Semi-supervised learning while controlling the FDR with an application to tandem mass spectrometry analysis

Jack Freestone[1*], Lukas Käll[2], William Stafford Noble[3,4], and Uri Keich[1*]

[1]School of Mathematics and Statistics F07, University of Sydney
[2]Science for Life Laboratory, KTH Royal Institute of Technology, Stockholm, Sweden
[3]Department of Genome Sciences, University of Washington
[4]Paul G. Allen School of Computer Science and Engineering, University of Washington

∗ Corresponding authors: J.Freestone@maths.usyd.edu.au, uri.keich@sydney.edu.au

Abstract

Canonical procedures to control the false discovery rate (FDR) among the list of putative discoveries rely on our ability to compute informative p-values. Competition-based approach offers a fairly novel and increasingly popular alternative when computing such p-values is impractical. The popularity of this approach stems from its wide applicability: instead of computing p-values, which requires knowing the entire null distribution for each null hypothesis, a competition-based approach only requires a single draw from each such null distribution. This drawn example is known as a "decoy" in the mass spectrometry community (which was the first to adopt the competition approach) or as a "knockoff" in the statistics community. The decoy is competed with the original observation so that only the higher scoring of the two is retained. The number of decoy wins is subsequently used to estimate and control the FDR among the target wins.

In this paper we offer a novel method to extend the competition-based approach to control the FDR while taking advantage of side information, i.e., additional features that can help us distinguish between correct and incorrect discoveries. Our motivation comes from the problem of peptide detection in tandem mass spectrometry proteomics data. Specifically, we recently showed that a popular mass spectrometry analysis software tool, Percolator, can apparently fail to control the FDR. We address this problem here by developing a general protocol called "RESET" that can take advantage of the additional features, such as the ones Percolator uses, while still theoretically and empirically controlling the FDR.

Keywords: proteomics, false discovery rate control, tandem mass spectrometry

1 Introduction

In the multiple testing problem we consider m null hypotheses $H_1, \ldots, H_m$ and our goal is to maximize the number of discoveries (rejections), subject to some control of the type I error (false discoveries). When m is large we typically control the false discovery rate (FDR), which is the expected value of the false discovery proportion (FDP) [2]. Canonically, we associate a test statistic T_i with each hypothesis H_i and compute p_i, the p-value of the observed value T_i. These p-values are then analyzed to determine the list of rejected hypotheses, so that the FDR is no more than the preselected threshold α.

The specific problem that motivated this research is the detection of the peptides that are present in a sample through the use of liquid chromatography tandem mass spectrometry (MS/MS) — a high-throughput method for detecting proteins in a biological sample. The approach involves first digesting the proteins to peptides that are then separated to some extent using liquid chromatography before entering a mass spectrometer. Inside, the peptides are subjected to two rounds of mass spectrometry. The first tries to further isolate the many copies of a specific peptide species (sequence) from other species that might co-elute with it. These copies are then subjected to a random fragmentation process and passed through a second round of mass spectrometry, which creates the fragmentation spectrum of the isolated peptide. Specifically, the spectrum records the abundance of each observed ion fragment characterized by its mass-to-charge ratio and corresponding intensity.

The (fragmentation) spectrum can be thought of as the fingerprint of the peptide that generated it, and our goal is to use those fingerprints to detect the peptides present in the sample (and hence to learn about the present proteins). Specifically, we have a relevant *target* database of peptides that could be in the sample, and we associate a null hypothesis with each: the peptide is not present in the sample. For example, when analyzing a human sample, our target database would typically contain all peptides encoded in the human proteome. Using a process that we will elaborate on in Section 3.1, we examine the spectra for evidence of the presence of each target peptide, summarizing it in a score / test statistic, Z_i.

Peptide detection is an atypical multiple testing problem because it is most commonly addressed without computing the p-values of the observed scores Z_i. Partly due to the difficulty in computing p-values, the mass spectrometry community instead settled on a competition-based approach to control the FDR [5]. Specifically, we construct a disjoint *decoy* database where each target peptide is replaced by a randomly shuffled or reversed version of itself (shared peptides are removed). We next use the spectra to assign each decoy peptide an evidence score $\tilde{Z}_i$ in the same way we used it to assign the score Z_i to the corresponding target peptide.

The target-decoy competition (TDC) approach starts by competing the scores of each target peptide and its associated decoy and keeping only the higher of the two, together with a label indicating whether the winning score corresponds to the target peptide ("target win") or the decoy peptide ("decoy win"). Because a decoy peptide is not expected to be in the sample, we can think of the score $\tilde{Z}_i$ as a single draw from the null distribution of H_i. Hence, for a true null H_i the target is equally likely to win as its decoy is. TDC takes advantage of this equality, using the number of decoy wins as an estimate of the number of false discoveries among the target wins, which subsequently allows TDC to estimate the FDR. This latter estimate is then leveraged to control the FDR, as explained in detail in Section 2.1 below.

The competition-based approach recently gained much attention in the statistics and machine learning communities after Barber and Candès used the same principle in their knockoff+ procedure to control the FDR in feature selection in linear regression [1]. While Barber and Candès' knockoff construction is significantly more elaborate than the construction of the analogous decoy peptides, the underlying procedure for competition-based FDR control is the same, regardless of whether it uses knockoffs or decoys. In either type of competition approach, the hypotheses are ordered by decreasing winning scores and all target wins above the cutoff, determined by the FDR threshold, are reported.

Subsequent to the development of TDC, it was recognized that if the hypotheses can be rescored using additional information so that the target wins are generally ranked higher, then TDC will report more discoveries. This idea was implemented, for example, in the widely used Percolator tool, which is a semi-supervised machine learning algorithm to re-rank the hypotheses using additional features such as the peptide's length or its presumed charge state [14].

