# APPENDIX TO
# FASM AND FAST-YB: SIGNIFICANT PATTERN MINING WITH FALSE DISCOVERY RATE CONTROL

## A. ADDITIONAL PRELIMINARIES

Two random variables $X_1$ and $X_2$ are *statistically independent* if their joint probability distribution factorises as $\mathbb{P}[X_1 = x_1, X_2 = x_2] = \mathbb{P}[X_1 = x_1]\,\mathbb{P}[X_2 = x_2]$. Note that the challenge in determining whether $g_P(X)$ and $Y$ are independent or not arises from the fact that we don't have access to the joint distribution of the random variables, but only to their realizations $((g_P(x_i), y_i))_{i=1,\dots,n}$, which can be extracted from the dataset $\mathcal{D}$.

Since both occurrence indicators and labels are binary, the contingency table takes the simple form reported in Table I.

TABLE I
CONTINGENCY TABLE FOR BINARY LABELS.

| Variables | $g_P(x) = 1$ | $g_P(x) = 0$ | Row total |
|---|---|---|---|
| $y = 1$ | $a_P$ | $n_1 - a_P$ | $n_1$ |
| $y = 0$ | $r_P - a_P$ | $n - n_1 - r_P$ | $n - n_1$ |
| Col. total | $r_P$ | $n - r_P$ | $n$ |

Then, we have that the joint distribution can be estimated using these counts (e.g. the estimator for $\mathbb{P}[g_P(x) = 1, y = 1]$ will be $a_P/n$). Since these are only estimates, which depend on the realizations of the random variables of interest, we cannot apply the definition of independence directly.

Frequentist hypothesis testing addresses this problem as follows. We first choose an appropriate test statistic $T$ and compute its distribution $\mathbb{P}[T = t|H_0]$ under the null hypothesis $H_0$ that the random variables $g_P(X)$ and $Y$ are independent. Then, if we call $t$ the value that the test statistic assumes on the data, we compute the corresponding *p-value*, that is the probability that under the null hypothesis $H_0$ the test $T$ takes a value at least as extreme as $t$, i.e. a value representing a statistical association that is at least as strong as the one represented by $t$. If the p-value $p$ is smaller or equal to a predetermined threshold $\alpha$, that is if $p \leq \alpha$, then we call $g_P(X)$ and $Y$ statistically associated. The threshold $\alpha$ can be seen as the type I error of the association test, that is the probability that the two random variables are declared associated when they in fact are not.

A common test statistic for association testing is Fisher's exact test, which is based on the fact that the conditional probability distribution of $a_P$ given the marginals, under the null hypothesis of independence $H_0 = g_P(X) \perp\!\!\!\perp Y$, is a hypergeometric distribution.

The following fact [20] states that there is a monotonically decreasing function $\psi$ that lower bounds the minimum attainable p-value for Fisher's exact test.

**Fact 1.** *Let $n_1 \leq n - n_1$. Then the minimum attainable p-value for Fisher's exact test $p_{P',\min}$ is lower bounded by the monotonically decreasing function*

$$\psi(r_P) = \begin{cases} \binom{n_1}{r_P}/\binom{n}{r_P} & \text{if } 0 \leq r_P < n_1 \\ 1/\binom{n}{n_1} & \text{if } n_1 \leq r_P \leq n. \end{cases}$$

We remark that, although our methods are presented using Fisher's exact test, they can be easily adapted to incorporate other tests, such as Pearson's $\chi^2$ test.

## B. PROOFS OF SECTION III

**Lemma 1.** *Let $\sigma \in [\![0, n]\!]$ be a given support value such that $\psi(\sigma)C_{m(\sigma)} > \alpha$. Let $p_1, p_2, \dots, p_{m(\sigma)}$ be the p-values for the patterns in $M(\sigma)$ in increasing order. Then any pattern $P_i$ with support $\sigma$ would not be deemed significant by the* BYS *procedure if one applied it to the patterns in $M(\sigma)$.*

*Proof.* If there is no such pattern, the claim holds by vacuity. If one such pattern $P_i$ exists, then its p-value is $p_i \geq \psi(\sigma) > \alpha/C_{m(\sigma)} \geq \frac{i}{m(\sigma)C_{m(\sigma)}}\alpha$ and it is not deemed as significant by the procedure. □

**Lemma 2.** *Let $\sigma \in [\![0, n]\!]$ be a given support value, and let there be a $\sigma' > \sigma$ such that $\psi(\sigma')\frac{m(\sigma)C_{m(\sigma)}}{m(\sigma')+1} > \alpha$. Let $p_1, p_2, \dots, p_{m(\sigma)}$ be the p-values for the patterns in $M(\sigma)$ in increasing order. Let $p_{\hat{i}}$ be the smallest p-value such that its corresponding pattern $P_{\hat{i}}$ is not testable at level $\psi(\sigma')$, if any. Then $P_{\hat{i}}$ would not be deemed significant by the* BYS *procedure if one applied it to the patterns in $M(\sigma)$.*

