The Power of Batching in Multiple Hypothesis Testing

Tijana Zrnic
UC Berkeley

Daniel L. Jiang
Amazon

Aaditya Ramdas
CMU

Michael I. Jordan
UC Berkeley

Abstract

One important partition of algorithms for controlling the false discovery rate (FDR) in multiple testing is into *offline* and *online* algorithms. The first generally achieve significantly higher power of discovery, while the latter allow making decisions sequentially as well as adaptively formulating hypotheses based on past observations. Using existing methodology, it is unclear how one could trade off the benefits of these two broad families of algorithms, all the while preserving their formal FDR guarantees. To this end, we introduce Batch_{BH} and $\text{Batch}_{\text{St-BH}}$, algorithms for controlling the FDR when a possibly infinite sequence of batches of hypotheses is tested by repeated application of one of the most widely used offline algorithms, the Benjamini-Hochberg (BH) method or Storey's improvement of the BH method. We show that our algorithms interpolate between existing online and offline methodology, thus trading off the best of both worlds.

1 INTRODUCTION

Consider the setting in which a large number of decisions need to be made (e.g., hypotheses to be tested), and one wishes to achieve some form of aggregate control over the quality of these decisions. For binary decisions, a seminal line of research has cast this problem in terms of an error metric known as the *false discovery rate* (FDR) (Benjamini and Hochberg, 1995). The FDR has a Bayesian flavor, conditioning on the decision to reject (i.e., conditioning on a "discovery") and computing the fraction of discoveries that are false. This should be contrasted with traditional metrics—such as sensitivity, specificity, Type I and Type II errors—where one conditions not on the decision but rather on the hypothesis—whether the null or the alternative is true. The scope of research on FDR control has exploded in recent years, with progress on problems such as dependencies, domain-specific constraints, and contextual information.

Classical methods for FDR control are "offline" or "batch" methods, taking in a single batch of data and outputting a set of decisions for all hypotheses at once. This is a serious limitation in the setting of emerging applications at planetary scale, such as A/B testing in the IT industry (Kohavi and Longbotham, 2017), and researchers have responded by developing a range of *online FDR control* methods (Foster and Stine, 2008; Aharoni and Rosset, 2014; Javanmard and Montanari, 2018; Ramdas et al., 2018; Tian and Ramdas, 2019). In the online setting, a decision is made at every time step with no knowledge of future tests, and with possibly infinitely many tests to be conducted overall. By construction, online FDR algorithms guarantee that the FDR is controlled during the whole sequence of tests, and not merely at the end.

Online and offline FDR methods both have their pros and cons. Online methods allow the testing of infinitely many hypotheses, and require less coordination in the setting of multiple decision-makers. Also, perhaps most importantly, they allow the scientist to choose new hypotheses adaptively, depending on the results of previous tests. On the other hand, offline FDR methods tend to make significantly more discoveries due to the fact that they have access to *all* test statistics before making decisions, and not just to the ones from past tests. That is, online methods are myopic, and this can lead to a loss of statistical power. Moreover, the decisions of offline algorithms are *stable*, in the sense that they are invariant to any implicit ordering of hypotheses; this is not true of online algorithms, whose discovery set can vary drastically depending on the ordering of hypotheses (Foster and Stine, 2008).

By analogy with batch and online methods in gradient-based optimization, these considerations suggest investigating an intermediate notion of "mini-batch," hop-

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

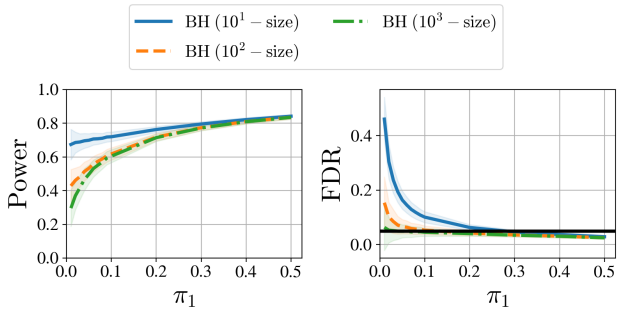

Figure 1: Statistical power and FDR versus probability of non-null hypotheses, π_1, for naively composed BH, at batch sizes 10, 100, and 1000. The total number of hypotheses is 3000, and the target FDR is 0.05.

ing to exploit and manage some form of tradeoff between methods that are purely batch or purely online.

Managing such a tradeoff is, however, more challenging in the setting of false-discovery-rate control than in the optimization setting. Indeed, consider a naive approach that would run offline algorithms on different batches of hypotheses in an online fashion. Unfortunately, such a method violates the assumptions behind FDR control, yielding uncontrolled, possibly meaningless FDR guarantees. To illustrate this point, Figure 1 plots the performance of the Benjamini-Hochberg (BH) algorithm (Benjamini and Hochberg, 1995), run repeatedly under the same FDR level 0.05 on different batches of hypotheses. We observe that the FDR can be much higher than the nominal value.

In this paper, we develop FDR procedures which are appropriate for multiple batches of tests. We allow testing of possibly infinitely many batches in an online fashion. We refer to this setting as *online batch testing*. More precisely, we improve the widely-used BH algorithm (Benjamini and Hochberg, 1995) and a variant that we refer to Storey-BH (Storey, 2002; Storey et al., 2004), such that their repeated composition does not violate the desired FDR guarantees. We refer to these sequential, FDR-preserving versions of BH and Storey-BH as Batch_{BH} and $\text{Batch}_{\text{St-BH}}$, respectively. As is the case for state-of-the-art online algorithms, our procedures allow testing an infinite sequence of batches of adaptively chosen hypotheses, but they also enjoy a higher power of discovery than those algorithms. Finally, since they consist of compositions of offline FDR algorithms with provable guarantees, they immediately imply FDR control over each constituent batch, and not just over the whole sequence of tests. This property has value in settings with natural groupings of hypotheses, where the scientist might be interested in the overall FDR, but also the FDR over certain subgroups of hypotheses.