Despite the popularity of Percolator, we have recently shown that in practice Percolator can fail to control the FDR [9]. The problem, in a nutshell, arises because Percolator uses decoys in two fundamentally different

ways: first to train a machine learning model to distinguish between correct and incorrect identifications, and second to estimate the FDR using the TDC procedure. These two uses make it difficult for Percolator's rescoring to retain the crucial underlying assumption that a true null hypothesis is equally likely to be a target or a decoy win, independently of its winning score. For further details about this problem, see Section 3.2.

In this paper we address Percolator's problem by offering a new approach for training a semi-supervised machine learning model in a way that does not break the FDR control. We achieve this goal by randomly placing each decoy win into one of two sets, where the first set is used for training the model and the second is reserved for estimating the number of false target discoveries. Accordingly, we refer to our new approach as **RESET** (REScoring via Estimating and Training). We establish that under a natural extension of TDC's key assumptions, RESET rigorously controls the FDR. We also use entrapment experiments to show that integrating RESET with Percolator (**Percolator-RESET**) empirically offers significantly more robust control of the FDR than Percolator does. Interestingly, in spite of its much tighter FDR control, Percolator-RESET gives up very little in terms of power (i.e., the number of discoveries) when compared with Percolator.

We also offer an extension of RESET that considers two decoys for every target. The use of a second decoy reduces RESET's variability, which is inherent to its random split of the decoy set. Moreover, integrating this enhanced version with Percolator delivers a slightly better trained version of Percolator-RESET, allowing the two-decoy version to deliver slightly more discoveries than the original Percolator and the single-decoy Percolator-RESET.

This paper focuses on the peptide detection problem; however, the two RESET procedures are more generally applicable to competition-based FDR control with side information. The latter problem, as well as the related problem of using side-information with p-value based control of the FDR, have been also studied recently in the statistics community with tools such as ZAP [18] AdaPT [17], Adaptive Knockoffs [24] offering finite sample FDR control. However, none of these tools are applicable in our context: ZAP assumes a parametric model where p-values can be computed, AdaPT addresses both the p-value and the competition-based contexts but only provides a meta scheme in the latter case, and, as we argue below, Adaptive Knockoffs is forbiddingly slow for a typical peptide detection problem, and moreover, when run to completion on a small dataset it offers significantly fewer discoveries than Percolator-RESET does.

2 The RESET (REScoring via Estimating and Training) procedure

2.1 Competition based FDR control (TDC)

Although we focus in this work on the peptide detection problem, we present our method in the following, more generally applicable context. Consider a setup where we have a list of proposed discoveries, of which only a subset are true (e.g., a list of all the target database peptides). Our goal is to report as many of the discoveries as possible while controlling the FDR, which in this case is the expected proportion of false discoveries (e.g., target peptides that are not in the sample) among the reported ones. Formally, we associate with each discovery a null hypothesis, H_i, that the discovery is false.

As in peptide-level TDC, we assume that each putative discovery is paired with a competing, artificial null-generated discovery (e.g., the paired decoy peptide). Both discoveries, which we refer to as the target and decoy discoveries, respectively, are scored, where a higher score indicates a discovery that is more likely to be true (e.g., a peptide's score summarizes the spectral evidence supporting its presence in the sample — more on that below). Let Z_i and $\tilde{Z}_i$ denote the target and decoy scores respectively, and note that we use the term decoys but in the context of variable selection the decoys are called *knockoffs*.

The competition approach starts by competing the paired scores to define a winning score $W_i = \max\{Z_i, \tilde{Z}_i\}$ and a label $L_i = \pm 1$, indicating whether the winning score corresponds to the target or the decoy win (ties are removed altogether, or randomly broken). Then, as all variants of TDC do, it uses the decoy wins to estimate the FDR among the corresponding target wins, which it subsequently uses to determine which winning target discoveries to report. The latter hinges on the following critical assumption.

Assumption 1. *Conditional on all the winning scores W, and on the labels L of the false null hypotheses, the labels of the true null hypotheses are independent uniformly distributed ± 1 random variables.*

It follows from the assumption that the number of decoy wins in the top k winning scores is an unbiased estimator of the number of false target discoveries among the same top k scores. Dividing this estimate by the complementary number of target wins in the top k winning scores gives an estimate of the FDR among those top target wins. To rigorously control the FDR at level α, we add 1 to the number of decoy wins before dividing by the number of target wins, and we look for k_0, the largest k for which this estimate is still $\leq \alpha$. Finally, we report all the target wins among the top k_0 peptides. This estimating/controlling step of TDC is further described in SeqStep+ (Algorithm S1 with $c = 1/2$ here), which in turn is essentially the Selective SeqStep+ procedure of [1]. It was shown that if Assumption 1 holds, then TDC rigorously controls the FDR [1, 13].

Importantly, the same approach will control the FDR for *any* ordering of the winning scores that does not peek at the target/decoy win labels [1]. In particular, if we can find a better ordering of the winning scores, that is, one that is likely to rank correct discoveries higher up, then applying the same estimating/controlling step that TDC does (SeqStep+ with $c = 1/2$) will yield more discoveries while still controlling the FDR. This reranking, which is the focus of this work, is what **feature-augmented TDC**, or TDC with side-information tries to achieve.

Specifically, in this setting we assume that we can associate with each discovery a list of features that may help us distinguish between correct and incorrect discoveries. For example, in the peptide detection problem we can associate with the target or decoy peptides inherent features, like their length, as well as features they inherit from the spectra that are used in detecting them, like the presumed charge state (more on that below). Importantly, the features should not be able to help us distinguish between the target and its decoy when the target discovery is incorrect. Indeed, a feature like this would compromise our approach, similar to how it can foil Percolator's FDR control [3, 11]. In the feature-augmented setting, we also need to strengthen Assumption 1 so that it holds conditional on the features as well (see Assumption 2 below).

Note that the same comment made earlier about TDC applies to feature-augmented TDC: applying TDC's SeqStep+ to any ordering of the discoveries that does not peek at the target-decoy win labels controls the FDR. We next describe two new procedures that take advantage of this fact by offering an improved ordering in this general context.