*Proof.* If there is no such $P_{\hat{i}}$, the claim holds by vacuity. If $P_{\hat{i}}$ exists, it would not be deemed significant by the BYS procedure on $M(\sigma)$ if $p_{\hat{i}} > \frac{i}{m(\sigma)C_{m(\sigma)}}\alpha$. Since $P_{\hat{i}}$ is non-testable at level $\psi(\sigma')$, we have $p_{\hat{i}} > \psi(\sigma')$. Moreover, since $p_{\hat{i}}$ is the first p-value among the non-testable ones at level $\psi(\sigma')$, $\hat{i}$ cannot be higher than $m(\sigma') + 1$ (but could also be lower). Then $\psi(\sigma')\frac{m(\sigma)C_{m(\sigma)}}{m(\sigma')+1} > \alpha$ implies that $p_{\hat{i}} > \frac{i}{m(\sigma)C_{m(\sigma)}}\alpha$ and that $P_{\hat{i}}$ would not be deemed significant if we applied the procedure to $M(\sigma)$. □

**Theorem 1.** *Let $\mathcal{P}_{\text{sig},\sigma^*}$ be the set of significant patterns obtained by applying the* BYS *procedure to the set $M(\sigma^*)$, with $\sigma^*$ the minimum support such that (i) $\psi(\sigma^*)C_{m(\sigma^*)} \leq \alpha$, and (ii) $\psi(\sigma')\frac{m(\sigma^*)C_{m(\sigma^*)}}{m(\sigma')+1} \leq \alpha,\ \forall \sigma' > \sigma^*$. Then, applying the* BYS *procedure to any other set $M(\sigma'')$ with $\sigma'' < \sigma^*$ would yield a set $\mathcal{P}_{\text{sig},\sigma''} \subseteq \mathcal{P}_{\text{sig},\sigma^*}$.*

*Proof. Case 1)* Let (i) not hold at $\sigma^* - 1$. Then, since $\psi(\sigma)C_{m(\sigma)}$ is monotonic decreasing in $\sigma$, it does not hold for any $\sigma'' < \sigma^*$.

Call $P_1, \dots, P_{m(\sigma'')}$ the patterns in $M(\sigma'')$ in increasing order of their p-values. Then, as stated by Lemma 1, all patterns that are deemed as significant by the BYS procedure on $M(\sigma'')$ must belong to $M(\sigma^*)$. Consider one such pattern $P_i$, then it would be in position $i$ also in the ordering of patterns considered by the BYS procedure on $M(\sigma^*)$. Then if $p_i \leq \frac{i}{m(\sigma'')C_{m(\sigma'')}}\alpha$, it holds that also $p_i \leq \frac{i}{m(\sigma^*)C_{m(\sigma^*)}}\alpha$, and $P_i$ would be deemed significant by the BYS procedure on $M(\sigma^*)$.

*Case 2)* Let (ii) not hold at $\sigma^* - 1$. Then there exists a $\sigma'$ for which $\psi(\sigma')\frac{m(\sigma^*-1)C_{m(\sigma^*-1)}}{m(\sigma')+1} > \alpha$. Moreover, note that $\psi(\sigma')\frac{m(\sigma)C_{m(\sigma)}}{m(\sigma')+1}$, for a fixed $\sigma'$, is a monotonically decreasing function of $\sigma$. Then, for any $\sigma'' < \sigma^*$, we have that $\psi(\sigma')\frac{m(\sigma'')C_{m(\sigma'')}}{m(\sigma')+1} > \alpha$.

Call $P_1, \dots, P_{m(\sigma'')}$ the patterns in $M(\sigma'')$ in increasing order of their p-values. Then only patterns $P_i$ in $M(\sigma')$ such that there is no pattern $P_j \in M(\sigma'') \setminus M(\sigma')$ with $j < i$ could be deemed as significant, since, due to Lemma 2, we have that the most significant pattern in $M(\sigma'') \setminus M(\sigma')$ would not be deemed significant, along with all the subsequent patterns.

We show that for these patterns, if they are deemed significant by the BYS procedure on $M(\sigma'')$, then they are deemed significant also by the BYS procedure on $M(\sigma^*)$. Consider one such pattern $P_i$. Since there are no patterns in $M(\sigma'') \setminus M(\sigma')$ before $P_i$, it would be in position $i$ also in the ordering of patterns considered by the BYS procedure on $M(\sigma^*)$. Then if $p_i \leq \frac{i}{m(\sigma'')C_{m(\sigma'')}}\alpha$, it holds that also $p_i \leq \frac{i}{m(\sigma^*)C_{m(\sigma^*)}}\alpha$, and $P_i$ would be deemed significant by the BYS procedure on $M(\sigma^*)$. □

**Theorem 2.** *Consider the Benjamini-Yekutieli step-down procedure. In particular, suppose to have $m$ hypotheses $H_1, \dots, H_m$ ordered such that their corresponding p-values are in increasing order, i.e. $p_1 \leq \dots \leq p_m$. The procedure rejects hypotheses $H_1, \dots, H_k$, with $k = \min\{i : p_i > \frac{i}{mC_m}\alpha\} - 1$. Then, the procedure controls the FDR at level less or equal to $\alpha$.*