1.1 Outline

In Section 2, we present preliminaries and sketch the main ideas behind our proofs. In Section 3, we define the Batch_{BH} family of algorithms and state its FDR guarantees. In Section 4, we do the same for $\text{Batch}_{\text{St-BH}}$ algorithms. In Section 5, we demonstrate the performance of our methods on synthetic data. In the Appendix, we consider online batch FDR control under positive dependence. The Appendix also contains a short overview of some related work, all proofs, as well as additional experimental results.

2 PRELIMINARIES

We introduce a formal description of the testing process, together with some preliminaries.

At every time $t \in \mathbb{N}$, a batch of n_t hypotheses is tested using a pre-specified offline FDR procedure. We consider two such procedures, the BH and Storey-BH procedures, which we review in the Appendix for the reader's convenience. The batches arrive sequentially, in a stream; at the time of testing the t-th batch, no information about future batches needs to be available, such as their size or their number. For each hypothesis, there is unknown ground truth that says whether the hypothesis is null or non-null. Denote the set of hypotheses in the t-th batch by $\mathbf{H}_t := \{H_{t,1}, \dots, H_{t,n_t}\}$. Each hypothesis has a p-value associated with it. Let $\mathbf{P}_t$ denote the p-values corresponding to the t-th batch of hypotheses, given by $\mathbf{P}_t := \{P_{t,1}, \dots, P_{t,n_t}\}$, where $P_{t,j}$ is the j-th p-value in batch t. Denote by $\mathcal{H}_t^0$ the indices corresponding to null hypotheses in batch t, and let $\mathcal{R}_t$ denote the indices of rejections, or *discoveries*, in batch t:

$$\mathcal{H}_t^0 := \{i : H_{t,i} \text{ is null}\}, \ \mathcal{R}_t := \{i : H_{t,i} \text{ is rejected}\}.$$

We will also informally say that a p-value is rejected, if its corresponding hypothesis is rejected.

We now define the *false discovery rate (FDR) up to time t*:

$$\text{FDR}(t) := \mathbb{E}\left[\text{FDP}(t)\right] := \mathbb{E}\left[\frac{\sum_{s=1}^{t} |\mathcal{H}_s^0 \cap \mathcal{R}_s|}{(\sum_{s=1}^{t} |\mathcal{R}_s|) \vee 1}\right],$$

where $\text{FDP}(t)$ denotes a random quantity called the *false discovery proportion* up to time t. To simplify notation, we also define $R_t := |\mathcal{R}_t|$. In real applications, it does not suffice to merely control the FDR (which we can do by making no discoveries, which results in $\text{FDR} = 0$); rather, we also need to achieve high

statistical *power*:

$$\text{Power}(t) := \mathbb{E}\left[\frac{\sum_{s=1}^{t}|([n_s]\setminus\mathcal{H}_s^0)\cap\mathcal{R}_s|}{\sum_{s=1}^{t}|([n_s]\setminus\mathcal{H}_s^0)|}\right],$$

where $[n_s]\setminus\mathcal{H}_s^0$ are the non-null hypotheses in batch s.

The goal of the Batch_{BH} procedure is to achieve high power, while guaranteeing $\text{FDR}(t) \le \alpha$ for a pre-specified level $\alpha \in (0,1)$ and for all $t \in \mathbb{N}$. To do so, the algorithm adaptively determines a *test level* α_t based on information about past batches of tests, and tests $\mathbf{P}_t$ under FDR level α_t using the standard BH method. The $\text{Batch}_{\text{St-BH}}$ method operates in a similar way, the difference being that it uses the Storey-BH method for every batch, as opposed to BH.

Define R_t^+ to be the maximum "augmented" number of rejections in batch t, if one p-value in $\mathbf{P}_t$ is "hallucinated" to be equal to zero, and all other p-values and level α_t are held fixed; the maximum is taken over the choice of the p-value which is set to zero. More formally, let $\mathcal{A}_t$ denote a map from a set of p-values $\mathbf{P}_t$ (and implicitly, a level α_t) to a set of rejections $\mathcal{R}_t$. Hence, $R_t = |\mathcal{A}_t(\mathbf{P}_t)|$. In our setting, $\mathcal{A}_t$ will be the BH algorithm in the case of Batch_{BH} and Storey-BH algorithm in the case of $\text{Batch}_{\text{St-BH}}$. Then, R_t^+ is defined as

$$R_t^+ := \max_{i\in[n_t]} |\mathcal{A}_t(\mathbf{P}_t \setminus P_{t,i}\cup 0)|. \tag{1}$$

Note that R_t^+ could be as large as n_t in general. For an extreme example, let $n_t = 3$, $\mathbf{P}_t := \{2\alpha/3, \alpha, 4\alpha/3\}$, and consider $\mathcal{A}_t$ being the BH procedure. Then $R_t = 0$, while $R_t^+ = 3$. However, such "adversarial" p-values are unlikely to be encountered in practice and we typically expect R_t^+ to be roughly equal to R_t+1. In other words, we expect that when an unrejected p-value is set to 0, it will be a new rejection, but typically will not result in other rejections as well. This intuition is confirmed by our experiments, where we plot $R_t^+ - R_t$ for Batch_{BH} with different batch sizes and observe that this quantity concentrates around 1. These plots are available in Figure 14 in the Appendix.