2.2 The RESET (REScoring via Estimating and Training) procedure

RESET was developed to take advantage of the additional features to rescore the discoveries so that correct discoveries will generally rank higher. While originally developed to integrate with the Percolator algorithm, RESET is generally compatible with any semi-supervised machine learning approach that takes advantage of the target/decoy labels in a similar fashion. RESET is described through the following main steps, with further details given in Algorithms S2a-S2b.

1. RESET begins with a TDC-like competition step: from each pair of target and decoy discoveries we record the maximal score W, together with a label L indicating whether this was a target or a decoy win. RESET also records the vector of features $\mathbf{x}$ associated with the winning discovery. As an aside, note that there is some flexibility in how the winning score is defined—for example, it can be the absolute value of the difference in the scores [1]—but here we focus on the maximum.

2. RESET then defines the **training** decoy set by independently randomly assigning to it each winning decoy with probability s (we used $s = 1/2$ throughout). We refer to the remaining set of winning discoveries (targets, as well as non-training decoys) as **pseudo** targets.

3. Next, RESET applies a user-selected semi-supervised machine learning model whose input includes the following:

   - a negative set containing the scores and features of the training decoys, and
   - a set containing the winning scores and features of the pseudo targets but *without* the labels distinguishing between targets and decoys.

The output of this step is a rescoring of the training decoys and pseudo targets, ideally ranking many of the correct pseudo target discoveries higher than they were ranked originally.

4. Once the training is complete, the training decoys are thrown out, and the pseudo targets are ordered according to the learned scores, with their corresponding original target/decoy winning labels revealed. Some of the revealed labels will correspond to decoy wins—we refer to those as **estimating** decoys—and the others correspond to target wins. RESET applies SeqStep+, as in TDC, but uses the number of estimating decoys in the top k pseudo target scores to estimate the number of false discoveries among the targets in the same top k scores. The key difference with TDC is that because in RESET we use approximately half of the decoys for training purposes, each estimating decoy estimates more than one false target discovery. To adjust for that difference, we multiply the number of decoys (plus 1) by 2. The rest of SeqStep+ works exactly as in TDC (Algorithm S1 with $\frac{c}{1-c} = 2$, or $c = 2/3$).

2.3 Using an extra decoy: an enhancement of RESET

We introduce an enhancement of RESET that makes use of an extra decoy for each target. It goes through the same steps as RESET with the following modifications:

- In step 1, the competition now takes place between all sets of triplets: the target and its two decoys. We record a target/decoy label L indicating whether the target or one of its two associated decoys is equal to the maximum of the three scores W, corresponding to a target or decoy win, respectively (again ties are broken randomly). In addition to the winning score and a target/decoy win label, we keep the vector of features $\mathbf{x}$ of the winning discovery. This multiple competition method is essentially the same as the *max method* introduced by Emery et al. [6].
- When applying SeqStep+, each estimating decoy now estimates a single false target discovery (as in TDC itself), so no adjustment is made to the observed number of decoy wins (Algorithm S1 with $c = \frac{1}{2}$).

Notably, in this enhanced version of RESET, each target competes in step 1 with two associated decoys rather than a single one. The sets of of training and estimating decoys are roughly equal-sized for both RESET procedures, and therefore both sets are typically larger in the case of the two-decoy version. While these larger training and estimating sets come at a cost of a reduced number of target wins, keep in mind that these lost targets only win one of their two decoy competitions; hence, they are unlikely to be high scoring enough to eventually make it to the reported discoveries list. At the same time, those larger sets of training and estimating decoys allow us to better train our model, as well as to have a more stable final procedure when defining the discovery list using SeqStep+. This is borne out in practice when we integrate Percolator with the two-decoy RESET in Section 3.3.2.

2.4 Both versions of RESET control the FDR

To establish that the RESET procedures provide finite-sample FDR control we need to strengthen Assumption 1 that forms the basis of TDC [1, 13] to refer to the side information $\mathbf{x}$ as well:

Assumption 2. *Conditional on all the winning scores W, on all the features $\mathbf{x}$, and on the labels L of the false null hypotheses, the labels of the true null hypotheses are independent identically distributed ± 1 random variables with*

- $P(L_i = 1) = P(L_i = -1) = 1/2$ *in the single-decoy case, and*
- $P(L_i = 1) = 1 - P(L_i = -1) = 1/3$ *in the two-decoy case.*

Note that because the enhanced RESET considers two decoys for each target instead of one, the last assumption is adjusted to reflect the fact that for an incorrect target and its two derived decoys, each are equally likely to win their triplet competition. The following theorem, whose proof is given in Supplementary Section S.9, underpins the theoretical basis of FDR control.

Theorem 1. *Under Assumption 2 both versions of RESET control the FDR in the feature-augmented TDC setting at the user-specified threshold of α.*

3 Percolator-RESET

3.1 Peptide detection using tandem mass spectrometry

The peptide detection process is typically initiated by scanning each input spectrum against the peptide database for its best matching peptide. Pioneered by SEQUEST [7], each such match is evaluated by a score function that measures the similarity between the observed spectrum and a theoretical spectrum constructed for the candidate peptide. The list of candidate peptides to be scored consists of all the database peptides whose mass lies within a specified tolerance to the experimental mass that was measured when the spectrum was generated. The optimal peptide-spectrum match (PSM) for the given spectrum is then recorded along with its score (ties are broken randomly) [22]. Note that when using TDC the peptide database is the concatenation of the target and decoy databases, and that, critically, the search engine cannot distinguish between the two parts.

The PSMs are then aggregated at the peptide level: the score assigned to each (target or decoy) peptide is the the maximum score of all PSMs associated with it (assuming for simplicity that a PSM score is > 0 so we assign a score of 0 if no PSM is associated with the peptide). From each pair of target-decoy peptides we can next choose the winning score and control the FDR as described above (PSM-and-peptide, [19]). Note that the same peptide-level aggregated analysis can be applied to more sophisticated score functions that take into account additional features of the PSM, as described in the next section.