*Proof.* We simply adapt the proof in [2]. Let $q_i = \frac{i}{mC_m}\alpha$. Let $m_0$ be the number of true null hypotheses and $m_1$ the number of false null hypotheses. Let $I_0 = \{i_0, \dots, i_{m_0}\}$ be the indexes of the true null hypotheses. Let $A_{v,s}$ denote the event that the BYS procedure rejects exactly $v$ true and $s$ false hypotheses. The FDR is then

$$FDR = \sum_{s=0}^{m_1}\sum_{v=1}^{m_0}\frac{v}{v+s}\mathbb{P}[A_{v,s}]$$

For fixed $v$ and $s$, let $\omega$ be a subset of $I_0$ of size $v$, and $A_\omega$ the event in $A_{v,s}$ that the $v$ true null hypotheses rejected are $\omega$. Note that $\mathbb{P}[(p_i \leq q_{v+s}) \cap A_\omega] = \mathbb{P}[A_\omega]$ if $i \in \omega$ and 0 otherwise. Indeed, $H_i$ is falsely rejected if and only if $p_i \leq q_{v+s}$. Then we have

$$\begin{aligned}\sum_{i\in I_0}\mathbb{P}[(p_i \leq q_{v+s}) \cap A_{v,s}] &= \sum_{i\in I_0}\sum_{\omega}\mathbb{P}[(p_i \leq q_{v+s}) \cap A_\omega] = \\ &= \sum_{\omega}\sum_{i\in I_0}\mathbb{1}[i \in \omega]\mathbb{P}[A_\omega] = \\ &= \sum_{\omega} v\mathbb{P}[A_\omega] = v\mathbb{P}[A_{v,s}]\end{aligned}$$

We then have

$$\begin{aligned}FDR &= \sum_{s=0}^{m_1}\sum_{v=1}^{m_0}\frac{v}{v+s}\left(\sum_{i\in I_0}\mathbb{P}[(p_i \leq q_{v+s}) \cap A_{v,s}]/v\right) \\ &= \sum_{i\in I_0}\sum_{s=0}^{m_1}\sum_{v=1}^{m_0}\frac{1}{v+s}\mathbb{P}[(p_i \leq q_{v+s}) \cap A_{v,s}]\end{aligned}$$

For $i \in I_0$, let $\mathbf{P}^{(i)}$ be the remaining $m-1$ p-values after dropping $p_i$. Let $C^{(i)}_{v,s}$ be the event in which if $H_i$ is rejected then $v-1$ true null hypotheses and $s$ false hypotheses are rejected with it. Then we have

$$(p_i \leq q_{v+s}) \cap A_{v,s} = (p_i \leq q_{v+s}) \cap C^{(i)}_{v,s},$$

as they both correspond to the event in which $H_i$ is rejected *and* $v-1$ true null hypotheses and $s$ false hypotheses are rejected with it.

Let $k = v + s$. Let $C^{(i)}_k = \bigcup\{C^{(i)}_{v,s} : v + s = k\}$. Then we have

$$FDR = \sum_{i\in I_0}\sum_{k=1}^{m}\frac{1}{k}\mathbb{P}[(p_i \leq q_{v+s}) \cap C^{(i)}_{v,s}]$$

Let $p_{ikj} = \mathbb{P}\left[\left(p_i \in \left[\frac{j-1}{mC_m}\alpha, \frac{j}{mC_m}\alpha\right]\right) \cap C^{(i)}_k\right]$. Then

$$\begin{aligned}FDR &= \sum_{i\in I_0}\sum_{k=1}^{m}\frac{1}{k}\sum_{j=1}^{k}p_{ikj} = \sum_{i\in I_0}\sum_{j=1}^{m}\sum_{k=j}^{m}\frac{1}{k}p_{ikj} \leq \\ &\leq \sum_{i\in I_0}\sum_{j=1}^{m}\sum_{k=j}^{m}\frac{1}{j}p_{ikj} \leq \sum_{i\in I_0}\sum_{j=1}^{m}\frac{1}{j}\sum_{k=1}^{m}p_{ikj} = \\ &= \sum_{i\in I_0}\sum_{j=1}^{m}\frac{1}{j}\mathbb{P}\left[\left(p_i \in \left[\frac{j-1}{mC_m}\alpha, \frac{j}{mC_m}\alpha\right]\right) \cap \left(\bigcup_{k=1}^{m}C^{(i)}_k\right)\right] \\ &= \sum_{i\in I_0}\sum_{j=1}^{m}\frac{1}{j}\mathbb{P}\left[p_i \in \left[\frac{j-1}{mC_m}\alpha, \frac{j}{mC_m}\alpha\right]\right]\end{aligned}$$

Let $P_{a,b} = \mathbb{P}\left[p_i \in \left[\frac{a}{mC_m}\alpha, \frac{b}{mC_m}\alpha\right]\right]$. Then,

$$FDR = \sum_{i \in I_0} \sum_{j=1}^{m} \frac{1}{j} P_{(j-1),j} =$$

$$\sum_{i \in I_0} \left( \sum_{j=1}^{m-1} \left( \frac{1}{j} - \frac{1}{j-1} \right) P_{0,j} \; + \frac{P_{0,m}}{m} \right) =$$

$$= \sum_{i \in I_0} \left( \sum_{j=1}^{m-1} \frac{1}{j+1} \frac{P_{0,j}}{j} \; + \frac{P_{0,m}}{m} \right) \leq$$

$$\leq \sum_{i \in I_0} \left( \sum_{j=1}^{m-1} \frac{1}{j+1} \frac{\alpha}{mC_m} \; + \frac{\alpha}{mC_m} \right) =$$

$$= \sum_{i \in I_0} \frac{\alpha}{mC_m} C_m = \frac{m_0}{m} \alpha \leq \alpha.$$

□

### C. The minePerm procedure

In this section, we detail how the procedure MINEPERM$(\mathcal{D}, \sigma)$ computes the quantities used to calculate the FDR estimator. The pseudocode of such procedure is given in Algorithm 3.