Let the natural filtration induced by the testing process be denoted

$$\mathcal{F}^t := \sigma(\mathbf{P}_1, \dots, \mathbf{P}_t),$$

which is the σ-field of all previously observed p-values. Naturally, we require α_t to be $\mathcal{F}^{t-1}$-measurable; the test level at time t is only allowed to depend on information seen before t. It is worth pointing out that this filtration is different from the corresponding filtration in prior online FDR work, which was typically of the form $\sigma(R_1, \dots, R_t)$. The benefits of this latter, smaller filtration arise when proving *modified* FDR (mFDR) guarantees, which we do not consider in this paper. Moreover, a richer filtration allows more freedom in choosing α_t, making our choice of $\mathcal{F}^t$ a natural one.

For the formal guarantees of Batch_{BH} and $\text{Batch}_{\text{St-BH}}$, we will require the procedures to be *monotone*. Let $(\{P_{1,1},\dots,P_{1,n_1}\},\dots,\{P_{t,1},\dots,P_{t,n_t}\})$ and $(\{\tilde{P}_{1,1},\dots,\tilde{P}_{1,n_1}\},\dots,\{\tilde{P}_{t,1},\dots,\tilde{P}_{t,n_t}\})$ be two sequences of p-value batches, which are identical in all entries but (s,i), for some $s \le t$: $\tilde{P}_{s,i} < P_{s,i}$. Then,

$$\text{a procedure is monotone if } \sum_{r=s+1}^{t} R_r \le \sum_{r=s+1}^{t} \tilde{R}_r.$$

Intuitively, this condition says that making any of the tested p-values smaller can only make the overall number of rejections larger. A similar assumption appears in online FDR literature (Javanmard and Montanari, 2018; Ramdas et al., 2018; Zrnic et al., 2018; Tian and Ramdas, 2019). In general, whether or not a procedure is monotone is a property of the p-value distribution; notice, however, that monotonicity can be assessed empirically (it does not depend on the unknown ground truth). One way to ensure monotonicity is to make α_t a coordinate-wise non-increasing function of $(P_{1,1},\dots,P_{1,n_1},P_{2,1},\dots,P_{t-1,n_{t-1}})$. In the Appendix, we give examples of monotone strategies.

Finally, we review a basic property of null p-values. If a hypothesis $H_{t,i}$ is truly null, then the corresponding p-value $P_{t,i}$ stochastically dominates the uniform distribution, or is *super-uniformly distributed*, meaning:

$$\text{If } H_{t,i} \text{ is null, then } \mathbb{P}\{P_{t,i} \le u\} \le u \text{ for all } u \in [0,1].$$

2.1 Algorithms via Empirical FDP Estimates

We build on Storey's interpretation of the BH procedure (Storey, 2002) as an empirical Bayesian procedure, based on empirical estimates of the false discovery proportion. In this section, we give a sketch of this idea, as it is at the core of our algorithmic constructions. The steps presented below are not fully rigorous, but are simply meant to develop intuition.

When an algorithm decides to reject a hypothesis, there is generally no way of knowing if the rejected hypothesis is null or non-null. Consequently, it is impossible for the scientist to know the achieved FDP. However, by exploiting the super-uniformity of null p-values, it is possible to estimate the behavior of the FDP *on average*. More explicitly, there are tools that utilize only the information available to the scientist to upper bound the average FDP, that is the FDR.

We sketch this argument for the Batch_{BH} procedure

here, formalizing the argument in Theorem 1. Theorem 2 gives an analogous proof for the $\text{Batch}_{\text{St-BH}}$ procedure.

By definition, the FDR is equal to

$$\mathbb{E}\left[\frac{\sum_{s=1}^{t}|\mathcal{H}_s^0\cap\mathcal{R}_s|}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}\right]=\sum_{s=1}^{t}\mathbb{E}\left[\frac{\sum_{i\in\mathcal{H}_s^0}\mathbf{1}\left\{P_{s,i}\le\frac{\alpha_s}{n_s}R_s\right\}}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}\right],$$

where we use the definition of the BH procedure. If the p-values are independent, we will show that it is valid to upper bound this expression by inserting an expectation in the numerator, approximately as

$$\sum_{s=1}^{t}\mathbb{E}\left[\frac{\sum_{i\in\mathcal{H}_s^0}\mathbb{P}\left\{P_{s,i}\le\frac{\alpha_s}{n_s}R_s\;\middle|\;\alpha_s,R_s\right\}}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}\right].$$

Invoking the super-uniformity of null p-values (and temporarily ignoring dependence between $P_{s,i}$ and R_s), we get

$$\sum_{s=1}^{t}\mathbb{E}\left[\frac{|\mathcal{H}_s^0|\frac{\alpha_s}{n_s}R_s}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}\right]\le\mathbb{E}\left[\frac{\sum_{s=1}^{t}\alpha_s R_s}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}\right].$$

Suppose we define $\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)\approx\frac{\sum_{s=1}^{t}\alpha_s R_s}{(\sum_{r=1}^{t}|\mathcal{R}_r|)\vee 1}$. This quantity is purely *empirical*; each term is known to the scientist. Hence, by an appropriate choice of α_s at each step, one can ensure that $\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)\le\alpha$ for all t. But by the sketch given above, this would immediately imply $\text{FDR}\le\alpha$, as desired. This proof sketch is the core idea behind our algorithms.