3.2 Percolator

The general idea of applying a machine learning classifier to re-rank peptide-spectrum matches (PSMs) from a database search was pioneered by the PeptideProphet algorithm [15], which involved training a linear discriminant analysis (LDA) model to classify PSMs on the basis of a hand-curated set of "correct" and "incorrect" PSMs. This approach suffered from poor generalizability when the data used to train the model had different characteristics than the data being classified. This problem led to the development of Percolator [14], which is a semi-supervised machine learning algorithm that trains its model using a collection of target and decoy PSMs. The idea is that the decoy PSMs can be labeled "incorrect," whereas the targets are a mixture of "correct" and "incorrect" PSMs. Importantly, because target and decoy PSMs can be generated automatically, no hand curation of labels is necessary, and Percolator can be retrained for every new dataset. This approach guarantees that the training set and test set distributions are similar. The semi-supervised approach has become the *de facto* standard in the field.

Percolator works by utilizing an abundance of features, in addition to the primary PSM score, to help distinguish between correct and incorrect PSMs. These features may include information about the input spectrum, such as the precursor charge state, and the matched peptide, such as the peptide length. Using this information Percolator defines a negative set from the decoy (winning) PSMs and an initial positive set of high-confident target PSMs. It then iteratively trains a linear support vector machine (SVM) [23] to distinguish the fixed negative set and the current positive set. Each newly fitted SVM is then used to rerank the PSMs, and hence update the positive set of high-confident target PSMs. This iterative procedure whose schematic is given in Figure S1 is repeated a pre-fixed number of times.

As mentioned, we have recently shown that in practice Percolator can fail to control the FDR due to its dual usage of the decoys: modeling incorrect PSMs during training, and estimating false discoveries when controlling the FDR [9]. While Percolator uses a cross-validation procedure to attempt to prevent information about the target/decoy labels to leak from the training set to the estimation set [12], this strategy can fail due to the same multiplicity problem that makes PSM-level FDR control challenging [13]. Specifically, multiple spectra can be generated from the same peptide species and hence the same PSMs might appear in the different cross-validation folds that Percolator uses, violating the separation principle on which cross-validation is based [9].

3.3 Percolator-RESET: combining RESET with Percolator

3.3.1 Overall approach: Percolator-RESET

RESET allows us to fix Percolator's lapses in FDR control by integrating the latter's iterative linear SVM training into the RESET wrapper framework. As a specialization of RESET, Percolator-RESET (Algorithm S3) can be described by applying the following adjustments to the steps described in Section 2.2:

- In Step 1 of Section 2.2, we integrate the PSM-and-peptide protocol of [19] (Section 2.1 and Algorithm S4a) with the dynamic-level competition we introduced in [8]. Specifically, we start with the same initial PSM-level competition that PSM-and-peptide employs, recording for each spectrum its best matching peptide in the concatenated target-decoy database while using the primary database search score. For example, in our experiments using the Tide search engine [4], we employ the Tailor similarity score [25].

  Assuming no variable modifications are considered, the subsequent competition occurs, as in the original PSM-and-peptide, at the peptide-level: we define each peptide's score as the maximal PSM score associated with that peptide, and we let the peptide inherit all the features associated with this maximal scoring PSM (e.g., charge, peptide length, delta mass — see Table S3 for the full list). Then for each pair of target-decoy peptides, we take the peptide with the higher peptide score, along with the associated feature information. The selected peptide may be a decoy ("decoy win") or a target ("target win").

  One challenge is that, when variable modifications are considered, the target database will contain clusters of target peptides with highly similar theoretical spectra. In this case, none of the canonically constructed decoy databases can offer a proper competition to a target peptide that was identified due to an incorrect match within its cluster. To account for this, the level of competition dynamically switches from the peptide level to what we call the **stem** level (where the stem is the peptide sequence with all modifications removed) [8]. At this level, we consider the maximal scoring peptide (a representative peptide) from each cluster in the target database, and we compete this peptide with the maximal scoring peptide from the associated decoy pairs. We provide further details in Section S.1 and Algorithms S4a-S4b.

- For Step 3 of Section 2.2, the negative set used to train an SVM is the set of training decoys. We extract a positive set from the pseudo targets analogously to Percolator by applying SeqStep+ (Algorithm S1), except we use the pseudo target peptides and the training decoy peptides instead of the target and decoy PSMs as Percolator does. More specifically, all the winning peptides are first ordered by decreasing Tailor scores. Using SeqStep+ we then look for the largest set for which the estimated proportion of false discoveries among the top pseudo target peptides is still $\leq \alpha_1$ (we use the same $\alpha_1 = 0.01$ as in Percolator). Because we only sampled $s = 1/2$ of the decoy wins to define the training decoys, we need to adjust for that by multiplying the observed number of training decoys (plus 1) by 3. The corresponding set of "discovered" pseudo targets is our positive set for training the SVM. In Section S.2 we explain how we deal with cases where the above defined positive set is empty. Using the positive and negative training sets, and their associated features (including the scores), we train an SVM to rescore each peptide (further details are given in Section S.2).

  We then iterate the procedure described in the above paragraph but using the SVM score instead of the Tailor score to re-rank the peptides. The SVM is trained this way for a fixed number of iterations (we used `total_iter` = 5). A schematic of Percolator-RESET is given in Figure S2.

3.3.2 Percolator-RESET with two decoys

If we search the spectra against two randomly generated peptide-shuffled decoy databases, then we can employ the two-decoy version of Percolator-RESET, which follows the steps of the enhanced two-decoy RESET procedure with the following adjustments:

- In step 1 we generalize the double competition protocol of Section 3.3.1 to account for an extra decoy database as follows. The PSM-level competition now takes place by assigning to each spectrum its best

matching peptide across the two searches, which is akin to searching a database made of the target concatenated to both decoy databases. Each peptide is then assigned its maximal Tailor-scoring PSM, and it inherits the maximal PSM's features (Section S.3 fills in some technical details that we glossed over here).