**Algorithm 3:** MINEPERM

```
1  Function MINEPERM(D, σ):
2      initialize R', r', R_β, m' to all 0's
3      PROCESS(∅)
4      r, R^(j) = prefix sums for r and R^(j), ∀j
5      m = suffix sums for m
6      R_β[i] = β-quantile for R[i], for
         i = ⌊−log_{1+ζ} ψ(σ)⌋ + 1 to i_max
7      return R, r, R_β, m
8  Function PROCESS(P):
9      compute r_P, a_P and p-value p_P using y
10     m'[r_P] ++,   r' [⌊− log_{(1+ζ)}(p_P)⌋] ++
11     for j = 1 to K do
12         compute a_P^(j) and p-value p_P^(j) using y^(j)
13         R'^(j) [⌊− log_{(1+ζ)}(p_P^(j))⌋] ++
14     for each P' ∈ children(P) do
15         if r_{P'} ≥ σ
16             PROCESS(P)
```

The algorithm keeps track of the following quantities. $m'[i]$ stores the number of patterns with support exactly $i$, $r'[i]$ stores the number of patterns that have p-value (computed using the non-permuted labels) between $(1+\zeta)^{-i}$ and $(1+\zeta)^{-(i+1)}$ and $R'^{(j)}[i]$ stores the number of patterns hat have p-value (computed using the $j$-th permuted labels) between $(1+\zeta)^{-i}$ and $(1+\zeta)^{-(i+1)}$. The procedure starts by exploring the pattern enumeration tree from the root, $\emptyset$. For each explored pattern $P$, it computes (Line 9) $r_P$ as well as $a_P$ and the p-value $p_P$, using the non-permuted labels $y$. Then $m'[r_P]$ is increased by 1, and $r'[\lfloor -\log_{(1+\zeta)}(p_P) \rfloor]$ is increased by 1. Then, for each of the $K$ independent label permutations, which have already been initialized by the main procedure, it computes (Line 12-14) the same quantities using the permuted labels $y^{(j)}$ and increases $R'^{(j)}[\lfloor -\log_{(1+\zeta)}(p_P^{(j)}) \rfloor]$ by 1. Then, for each of the children $P'$ of $P$ in the pattern enumeration tree, if not prunable, the procedure is called recursively on $P'$.

Upon completing the recursion, the quantities $r$ and $R^{(j)}$ are computed on Line 4 as the prefix sums of $r'$ and $R'^{(j)}$ respectively. Similarly, $m$ is computed as the suffix sum of $m'$. Finally, on Line 6, the procedure computes, for $i = \lfloor -log_{(1+\zeta)}\psi(\sigma) \rfloor + 1, \ldots, i_{\max}$ the $\beta$-quantile (with respect to the $K$ independent realizations) of the $R^{(j)}[i]$'s. The correctness of the procedure is proved in the Appendix.

The following theorem establishes the correctness of MINEPERM$(\mathcal{D}, \sigma)$.

**Theorem 3.** *For each $i \in [\![\sigma, n, ]\!]$, after running procedure* MINEPERM$(\mathcal{D}, \sigma)$*, we have that $m[i] = m(i)$, the number of patterns with support at least $i$. Similarly, for each $i = \lfloor -log_{(1+\zeta)}\psi(\sigma) \rfloor + 1, \ldots, i_{\max}$, we have that $r[i] = r((1+\zeta)^{-i})$, $R^{(j)}[i] = R^{(j)}((1+\zeta)^{-i}), \forall j$ and $R_\beta[i] = R_\beta((1+\zeta)^{-i})$, as defined in Section II-C2.*

*Proof.* We have that, for each $i \in [\![\sigma, n, ]\!]$, $m'[i]$ stores the number of patterns with support $i$. Indeed, the procedure explores all the patterns with $r_P \geq \sigma$, as thanks to the antimonotonicity property, if a pattern is pruned in Lines 15-17, then also all of its descendants have support less than $\sigma$. Then, setting $m[i] = \sum_{i'=i}^{n} m'[i']$, computed by taking the suffix sum of $m'$, yields $m[i] = |M(i)|$.