It is important to point out that there is not a single way of ensuring $\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)\le\alpha$; this approach gives rise to a whole family of algorithms. Naturally, the choice of α_s can be guided by prior knowledge or importance of a given batch, as long as the empirical estimate is controlled under α.

3 ONLINE BATCH FDR CONTROL VIA Batch_{BH}

In this section, we define the Batch_{BH} class of algorithms and state our main technical result regarding its FDR guarantees.

Definition 1 (Batch_{BH})**.** The Batch_{BH} procedure is any rule for assigning test levels α_s such that

$$\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t):=\sum_{s\le t}\alpha_s\frac{R_s^+}{R_s^+ + \sum_{r\le t,r\neq s}R_r}$$

is always controlled under a pre-determined level α.

Note that if we were to approximate R_s^+ by R_s, we would arrive exactly at the estimate derived in the proof sketch of the previous section.

This way of controlling $\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)$ interpolates between prior offline and online FDR approaches. First, suppose that there is only one batch. Then, the user is free to pick α_1 to be any level less than or equal to α, in which case it makes sense to simply pick α. On the other hand, if every batch is of size one we have $R_s^+=1$, hence the FDP estimate reduces to

$$\begin{aligned}\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)&=\sum_{s\le t}\frac{\alpha_s}{1+\sum_{r\le t,r\neq s}R_r}\\&\le\frac{\sum_{s\le t}\alpha_s}{\sum_{r\le t}R_r}\\&:=\widehat{\text{FDP}}_{\text{LORD}}(t),\end{aligned}$$

where the intermediate inequality is almost an equality whenever the total number of rejections is non-negligible. The quantity $\widehat{\text{FDP}}_{\text{LORD}}(t)$ is an estimate of FDP that is implicitly used in an existing online algorithm known as LORD (Javanmard and Montanari, 2018), as detailed by Ramdas et al. (2017). Thus, Batch_{BH} can be seen as a generalization of both BH and LORD, simultaneously allowing arbitrary batch sizes (like BH) and an arbitrary number of batches (like LORD).

We now state our main formal result regarding FDR control of Batch_{BH}. As suggested in Section 2, together with the requirement that $\widehat{\text{FDP}}_{\text{Batch}_{\text{BH}}}(t)\le\alpha$ for all $t\in\mathbb{N}$ we also need to guarantee that the procedure is monotone. Recall that monotonicity roughly means that making any of the tested p-values smaller can only result in more rejections. In general, any reasonable update for α_t satisfying Definition 1 is expected to be monotone for non-adversarially chosen p-values. We analyze one such natural update in Section 5. However, one can also construct more conservative algorithms which are guaranteed to be monotone uniformly across *all* p-value sequences. We present multiple such procedures in the Appendix.

Theorem 1. *If all null p-values in the sequence are independent of each other and the non-nulls, and the Batch_{BH} procedure is monotone, then it provides anytime* FDR *control: for every $t\in\mathbb{N}$, $\text{FDR}(t)\le\alpha$.*

We defer the proof of Theorem 1 to the Appendix.

4 ONLINE BATCH FDR CONTROL VIA $\text{Batch}_{\text{St-BH}}$

In addition to the FDR level α, the Storey-BH algorithm also requires a user-chosen constant $\lambda\in(0,1)$ as a parameter. This extra parameter allows the algorithm to be more adaptive to the data at hand, constructing a better FDP estimate (Storey, 2002). We revisit this estimate in the Appendix.

Thus, our extension of Storey-BH, $\text{Batch}_{\text{St-BH}}$, requires a user-chosen constant $\lambda_t \in (0,1)$ as an input to the algorithm at time $t \in \mathbb{N}$. Unless there is prior knowledge of the p-value distribution, it is a reasonable heuristic to simply set $\lambda_t = 0.5$ for all t (Storey, 2002; Storey et al., 2004).

Denote by $\max_t := \arg\max_i \{P_{t,i} : i \in [n_t]\}$ the index corresponding to the maximum p-value in batch t. With this, define the *null proportion sensitivity* for batch t as:

$$k_t := \frac{\sum_{i \le n_t} \mathbf{1}\{P_{t,i} > \lambda_t\}}{1 + \sum_{j \le n_t, j \neq \max_t} \mathbf{1}\{P_{t,j} > \lambda_t\}}.$$

Now we can define the $\text{Batch}_{\text{St-BH}}$ family of methods.

Definition 2. The $\text{Batch}_{\text{St-BH}}$ procedure is any rule for assigning test levels α_s, such that

$$\widehat{\text{FDP}}_{\text{Batch}_{\text{St-BH}}}(t) := \sum_{s \le t} \frac{\alpha_s k_s R_s^+}{R_s^+ + \sum_{r \le t, r \neq s} R_r}$$

is controlled under a pre-determined level α.

Just like Batch_{BH}, $\text{Batch}_{\text{St-BH}}$ likewise interpolates between existing offline and online FDR procedures. If there is a single batch of tests, the user can pick the test level α_1 to be at most α, in which case it makes sense to simply pick α. This follows due to $k_i \le 1$ by definition. On the other end of the spectrum, in the fully online setting, $\text{Batch}_{\text{St-BH}}$ reduces to the SAFFRON procedure (Ramdas et al., 2018). Indeed, since $k_t = \mathbf{1}\{P_{t,1} > \lambda_t\}$, the FDP estimate reduces to:

$$\begin{aligned}\widehat{\text{FDP}}_{\text{Batch}_{\text{St-BH}}}(t) &= \sum_{s \le t} \frac{\alpha_s \mathbf{1}\{P_{s,1} > \lambda_s\}}{1 + \sum_{r \le t, r \neq s} R_r} \\ &\le \frac{\sum_{s \le t} \alpha_s \mathbf{1}\{P_{s,1} > \lambda_s\}}{\sum_{r \le t} R_r} \\ &:= \widehat{\text{FDP}}_{\text{SAFFRON}}(t),\end{aligned}$$

which is equivalent to the FDP estimate defined by Ramdas et al. (2018). We discuss the connections between the two FDP estimates in more detail in the Appendix.