As with the single-decoy case, we next employ dynamic-level competition: if no variable modifications are used, then the second competition takes place between all sets of matched peptide triplets: a target and its two shuffled decoy derivatives. The peptide with the maximal score wins the triplet competition, and the triplet is represented by its winning peptide with its inherited features while the two losing peptides are discarded. If variable modifications are used, then a stem-level competition takes place between all sets of matched cluster triplets: the cluster of target peptides and the two clusters of their corresponding decoys, one from each decoy database. The representative peptide with the maximal score from each cluster wins the triplet competition, and the triplet is represented by its winning representative peptide together with its inherited features while the two losing representative peptides are discarded (see Algorithms S5a-S5b.)

- The SVM training phase (step 3) is mostly the same as in the single-decoy Percolator-RESET. The positive set is similarly defined by applying SeqStep+ to the pseudo targets and training decoys using the same $\alpha_1 = 0.01$. However, we now multiply the observed number of training decoy (plus 1) by 2 rather than 3 as in the analogous step in the single-decoy Percolator-RESET.

Because the RESET procedures are general wrappers for any machine learning model, Theorem 1 trivially holds true for the two Percolator-RESET procedures under the same assumptions. In Section 4.1 below we bolster this theoretical observation by empirical experimental evidence suggesting the procedures indeed control the FDR.

3.3.3 Supplementary Information and Availability

The above description of Percolator-RESET glossed over a few complex technical details whose inclusion here would have mostly muddied the water, so accordingly they are covered in Supplementary Sections S.1-S.3 and S.6-S.8.

Finally, an open source, Apache licensed Python implementations of both Percolator-RESET procedures are available at `https://github.com/freejstone/Percolator-RESET`. Both tools will be implemented as part of an upcoming release of Percolator (v4.0).

4 Experimental Results

4.1 Both versions of RESET control the FDR

There are two complementary ways in which a given procedure can be said to control the FDR: theoretically and empirically. A rigorous proof of theoretical FDR control is highly desirable, but even with such a proof, empirical evidence of FDR control is also useful, as a way to check that the assumptions underlying the proof hold in practice. Accordingly, here we complement Theorem 1 by empirically demonstrating FDR control using real data.

To provide an empirical assessment of FDR control, we carried out a series of "entrapment" experiments [10]. In this setup, spectra generated in a controlled experiment are searched against a combined database containing those in-sample peptides plus an irrelevant database of entrapment sequences, typically taken from a species known not to be present in the sample. For our experiments, we used spectra from an 18-protein mixture (ISB18) [16]. Furthermore, as first introduced in [9], to increase the amount of foreign spectra and thus vary the signal-to-noise ratio, the in-sample database is randomly downsampled to use increasingly smaller proportions (100%, 75%, 50%, 25%). The resulting four target databases have a ratio of entrapment to ISB18 peptides ranging from 636:1 to 2539:1. Because the number of entrapment peptides is so much larger than the number of ISB18 peptides, incorrect PSMs will usually involve an entrapment peptide, thereby allowing us to reliably estimate the false discovery proportion (FDP) among the set of discovered peptides. Furthermore, by averaging the estimated FDP over 100 randomly generated decoy

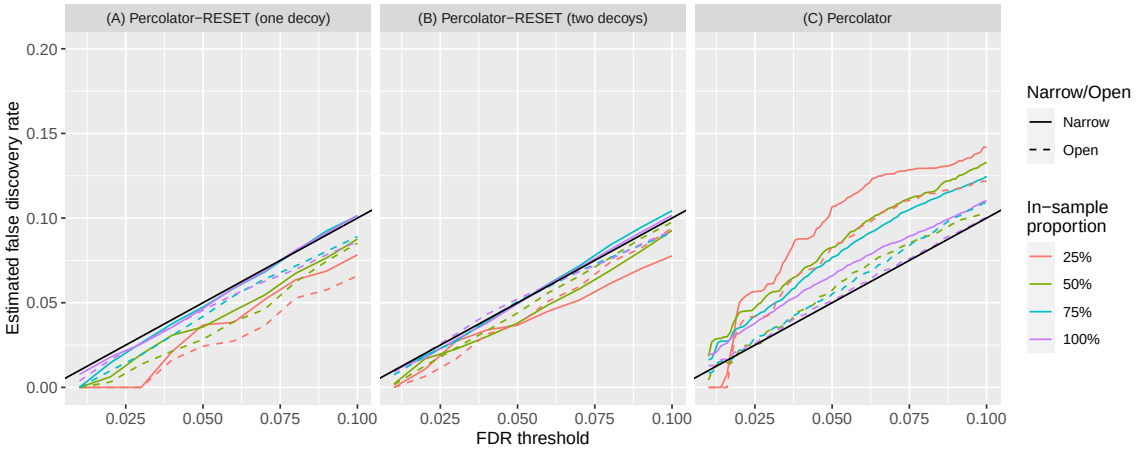

Figure 1: **Empirical FDRs of Percolator Algorithms.** The empirical FDR as estimated from our entrapment experiments on the ISB18 dataset using (A) single-decoy Percolator-RESET, (B) two-decoy Percolator-RESET, (C) Percolator. The FDP is estimated at a range of FDR thresholds ($[0.01, 0.1]$), and its average over 100 randomly generated decoys (and over 2 randomly generated training decoy sets for (A) and (B)) is the empirical FDR. Each curve represents the empirical FDR using a target database constructed with the specified proportion of the in-sample database, in either narrow- or open-search mode.

databases, along with 2 random draws of the training decoy set for each decoy database, we obtain an estimated FDR for a given FDR threshold. If the FDR control procedure is valid, then this empirical FDR should not significantly exceed the threshold. We conducted the entrapment experiments using both narrow and open searches, where the distinction between the two searches is in the specified tolerance of the difference between the database candidate peptide and the spectrum's experimentally measured mass (see Supplementary Section S.4 for further details).