We now show that, for each $i = \lfloor -log_{(1+\zeta)}\psi(\sigma) \rfloor + 1, \ldots, i_{\max}$, $r'[i]$ stores the number of patterns that have p-value (computed using the non-permuted labels) between $(1+\zeta)^{-i}$ and $(1+\zeta)^{-(i+1)}$ and $R'^{(j)}[i]$ stores the number of patterns hat have p-value (computed using the $j$-th permuted labels) between $(1+\zeta)^{-i}$ and $(1+\zeta)^{-(i+1)}$. All the pruned patterns cannot contribute to the above quantities. Indeed, consider one such pattern $P$. Its p-value must satisfy $p_P \geq \psi(\sigma)$. Then, it can only contribute to $r'(\delta)$ for $\delta = (1+\zeta)^{-i} \geq \psi(\sigma)$. The same holds for the $R'^{(j)}$'s. Finally, computing the prefix sums of $r'$ yields $r[i] = \sum_{i'=0}^{i} r'[i'] = |\{P \in \mathcal{P} : p_P \leq (1+\zeta)^{-i}\}|$. A similar argument yields the result for the $R^{(j)}$'s and, and consequently, for $R_\beta$. □

The procedure MINEPERM gets called at most $O(\log n)$ times, and each call takes $O(|M(\sigma)|(n+K))$ time. Indeed, for each explored pattern it has to compute its support and the $K$ p-values, which can be obtained in $O(1)$ time for each permutation by precomputing them.

### D. Additional pseudocodes

Algorithm 4 details the MINETESTABLEPATTERNS procedure, which is used by FASM to retrieve the testable patterns on which to run the Benjamini-Yekutieli procedure.

Algorithm 5 details the MINESIGNPATTERNS procedure, which is used by FAST-YB to retrieve the significant patterns after having obtained the significance threshold $\hat{\delta}_{\text{yb}}$.

**Algorithm 4:** MINETESTABLEPATTERNS

**Function** MINETESTABLEPATTERNS($\mathcal{D}, \sigma$)**:**
  initialize $testPatterns$ to an empty list
  PROCESS($\emptyset$)
  **return** $testPatterns$
**Function** PROCESS($P$)**:**
  compute $r_P$
  append $P$ to $testPatterns$
  **for each** $P' \in$ children($P$) **do**
    **if** $r_{P'} \geq \sigma$
      PROCESS($P$)

**Algorithm 5:** MINESIGNPATTERNS

**Function** MINESIGNPATTERNS($\mathcal{D}, \delta$)**:**
  initialize $signPatterns$ to an empty list
  PROCESS($\emptyset$)
  **return** $signPatterns$
**Function** PROCESS($P$)**:**
  compute $r_P$, $a_P$ and p-value $p_P$ using labels $y$
  **if** $p_P \leq \delta$
    append $P$ to $signPatterns$
  **for each** $P' \in$ children($P$) **do**
    **if** $\psi(r_{P'}) \leq \delta$
      PROCESS($P$)

TABLE II
KEY PROPERTIES OF DATASETS. $|\mathcal{D}|$ DENOTES THE DATASET SIZE, $|\mathcal{I}|$ THE NUMBER OF ITEMS ON WHICH THE ITEMSETS ARE BUILT, AVG. $|x|$ THE AVERAGE ITEMSET LENGTH AND $n_1/n$ THE MINORITY CLASS RATIO.

| Name | $\lvert\mathcal{D}\rvert$ | $\lvert\mathcal{I}\rvert$ | avg. $\lvert x\rvert$ | $n_1/n$ |
|---|---|---|---|---|
| svmguide3 | 1243 | 44 | 21.9 | 0.23 |
| mushroom | 8124 | 118 | 22.0 | 0.48 |
| phishing | 11055 | 813 | 43.0 | 0.44 |
| breast_cancer | 12773 | 1129 | 6.7 | 0.09 |
| a9a | 32561 | 247 | 13.9 | 0.24 |
| ijcnn1 | 91701 | 44 | 13.0 | 0.10 |
| codrna | 271617 | 16 | 8.0 | 0.33 |
| covtype | 581012 | 64 | 11.9 | 0.49 |

TABLE III
NUMBER OF SIGNIFICANT PATTERNS USING THE BYS PROCEDURE AT THE TESTABILITY THRESHOLD $\sigma^*$, AT THE THRESHOLD $\sigma^{gil}$ USED BY [8] AND AT $\sigma^* - 1$. BOLD NUMBERS DENOTE THE ROW'S HIGHEST ENTRY.

| Dataset | $\lvert\mathcal{P}_{\text{sig},\sigma^*}\rvert$ | $\lvert\mathcal{P}_{\text{sig},\sigma^{gil}}\rvert$ | $\lvert\mathcal{P}_{\text{sig},\sigma^*-1}\rvert$ |
|---|---|---|---|
| svmguide3 | **115091** | 114674 | 115089 |
| mushroom | **120375** | 81982 | **120375** |
| phishing | **739421791** | 670558206 | 739411643 |
| breast_cancer | **0** | **0** | **0** |
| a9a | 710058 | **711665** | 706134 |
| ijcnn1 | **1134331** | 1132255 | 1134296 |
| codrna | **4216** | **4216** | **4216** |
| covtype | **646438** | 629085 | 646410 |

## E. ADDITIONAL EXPERIMENTAL RESULTS

In this section we report some additional experimental results that were omitted from the main paper due to space constraints.

### A. Datasets

Table II reports some details for the datasets we used in the experiments in the main paper.