We are now ready to state our main result for $\text{Batch}_{\text{St-BH}}$. Just like Batch_{BH}, the $\text{Batch}_{\text{St-BH}}$ procedure requires monotonicity to control the FDR (as per the argument outlined in Section 2). We describe multiple monotone versions of $\text{Batch}_{\text{St-BH}}$ in the Appendix, and discuss some useful heuristics in Section 5.

Theorem 2. *If the null p-values in the sequence are independent of each other and the non-nulls, and the $\text{Batch}_{\text{St-BH}}$ procedure is monotone, then it provides anytime* FDR *control: for every $t \in \mathbb{N}$, $\text{FDR}(t) \le \alpha$.*

The proof of Theorem 2 is presented in the Appendix.

5 NUMERICAL EXPERIMENTS

We compare the performance of Batch_{BH} and $\text{Batch}_{\text{St-BH}}$ with two state-of-the-art online FDR algorithms: LORD (Javanmard and Montanari, 2018; Ramdas et al., 2017) and SAFFRON (Ramdas et al., 2018). Specifically, we compare the achieved power and FDR of these methods on synthetic data, while in the Appendix we study a real fraud detection data set.

As explained in prior literature (Ramdas et al., 2018), LORD and BH are non-adaptive methods, while SAFFRON and Storey-BH adapt to the tested p-values through the parameter λ_t. We keep comparisons fair by comparing Batch_{BH} with LORD, and $\text{Batch}_{\text{St-BH}}$ with SAFFRON.

As discussed in Section 2, there are various ways to assign α_i such that the appropriate FDP estimate is controlled under α. Moreover, as we argued in Section 3 and Section 4, this needs to be done in a monotone way to guarantee FDR control for an arbitrary p-value distribution. In the experimental sections of this paper, however, we resort to a heuristic. Enforcing monotonicity uniformly across all distributions diminishes the power of FDR methods. Hence, we apply algorithms which control the corresponding FDP estimates and are expected to be monotone under natural p-value distributions, but possibly not for adversarially chosen ones. In the Appendix we test the monotonicity of these procedures empirically, and demonstrate that it is satisfied with overwhelming probability. We now present the specific algorithms that we studied.

Algorithm 1 The Batch_{BH} algorithm

Input: FDR level α, non-negative sequence $\{\gamma_s\}_{s=1}^{\infty}$ such that $\sum_{s=1}^{\infty} \gamma_s = 1$.

Set $\alpha_1 = \gamma_1 \alpha$;

for $t = 1, 2, \ldots$ **do**

  Run the BH method at level α_t on batch $\mathbf{P}_t$;

  Set $\beta_{t+1} = \sum_{s \le t} \alpha_s \frac{R_s^+}{R_s^+ + \sum_{r \neq s, r \le t} R_r}$;

  Set $\alpha_{t+1} = \left(\sum_{s \le t+1} \gamma_s \alpha - \beta_{t+1}\right) \frac{n_{t+1} + \sum_{s \le t} R_s}{n_{t+1}}$;

end

Algorithm 2 The $\text{Batch}_{\text{St-BH}}$ algorithm

Input: FDR level α, non-negative sequence $\{\gamma_s\}_{s=1}^{\infty}$ such that $\sum_{s=1}^{\infty} \gamma_s = 1$

Set $\alpha_1 = \gamma_1 \alpha$;

for $t = 1, 2, \ldots$ **do**

  Run the Storey-BH method at level α_t with parameter λ_t on batch $\mathbf{P}_t$;

  Set $\beta_{t+1} = \sum_{s \le t} k_s \alpha_s \frac{R_s^+}{R_s^+ + \sum_{r \neq s, r \le t} R_r}$;

  Set $\alpha_{t+1} = \left(\sum_{s \le t+1} \gamma_s \alpha - \beta_{t+1}\right) \frac{n_{t+1} + \sum_{s \le t} R_s}{n_{t+1}}$;

end

The choice of λ_t should generally depend on the number and strength of non-null p-values the analyst expects to see in the sequence. As suggested in previous works on similar adaptive methods (Storey, 2002; Storey et al., 2004; Ramdas et al., 2018), it is reasonable to set $\lambda_t \equiv 0.5$ if no prior knowledge is assumed.

The reason why we add a sequence $\{\gamma_s\}_{s=1}^{\infty}$ as a hyperparameter is to prevent α_t from vanishing. If we immediately invest the whole error budget α, i.e. we set $\gamma_1 = 1$ and $\gamma_s = 0, s \neq 1$, then α_t might be close to 0 for small batches, given that R_t^+ could be close to n_t. For this reason, for the smallest batch size we consider (which is 10), we pick $\gamma_s \propto s^{-2}$. Similar error budget investment strategies have been considered in prior work (Ramdas et al., 2018; Tian and Ramdas, 2019). For larger batch sizes, R_t^+ is generally much smaller than n_t, so for all other batch sizes we invest more aggressively by picking $\gamma_1 = \gamma_2 = \frac{1}{2}$, $\gamma_s = 0$, $s \notin \{1, 2\}$. This is analogous to the default choice of "initial wealth" for LORD and SAFFRON of $\frac{\alpha}{2}$, which we also use in our experiments. We only adapt our choice of $\{\gamma_s\}_{s=1}^{\infty}$ to the batch size, as that is information available to the scientist. In general, one can achieve better power if $\{\gamma_s\}_{s=1}^{\infty}$ is tailored to parameters such as the number of non-nulls and their strength, but given that such information is typically unknown, we keep our hyperparameters agnostic to such specifics.