The results of our entrapment experiments (Figures 1A-B) confirm that the empirical FDRs of the Percolator-RESET procedures are typically below their corresponding thresholds. We observe a few exceptions to this trend, the worst (as a percentage) being for the two-decoy Percolator-RESET at an FDR threshold of 0.04, with an empirical FDR of 0.0431 (using 100% of the in-sample peptides and using an open-search mode). Given that we averaged the result over 100 decoys, this 8% violation does not seem that significant. Moreover, keep in mind that even if we were able to average the FDP over a much larger number of decoys, we still do not average out all the variability in the experiment: the MS/MS runs themselves introduce a random effect that cannot be eliminated just by averaging over multiple drawn decoys. Indeed, accounting for that latter effect would have required repeating these MS/MS experiments many times.

We also analyzed Percolator's behavior in the context of this entrapment setup. The results (Figure 1C), done here with 100 decoy databases, reinforce our previously reported observation (based on only 20 decoy databases), that Percolator seems to fail in controlling the FDR in these entrapment experiments [9]: the worst empirical FDRs at thresholds 0.01, 0.05 and 0.1 were 0.0195, 0.1068 and 0.1419, respectively. As noted previously, we believe that Percolator's apparent failure here stems from the presence of multiple spectra generated from the same peptide species. Consequently, these essentially identical spectra may appear in both the training and test sets in Percolator's cross-validation scheme causing the model to overfit and violate FDR control.

4.2 The Percolator-RESET procedures maintain a similar number of discoveries to Percolator

Percolator can fail to control the FDR, and therefore its results should be taken with a grain of salt. However, particularly in light of Percolator's popularity, it is instructive to compare its apparent statistical power (i.e., the number of peptides discovered at a specified FDR threshold) with that of our two new procedures. To do so, we used a collection of publicly available MS/MS datasets from 20 projects in the Proteomics Identifications Database (PRIDE) [20]. The datasets were essentially selected at random, with the constraint that Param-Medic [21] detected post-translational modifications in 10 of the 20 files. For each spectrum file, we used Tide-index to create 10 randomly shuffled decoy databases. Then for each spectrum

file we used Tide-search to search against each of the target-decoy databases in both narrow and open search mode, resulting in a total of $20 \times 10 \times 2$ searches (see Supplementary Section S.5 for further details).

The results in Figure 2(A-B) demonstrate that both versions of Percolator-RESET provide a comparable number of detected peptides as Percolator on the PRIDE-20 datasets. Indeed, at a 1% FDR threshold the single-decoy Percolator-RESET marginally loses to Percolator: considering the ratio of the number of reported discoveries averaged over the 10 decoy sets, we find its median across the PRIDE-20 dataset (expressed as percentages) is 99% in both narrow and open search modes. At higher FDR thresholds, this difference in number of discoveries shrinks even more with the median average ratio now at 99.8% and 99.9% in narrow mode and open mode, respectively, at the 5% FDR threshold. Moreover, the two-decoy Percolator-RESET appears to marginally edge out Percolator, with median average ratios of 100.2% and 100.4% at a 1% FDR threshold and a median greater than 100% for all remaining FDR thresholds. The slight improvement is perhaps not surprising, given that the algorithm utilizes an additional decoy database. Overall, these results suggest that our new procedures deliver better FDR control without a loss in statistical power.

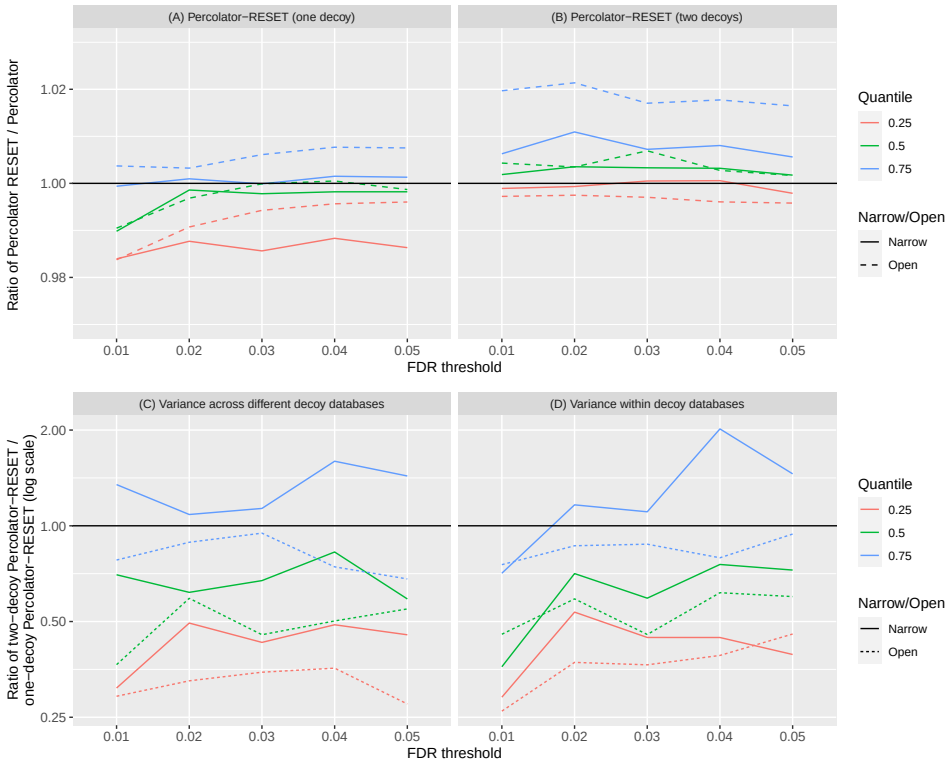

Figure 2: **Performance of Percolator-RESET.** Figures(A-B) display the quartiles, with respect to the 20 PRIDE-20 datasets, of the mean ratio of the single-decoy (A) and two-decoy (B) Percolator-RESET discovered peptides to Percolator discovered peptides. For (A), the mean ratio is taken with respect to each dataset's 10 decoys. For (B), because this version of Percolator-RESET considers two decoy databases, we calculated the number of discoveries 10 times, each using a unique pair from the 10 available decoy databases, while ensuring that each decoy database was used exactly twice. Subsequently, the discovery ratio in that case is computed as the number of discoveries of the two-decoy Percolator-RESET to the average number of discoveries of Percolator using the two decoy databases separately. Figures (C-D) display the quartiles, taken over the 20 PRIDE-20 datasets, of the ratios of variances in the number of discoveries from two-decoy Percolator-RESET (numerator) to one-decoy Percolator-RESET (denominator). Each PRIDE-20 dataset had 10 decoy databases constructed. In (C), the variance is calculated over different decoy databases. In (D), the variance is calculated over the random split between the training and estimating decoys, keeping the decoy databases fixed.