These datasets are originally endowed with labels and are therefore prone to be studied in the context of significant pattern mining. Moreover, this selection includes datasets with a high number of samples (e.g. `covertype`), high number of frequent patterns (e.g. `phishing`) and high number of items on which the itemsets are built (e.g. `breast_cancer`).

### B. Comparison of testability thresholds for FASM

In this section we compare the number of patterns, which are reported in Table III, that are deemed as significant by the BYS procedure used by FASM when it is applied to different sets of testable patterns. In particular, we report (i) the number of significant patterns $\mathcal{P}_{\text{sig},\sigma^*}$ using $\sigma^*$ as the testability threshold, as suggested by Theorem 1, (ii) the number of significant patterns $\mathcal{P}_{\text{sig},\sigma^{gil}}$ using Tarone's method to select the testability threshold $\sigma^{gil}$, as described in Gilbert's heuristic procedure [8], and (iii) the number of significant patterns $\mathcal{P}_{\text{sig},\sigma^{gil}}$ using $\sigma^* - 1$ as the testability threshold.

First, we note that on all datasets $\mathcal{P}_{\text{sig},\sigma^*-1} \subseteq \mathcal{P}_{\text{sig},\sigma^*}$, and that therefore the power at $\sigma^*$ is higher, confirming the results of Theorem 1. Moreover, we show that on almost all datasets, using $\sigma^*$ as the testability threshold yields the highest statistical power. On `a9a`, the higher number of patterns in $M(\sigma^*)$ compared to the ones in $M(\sigma^{gil})$, which is a subset of $M(\sigma^*)$, leads to a slightly stricter significance threshold and thus lower power. We point out that the patterns in $\mathcal{P}_{\text{sig},\sigma^{gil}} \setminus \mathcal{P}_{\text{sig},\sigma^*}$ are the ones with p-value between the two significance thresholds, and are thus the least significant ones.

### C. Additional details on synthetic data generation

We describe in more detail the generated datasets we used in Section V-C.

The synthetic dataset is built on a ground set $\mathcal{I}$ of 50 items. The dataset are created by inserting in each itemset the items of $\mathcal{I}$, each with probability $q = 0.2$. We also define a set $\mathcal{I}_0 \subset \mathcal{I}$ of *special* items. Then, the labels are generated to be 1 with probability $p_1 = 0.5$ if in the corresponding sample there are no items from $\mathcal{I}_0$, and with probability $p_2 = 0.7$ if there is at least one item belonging to $\mathcal{I}_0$.

Then the truly significant patterns are all and only the itemsets that have at least one item belonging to $\mathcal{I}_0$, which makes it easy to identify true and false positives. We generated such synthetic datasets for multiple values of $|\mathcal{I}_0|$ to represent multiple scenarios. Indeed, when $|\mathcal{I}_0| = 0$ the global null hypothesis holds, that is there are no truly significant patterns. When the number of special items is positive, there are patterns that are statistically associated with the labels, and their number grows with $|\mathcal{I}_0|$.

### D. Sensitivity to $\zeta$

Our algorithm FAST-YB tests significance thresholds in a geometric sequence $\Delta = \{(1+\zeta)^{-i} : i = 0, \ldots, i_{\max}\}$, for $\zeta > 0$. Smaller $\zeta$'s lead to more precision in the selection

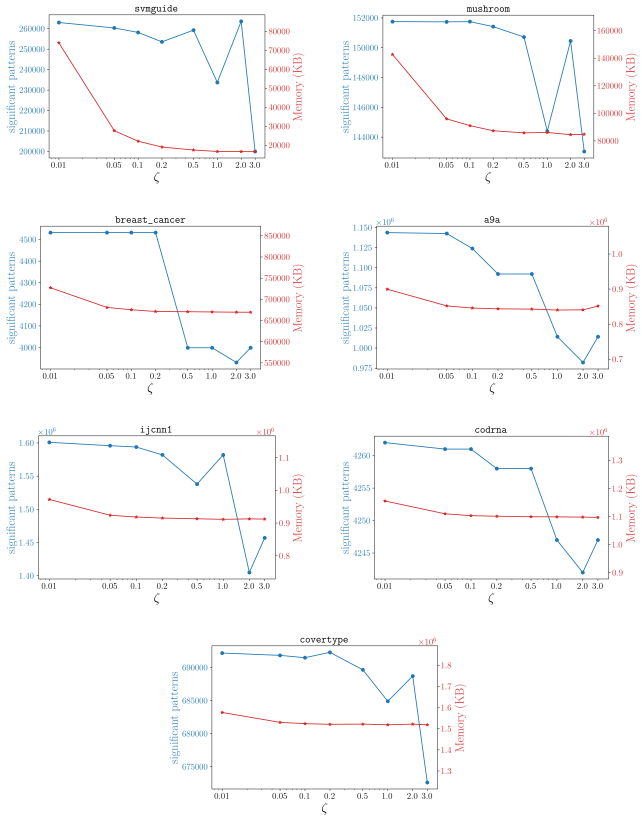


Fig. 4. Number of returned significant patterns and memory usage of FAST-YB for various values of $\zeta$. We fix $K = 1000$.