In the Appendix we prove Fact 1, which states the Algorithm 1 controls the appropriate FDP estimate. We omit the analogous proof for Algorithm 2 due to the similarity of the two proofs.

Fact 1. Algorithm 1 maintains $\widehat{\mathrm{FDP}}_{\mathrm{Batch_{BH}}}(t) \leq \alpha$.

We test for the means of a sequence of $T = 3000$ independent Gaussian observations. Under the null, the mean is $\mu_0 = 0$. Under the alternative, the mean is μ_1, whose distribution differs in two settings that we studied. For each index $i \in \{1, \ldots, T\}$, the observation Z_i is distributed according to

$$Z_i \sim \begin{cases} N(\mu_0, 1), \text{with probability } 1 - \pi_1, \\ N(\mu_1, 1), \text{with probability } \pi_1. \end{cases}$$

In all experiments we set $\alpha = 0.05$. All plots display the average and one standard deviation around the average of power or FDR, against $\pi_1 \in \{0.01, 0.02, \ldots, 0.09\} \cup \{0.1, 0.2, 0.3, 0.4, 0.5\}$ (interpolated for in-between values). All quantities are averaged over 500 independent trials.

5.1 Constant Gaussian Means

In this setting, we choose the mean under the alternative to be constant, $\mu_1 = 3$. Each observation is converted to a one-sided p-value as $P_i = \Phi(-Z_i)$, where Φ is the standard Gaussian CDF.

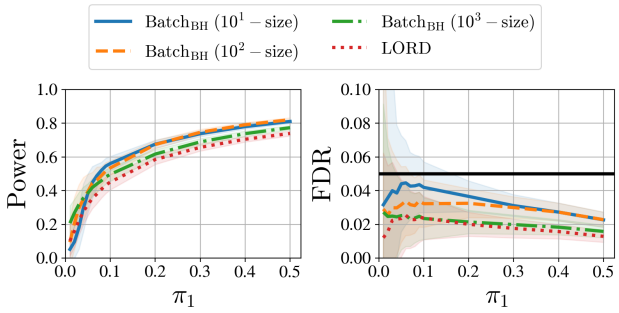

Figure 2: Statistical power and FDR versus probability of non-null hypotheses π_1 for Batch$_{\mathrm{BH}}$ (at batch sizes 10, 100, and 1000) and LORD. The observations under the null are $N(0, 1)$, and the observations under the alternative are $N(3, 1)$.

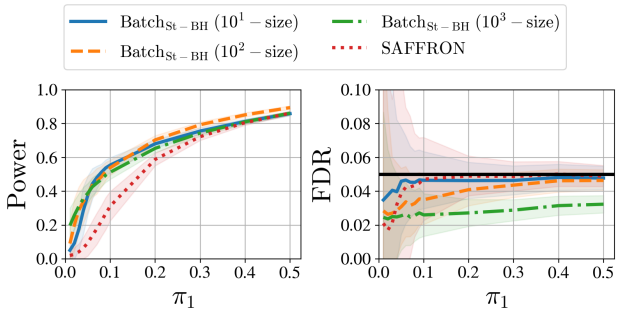

Figure 3: Statistical power and FDR versus probability of non-null hypotheses π_1 for Batch$_{\mathrm{St\text{-}BH}}$ (at batch sizes 10, 100, and 1000) and SAFFRON. The observations under the null are $N(0, 1)$, and the observations under the alternative are $N(3, 1)$.

Non-adaptive procedures. Figure 2 compares the statistical power and FDR of Batch$_{\mathrm{BH}}$ and LORD as functions of π_1. Across almost all values of π_1, the online batch procedures outperform LORD, with the exception of Batch$_{\mathrm{BH}}$ with the smallest considered batch size, for small values of π_1.

Adaptive procedures. Figure 3 compares the statistical power and FDR of Batch$_{\mathrm{St\text{-}BH}}$ and SAFFRON as functions of π_1. The online batch procedures dominate SAFFRON for all values of π_1. The difference in power is especially significant for $\pi_1 \leq 0.1$, which is a reasonable range for the non-null proportion in most real-world applications.

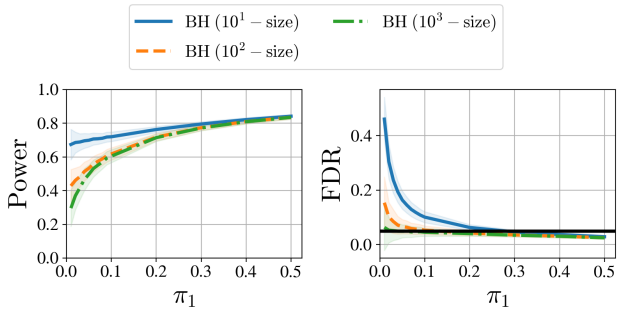

Figure 4: Statistical power and FDR versus probability of non-null hypotheses π_1 for naively composed BH (at batch sizes 10, 100, and 1000). The observations under the null are $N(0,1)$, and the observations under the alternative are $N(3,1)$.