4.3 Percolator-RESET's discoveries are less variable using two decoy databases instead of one

As suggested in Section 3.3, we hypothesize that the use of an extra decoy database should allow Percolator-RESET to reduce some of the variability we observe in the number of discoveries reported by the single-

decoy version. This variability is inherent to the process of randomly splitting the decoy set into training and estimating decoys and, as explained, the two-decoy version typically increases the size of both sets. It follows that the two-decoy Percolator-RESET typically utilizes a larger number of training decoys as the negative set for SVM training, as well as a larger number of estimating decoys for the FDR estimation, thus reducing the overall variability in the reported list of peptides.

To test this hypothesis, we conducted two experiments. We began the first by finding, for each PRIDE-20 spectrum file, the sample variance in the number of discoveries reported in 10 applications of the single-decoy Percolator-RESET, each using a different decoy database. We then compared each of those 20 sample variances with corresponding sample variances of 10 applications of the two-decoy Percolator-RESET, each using a distinct pair of decoy databases (each of the 10 available decoy databases was used in exactly two applications). Specifically, for each PRIDE-20 spectrum file we calculated the ratio of the two-decoy Percolator-RESET to the single-decoy Percolator-RESET sample variances. Figure 2C shows the quartiles (over the 20 datasets) of these ratios in logarithmic scale, using both narrow and open searches. At the 1% FDR threshold the two-decoy version typically exhibits significantly less variability than the one-decoy version, yielding a median reduction of 29.8% in the narrow-search and 63.4% in the open-search.

In the second experiment, we focused on the variation in the number of discoveries due to the random splitting of the decoys into training and estimating decoys. For each PRIDE-20 spectrum file we first pre-selected two decoy databases arbitrarily. Next, we applied the two-decoy Percolator-RESET 20 times, each using a different random split between the training and estimating decoys, and we calculated the sample variance in the number of discoveries over these 20 runs. For the single-decoy version, we applied it 10 times using the first decoy database and 10 times using the second decoy database, randomly splitting the decoys in all those applications. Subsequently, we calculated the pooled sample variance in the number of discoveries reported by the single decoy Percolator-RESET, i.e., the average of the two sample variances calculated with respect to the 10 runs using each of the two databases. Finally, we calculated the ratio of the two-decoy version's sample variance to the single-decoy version's pooled sample variance for each PRIDE-20 spectrum file. Figure 2D plots the quartiles of these ratios in logarithmic scale while varying the FDR threshold. Again, we see that the two-decoy version typically exhibits substantially lower variability than the single-decoy Percolator-RESET, with a median reduction of 64.0% and 54.4% at the 1% FDR threshold when using a narrow and open search respectively.

4.4 Adaptive Knockoffs does not offer a practical general alternative

As mentioned in the introduction, designed specifically for FDR control in the setup of competition with side-information, Adaptive Knockoffs [24] can in principle offer an alternative to Percolator-RESET. When applying Adaptive Knockoffs, the user needs to select one of several available "filters", which iteratively determines the sequence at which the next label L_i should be exposed based on maximizing the likelihood that it is either a true null or a decoy/knockoff win (see Section S.11 for more details).

Empirically, the *EM filter*, which uses a two-group parametric model, fitted by the EM algorithm, appears to be the most powerful in their paper. These parametric assumptions do not appear applicable in our setting, but regardless we found this selection, as well as the second best performing *Random Forest filter* impractical for our context in general. Specifically, we chose the largest of our PRIDE-20 datasets (project ID PXD019354) and compared the runtimes of Percolator-RESET and Adaptive Knockoffs with the EM and Random Forest filters. While Percolator-RESET completed its task in 48 seconds we had to resort to estimating the time it would take Adaptive Knockoffs' versions: 4.4 days for the EM and 107 days for the Random Forest filter (see Section S.11 for details).

We then used one of our smaller PRIDE-20 datasets (project ID PXD022257) to first gauge our method of estimating the runtime of Adaptive Knockoffs. Reassuringly, we found that at 2.43 hours it was underestimating the actual time it took the EM filter: 3.63 hours (Percolator-RESET finished in 14 seconds). Interestingly, not only was Adaptive Knockoffs significantly slower, it also reported substantially less discoveries than Percolator-RESET: 2264 vs. 3024 at 1% FDR with similar results for other FDR thresholds (Section S.11). The relatively poor power of Adaptive Knockoffs is probably related to the EM filter's use of a specific parametric model whose applicability to our setting seems questionable.

Taken together we can conclude that Adaptive Knockoffs is not a viable alternative to Percolator-RESET in the peptide detection problem.

5 Discussion

Competition-based FDR control, as well as FDR control with side-information have drawn increased interest in recent years. However, none of the published solutions that guarantee finite sample control is practically applicable to the peptide detection problem. At the same time, while Percolator is a popular tool that can effectively address this problem, our previous analysis in [9], as well as our analysis in Section 4.1, showed that in practice it can apparently fail to control the FDR. RESET was designed to bridge this gap between the available procedures that guarantee finite-sample FDR control and Percolator's utility.

The RESET procedures achieve this by introducing a decoy-splitting procedure, which may be used in conjunction with any semi-supervised machine learning models that rely on a negative and an unlabelled set. We prove that, under mild assumptions analogous to those that TDC relies on, both versions of RESET control the FDR in the feature-augmented TDC setup. We supplement this theoretical result by showing that our two RESET procedures, integrated with a Percolator-like model, appear to effectively control the FDR in the same entrapment experiments that Percolator failed to do so. Importantly, in spite of its tighter FDR control, the power of the single-decoy version of Percolator-RESET is only marginally lower than Percolator's. Moreover, utilizing a second decoy dataset, the enhanced version of Percolator-RESET is marginally more powerful than Percolator while exhibiting significantly less variability than the single-decoy version in the number of discoveries it reports.