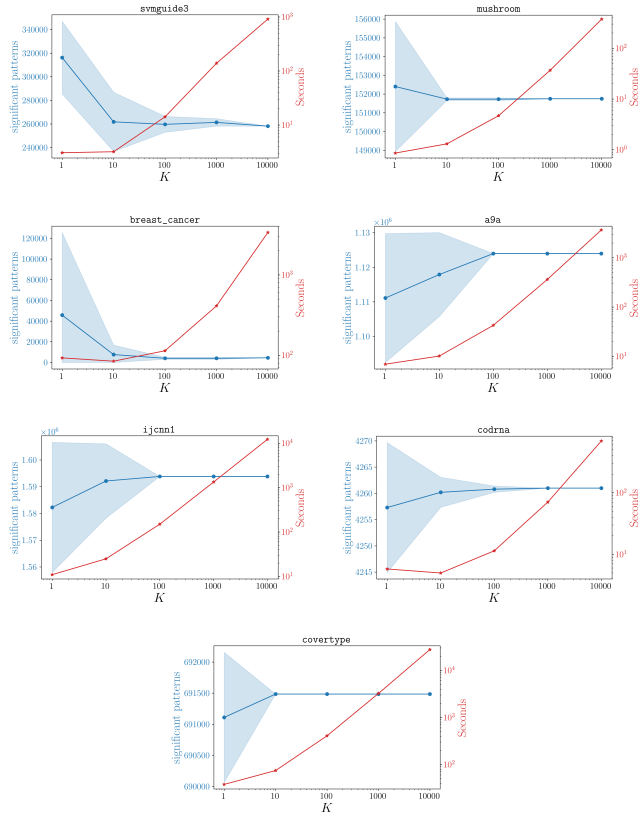


Fig. 5. Number of returned significant patterns and running times of FAST-YB for various values of $K$. We fix $\zeta = 1.1$.

of the significance threshold, but incur in a computational overhead, especially in terms of memory consumption. In our experiments we fix $1+\zeta = 1.1$. Figure 4 shows the number of significant patterns and the memory consumption for several values of $\zeta$. As shown by the plots, choosing $1 + \zeta = 1.1$ yields a number of significant patterns that is on par with the one obtained by setting $1+\zeta = 1.01$, without incurring in the large memory overhead. In fact, on most datasets choosing $1 + \zeta$ up to 1.5 yields good results.

### E. Sensitivity to $K$

Our algorithm FAST-YB computes the local FDR estimator as an average over $K$ independent permutations of the labels in order to reduce the variance. Therefore, high values of $K$ yield more consistent values for the FDR estimator, which in turn yield more consistent numbers of significant patterns, at the cost of increased running times. Figure 5 reports the number of significant patterns returned by FAST-YB, as means and standard deviations over 10 runs with different seeds, and the average running times for various values of $K$. We fix $\zeta = 1.1$.

Setting $K = 1$ results in running times on par with the ones of FASM, as we consider a single permutation of the labels, but yields extremely noisy results. On the other end of the spectrum, setting $K = 10^4$ results in extremely stable FDR estimators, but the running times are several orders of magnitude worse compared to the one of FASM.

Setting $K = 10^3$ provides a good trade-off between stability of the results and moderate running times, so we select this value in our experiments.

### F. Sensitivity to $\Sigma$

Both our algorithms FASM and FAST-YB mine patterns at frequency thresholds $\sigma$ in a set $\Sigma$, which in our experiments is set to a geometrically decreasing schedule $\Sigma_\psi = \{\lfloor 2^{-k}\sigma_{\text{start}} \rfloor : k \geq 0\}$, with $\sigma_{\text{start}}$ the largest $\sigma$ such that $\psi(\sigma) > \alpha$, as described in Section III-B. While the choice of $\Sigma$ has no effect on the output of the algorithms, it can have a substantial impact on running times.

Other choices can be the set $\Sigma_{\log} = \{\lfloor 2^{-k} n \rfloor : k \geq 0\}$, which tests thresholds in a geometric sequence starting from the number of samples $n$, and $\Sigma_{\text{all}} = [\![1, n]\!]$, which tests in decreasing order all the thresholds from $n$ to 1.

Figure 6 shows, on the easiest datasets, `svmguide` and `mushroom`, the running times with different choices of $\Sigma$. Note that on more challenging datasets, using $\Sigma_{\text{all}}$ yields running times so high that both algorithms cannot terminate in several hours, so we omit them from the plot.

As shown by the plots, both the choices of $\Sigma_\psi$ and of $\Sigma_{\log}$ offer a substantial advantage over the naive choice of $\Sigma_{\text{all}}$ used by [22]. In fact, using $\Sigma_\psi$ yields the best performance, as it allows to call the miner a single time for FASM at a

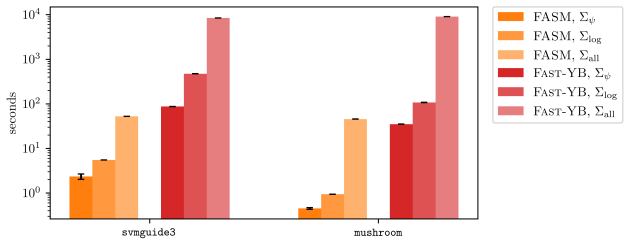


Fig. 6. Running times of FASM and FAST-YB for different choices of $\Sigma$.

