_t = 1$, and hence also $\bar{R}_{1:k} = R_{1:k}$. Then the following equation holds:

$$\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(R_{1:k})} = \frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(\bar{R}_{1:k})}.$$

We note that when $\log p_t + Z_t > \log \alpha_t + Z_\alpha - A$, the above equation still holds since both sides will be zero.

Since $\bar{R}_{1:k}$ and Z_t, Z_α are independent of p_t , we can take conditional expectations to obtain

$$\begin{aligned} \mathbb{E} \left[\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(R_{1:k})} \middle| \mathcal{F}'^{t-1} \right] &= \mathbb{E} \left[\frac{I(\log p_t + Z_t \leq \log \alpha_t + Z_\alpha - A)}{h(\bar{R}_{1:k})} \middle| \mathcal{F}'^{t-1} \right] \\ &\leq \mathbb{E} \left[\frac{\min\{\alpha_t \exp(Z_\alpha - Z_t - A), 1\}}{h(\bar{R}_{1:k})} \middle| \mathcal{F}'^{t-1} \right] \end{aligned} \quad (22)$$

$$\leq \mathbb{E} \left[\frac{\min\{\alpha_t \exp(Z_\alpha - Z_t - A), 1\}}{h(R_{1:k})} \middle| \mathcal{F}'^{t-1} \right], \quad (23)$$

where Inequality (22) follows by taking expectation only with respect to p_t by invoking the conditional uniformity property and the fact that the support of p-values is $[0, 1]$, and Inequality (23) follows from the facts that $h(R_{1:k}) \leq h(\bar{R}_{1:k})$ since $R_i \leq \bar{R}_i$ for all i and that the function h is non-decreasing. \square