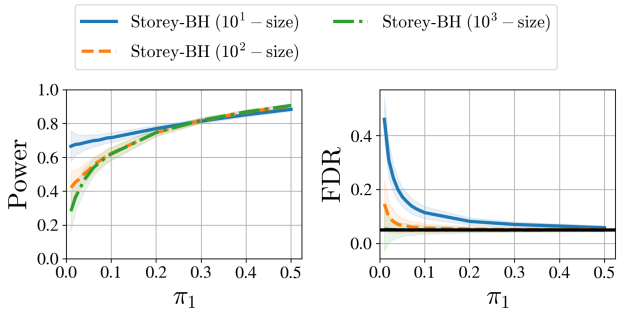

Figure 5: Statistical power and FDR versus probability of non-null hypotheses π_1 for naively composed Storey-BH (at batch sizes 10, 100, and 1000). The observations under the null are $N(0,1)$, and the observations under the alternative are $N(3,1)$.

Naively composed procedures. Figure 4 and Figure 5 show the statistical power and FDR versus π_1 for BH and Storey-BH naively run in a batch setting where each individual batch is run using test level $\alpha = 0.05$. Although there is a significant boost in power, the FDR is generally much higher than the desired value for reasonably small π_1; this is not true of batch size 1000 because only 3 batches are composed, where we know that in the worst case FDR $\leq 3\alpha$.

5.2 Random Gaussian Alternative Means

Now we consider random alternative means; we let $\mu_1 \sim N(0, 2\log T)$. Unlike the previous setting, this is a hard testing problem in which non-nulls are barely detectable (Javanmard and Montanari, 2018). Each observation is converted to a two-sided p-value as $P_i = 2\Phi(-|Z_i|)$, where Φ is the standard Gaussian CDF.

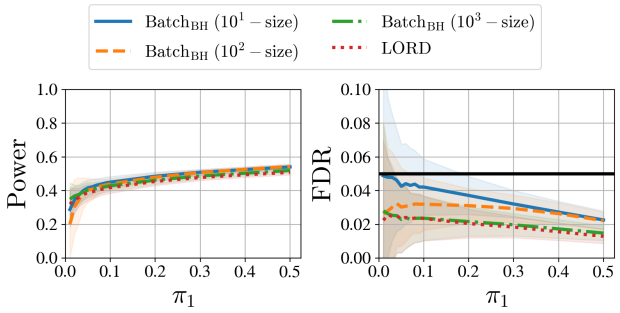

Figure 6: Statistical power and FDR versus probability of non-null hypotheses π_1 for Batch_{BH} (at batch sizes 10, 100, and 1000) and LORD. The observations under the null are $N(0,1)$, and the observations under the alternative are $N(\mu_1, 1)$ where $\mu_1 \sim N(0, 2\log T)$.

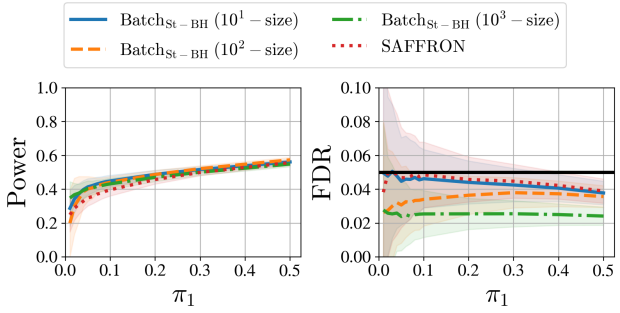

Figure 7: Statistical power and FDR versus probability of non-null hypotheses π_1 for $\text{Batch}_{\text{St-BH}}$ (at batch sizes 10, 100, and 1000) and SAFFRON. The observations under the null are $N(0,1)$, and the observations under the alternative are $N(\mu_1, 1)$ where $\mu_1 \sim N(0, 2\log T)$.

Non-adaptive procedures. Figure 6 compares the statistical power and FDR of Batch_{BH} and LORD as functions of π_1. Again, for most values of π_1 all batch procedures outperform LORD.

Adaptive procedures. Figure 7 compares the statistical power and FDR of $\text{Batch}_{\text{St-BH}}$ and SAFFRON as functions of π_1. For high values of π_1, all procedures behave similarly, while for small values of π_1 the batch procedures dominate.

Naively composed procedures. Figure 8 and Figure 9 show the statistical power and FDR versus π_1 for BH and Storey-BH naively run in a batch setting where each individual batch is run using test level $\alpha = 0.05$. In this hard testing problem, there is not as much gain in power, and the FDR is extremely high, as expected.

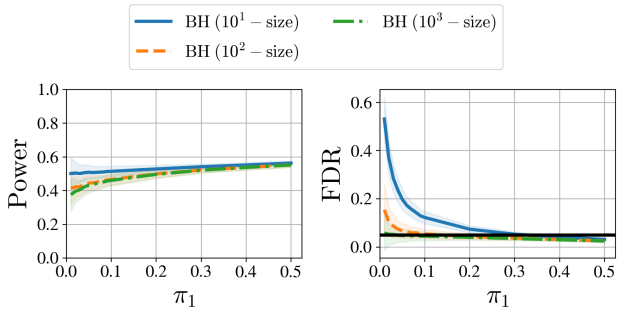

Figure 8: Statistical power and FDR versus probability of non-null hypotheses π_1 for naively composed BH (at batch sizes 10, 100, and 1000). The observations under the null are $N(0,1)$, and the observations under the alternative are $N(\mu_1, 1)$ where $\mu_1 \sim N(0, 2\log T)$.