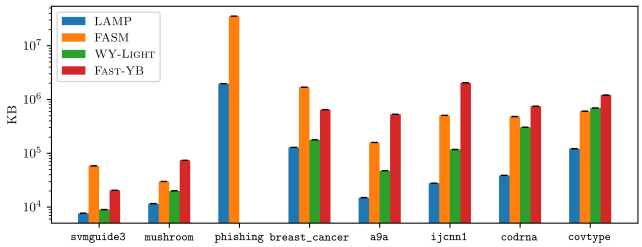


Fig. 7. Peak memory usage of LAMP, WY-LIGHT, FASM and FAST-YB when controlling the FDR at level $\alpha = 0.05$. For permutation-testing-based algorithms, we fix $K = 1000$.

threshold close to the optimal one, and only a couple of times for FAST-YB.

### G. Peak memory consumption

Figure 7 reports the peak memory usage of all four methods, as described in Section V-D. The experiments show that, although our methods use more memory than their FWER-controlling counterparts, the memory usage of our methods never exceeds 4GB of RAM, except for the `phishing` dataset. The gap in memory usage is due the need to keep testable patterns in memory, as well as some design choices, as for example we pre-compute a large number of p-values to speed up the computation.

### H. Additional datasets

The 8 datasets that we use in the main paper, reported in Table II, are originally endowed with labels and are therefore prone to be studied in the context of significant pattern mining.

We selected 4 additional commonly used itemset mining datasets due to their high number of samples and of frequent patterns, so that they can serve as additional stress tests for our methods. Since these datasets are not originally endowed with labels, the authors of [18] artificially labeled them by selecting the single item whose frequency is closer from below to $1/2$, removed the corresponding item from every sample, and use its appearance to define the target class label. We use such artificially labeled datasets, whose properties are reported in Table IV, for our additional analysis.

Figure 8 reports the comparison of the statistical power of our algorithms and of the baselines, similarly to Section V-B. As on the other datasets, FASM and FAST-YB have higher

TABLE IV
KEY PROPERTIES OF ADDITIONAL DATASETS. $|\mathcal{D}|$ DENOTES THE DATASET SIZE, $|\mathcal{I}|$ THE NUMBER OF ITEMS ON WHICH THE ITEMSETS ARE BUILT, AVG. $|x|$ THE AVERAGE ITEMSET LENGTH AND $n_1/n$ THE MINORITY CLASS RATIO. LABELS ARE OBTAINED ARTIFICIALLY AS THE APPEARANCE OF A SPECIFIC ITEM IN THE SAMPLE.

| Name | $\lvert\mathcal{D}\rvert$ | $\lvert\mathcal{I}\rvert$ | avg. $\lvert x\rvert$ | $n_1/n$ |
|---|---|---|---|---|
| `chess` | 3196 | 75 | 37 | 0.05 |
| `bms-web1` | 58136 | 60978 | 2.51 | 0.03 |
| `T10I4D100K` | 100000 | 870 | 10.1 | 0.08 |
| `T40I10D100K` | 100000 | 942 | 39.6 | 0.28 |

statistical power compared to the baselines, with the latter having the highest.

Moreover, Figure 9 reports the running times of all the methods on the additional datasets. Note that on `chess`, the permutation-testing-based methods failed to complete in 24 hours.

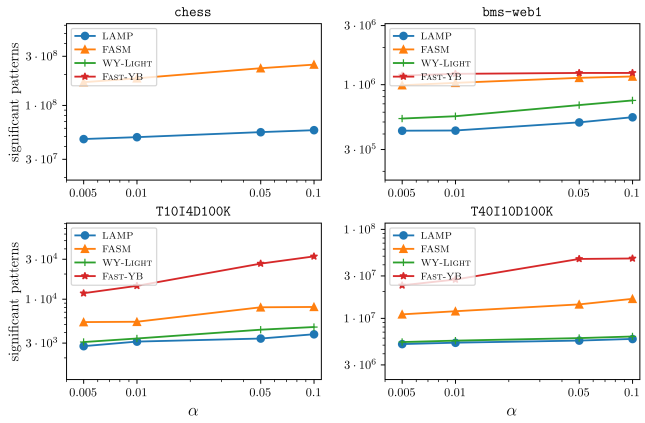


Fig. 8. Number of patterns deemed as significant by LAMP, WY-LIGHT, FASM and FAST-YB when controlling the FDR at level $\alpha \in \{0.1, 0.05, 0.01, 0.005\}$.

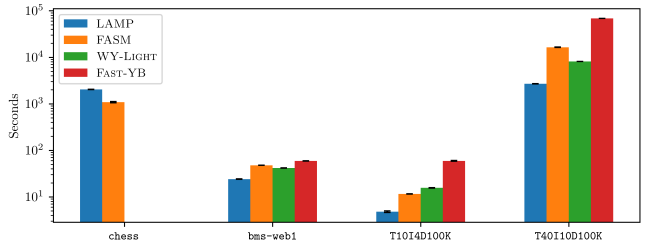


Fig. 9. Running times of LAMP, WY-LIGHT, FASM and FAST-YB when controlling the FDR at level $\alpha = 0.05$. For permutation-testing-based algorithms, we fix $K = 1000$.