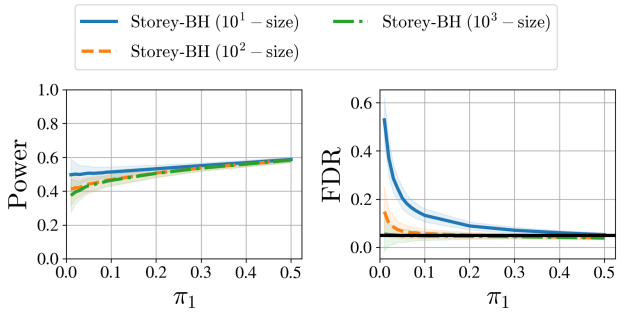

Figure 9: Statistical power and FDR versus probability of non-null hypotheses π_1 for naively composed Storey-BH (at batch sizes 10, 100, and 1000). The observations under the null are $N(0,1)$, and the observations under the alternative are $N(\mu_1, 1)$ where $\mu_1 \sim N(0, 2\log T)$.

6 DISCUSSION

In this paper, we have presented algorithms for FDR control in online batch settings; at every time step, a batch of decisions is made via the BH or Storey-BH algorithm, and batches arrive sequentially, in a stream. We discuss several possible extensions of this framework.

Alpha-investing version of Batch$_{\text{St-BH}}$. In the definition of Batch$_{\text{St-BH}}$, we considered deterministic values of λ_t for simplicity. By imposing a monotonicity constraint on λ_t (Ramdas et al., 2018), one could generalize Batch$_{\text{St-BH}}$ to handle random λ_t as well. In particular, this would lead to a batch generalization of alpha-investing (Foster and Stine, 2008), in which $\lambda_t = \alpha_t$.

Asynchronous online batch testing. Zrnic et al. (2018) consider the setting of asynchronous online testing, in which one conducts a possibly infinite number of sequential experiments which could, importantly, be running in parallel. They generalize multiple online FDR algorithms to handle this so-called *asynchronous testing* problem. Using their technical tools, namely the idea of conflict sets, one can adjust Batch$_{\text{BH}}$ and Batch$_{\text{St-BH}}$ to operate in an asynchronous manner.

ADDIS algorithm. Tian and Ramdas (2019) have presented an adaptive online FDR algorithm called ADDIS that was designed with the goal of improving the power of online FDR methods when the null p-values are conservative. The same paper also gives the offline analog of ADDIS. Using our proof technique, one can design online batch corrections for the offline counterpart of ADDIS, thus interpolating between the two algorithms of Tian and Ramdas.

Batch size versus power. As our experiments indicate, it is not clear that bigger batch sizes give better power. Intuitively, if a batch is very large, say of size n, the slope α/n of the BH procedure is very conservative, and it might be better to split up the tests into multiple batches. It would be of great importance for the practitioner to conduct a rigorous analysis of the relationship between batch size and power.

mFDR control. Many treatments of online FDR have focused on mFDR guarantees (together with FDR guarantees), mostly due to simplicity of the proofs, but also because mFDR can be a reasonable error metric in some settings. Indeed, in the online batch setting, mFDR is potentially a reasonable target measure, because mFDR, unlike FDR, is preserved under composition; if two disjoint batches of tests are guaranteed to achieve mFDR $\leq \alpha$, pooling their results also ensures mFDR $\leq \alpha$. This favorable property has been recognized in prior work (van den Oord, 2008). Unfortunately, the BH algorithm controls mFDR only asymptotically (Genovese and Wasserman, 2002; Sun and Cai, 2007). Moreover, how closely it controls mFDR depends on its "stability," as we show in the Appendix. In fact it has been noted that BH is not stable (Gordon et al., 2007), making FDR our preferred choice of metric.

Acknowledgments

AR thanks Adel Javanmard for a discussion during the early phases of this work.

References

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.

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.

Dean Foster and Robert 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.

Christopher Genovese and Larry Wasserman. Operating characteristics and extensions of the false discovery rate procedure. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3): 499–517, 2002.

Alexander Gordon, Galina Glazko, Xing Qiu, Andrei Yakovlev, et al. Control of the mean number of false discoveries, bonferroni and stability of multiple testing. *The Annals of Applied Statistics*, 1(1):179–190, 2007.

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.

Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. *Encyclopedia of machine learning and data mining*, pages 922–929, 2017.

Aaditya Ramdas, Fanny Yang, Martin Wainwright, and Michael Jordan. Online control of the false discovery rate with decaying memory. In *Advances In Neural Information Processing Systems*, pages 5655–5664, 2017.

Aaditya Ramdas, Tijana Zrnic, Martin Wainwright, and Michael Jordan. SAFFRON: an adaptive algorithm for online control of the false discovery rate. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4286–4294, 2018.

John Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 64(3):479–498, 2002.

John Storey, Jonathan Taylor, and David Siegmund. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 66(1):187–205, 2004.

Wenguang Sun and T Tony Cai. Oracle and adaptive compound decision rules for false discovery rate control. *Journal of the American Statistical Association*, 102(479):901–912, 2007.

Jinjin Tian and Aaditya Ramdas. ADDIS: adaptive algorithms for online FDR control with conservative nulls. *Advances in Neural Information Processing Systems*, 2019.

Edwin JCG van den Oord. Controlling false discoveries in genetic studies. *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics*, 147(5): 637–644, 2008.

Tijana Zrnic, Aaditya Ramdas, and Michael I Jordan. Asynchronous online testing of multiple hypotheses. *arXiv preprint arXiv:1812.05068*, 2018.