By splitting the winning decoys into training and estimating ones, our new procedures do away with Percolator's cross-validation scheme for FDR control, while still using cross-validation for the selection of hyperparameters. Notably, it is this former cross-validation that seems to be at the heart of Percolator's potential struggles with FDR control [9]. The cross-validation was deemed necessary at the time, because Percolator uses the *entire* set of decoy wins as its negative set [12]. Removing the need for cross-validation is also computationally efficient, particularly if our RESET procedures are to be extended to other machine learning models in the future. It would be interesting to compare the impact of different such models on the performance of RESET.

Another notable difference between Percolator and both versions of Percolator-RESET is that the former trains its SVM at the PSM level, whereas the latter two train at the peptide level. In fact, it is impossible to train the single-decoy version of Percolator-RESET at the PSM level without violating the assumptions that guarantee its FDR control. Indeed, the same multiplicity problem that makes it impossible to use TDC for PSM-level FDR control applies here: multiple spectra generated by the same peptide species could end up creating nearly identical PSMs that will be shared between the training and estimating decoys. In practice, this means that some sort of over-fitting will occur in this case. In contrast, we can adjust the two-decoy Percolator-RESET so that it can train its SVM at the PSM level while still rigorously controlling the FDR. We plan on looking further into the difference between peptide- and PSM-level training in this case.

Finally, we stress that while the motivation for RESET was the peptide detection problem it is applicable more generally to any setup where the competition-based approach with side-information applies. Indeed, we are working on testing its performance in the regression setting.

Acknowledgments We are grateful to the anonymous referees for their comments and suggestions which helped significantly improve this manuscript.

References

[1] R. F. Barber and Emmanuel J. Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.

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

[3] Y. Danilova, A. Voronkova, P. Sulimov, and A. Kertész-Farkas. Bias in false discovery rate estimation in mass-spectrometry-based peptide identification. *Journal of Proteome Research*, 18(5):2354–2358, 2019.

[4] B. Diament and W. S. Noble. Faster SEQUEST searching for peptide identification from tandem mass spectra. *Journal of Proteome Research*, 10(9):3871–3879, 2011.

[5] J. E. Elias and S. P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, 2007.

[6] K. Emery, S. Hasam, W. S. Noble, and U. Keich. Multiple competition-based fdr control and its application to peptide detection. In *International Conference on Research in Computational Molecular Biology*, pages 54–71. Springer, 2020.

[7] J. K. Eng, A. L. McCormack, and J. R. Yates, III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5:976–989, 1994.

[8] J. Freestone, W. S. Noble, and U. Keich. Analysis of tandem mass spectrometry data with CONGA: Combining open and narrow searches with group-wise analysis. *bioRxiv*, 2023. https://www.biorxiv.org/content/10.1101/2023.05.02.539167v1.

[9] J. Freestone, W. S. Noble, and U. Keich. Re-investigating the correctness of decoy-based false discovery rate control in proteomics tandem mass spectrometry. *bioRxiv*, 2023. https://doi.org/10.1101/2023.06.21.546013.

[10] V. Granholm, J. F. Navarro, W. S. Noble, and L. Käll. Determining the calibration of confidence estimation procedures for unique peptides in shotgun proteomics. *Journal of Proteomics*, 80(27):123–131, 2013.

[11] V. Granholm, W. S. Noble, and L. Käll. On using samples of known protein content to assess the statistical calibration of scores assigned to peptide-spectrum matches in shotgun proteomics. *Journal of Proteome Research*, 10(5):2671–2678, 2011.

[12] V. Granholm, W. S. Noble, and L. Käll. A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC Bioinformatics*, 13(Suppl 16):S3, 2012.

[13] K. He, Y. Fu, W.-F. Zeng, L. Luo, H. Chi, C. Liu, L.-Y. Qing, R.-X. Sun, and S.-M. He. A theoretical foundation of the target-decoy search strategy for false discovery rate control in proteomics. *arXiv*, 2015. https://arxiv.org/abs/1501.00537.

[14] L. Käll, J. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss. A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4:923–25, 2007.

[15] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identification made by MS/MS and database search. *Analytical Chemistry*, 74:5383–5392, 2002.

[16] J. Klimek, J. S. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. R. Gafken, J. E. Katz, P. Mallick, H. Lee, A. Schmidt, R. Ossola, J. K. Eng, R. Aebersold, and D. B. Martin. The standard protein mix database: a diverse data set to assist in the production of improved peptide and protein identification software tools. *Journal of Proteome Research*, 7(1):96–1003, 2008.

[17] L. Lei and W. Fithian. Adapt: an interactive procedure for multiple testing with side information. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):649–679, 2018.

[18] D. Leung and W. Sun. Zap: z-value adaptive procedures for false discovery rate control with side information. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(5):1886–1946, 2022.

[19] A. Lin, T. Short, W. S. Noble, and U. Keich. Improving peptide-level mass spectrometry analysis via double competition. *Journal of Proteome Research*, 21(10):2412–2420, 2022.

[20] L. Martens, H. Hermjakob, P. Jones, M. Adamsk, C. Taylor, D. States, K. Gevaert, J. Vandekerckhove, and R. Apweiler. PRIDE: The proteomics identifications database. *Proteomics*, 5(13):3537–3545, 2005.

[21] D. H. May, K. Tamura, and W. S. Noble. Detecting modifications in proteomics experiments with Param-Medic. *Journal of Proteome Research*, 18(4):1902–1906, 2019.

[22] A. I. Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73(11):2092 – 2123, 2010.

[23] W. S. Noble. What is a support vector machine? *Nature Biotechnology*, 24(12):1565–1567, 2006.

[24] Z. Ren and E. Candès. Knockoffs with side information. *The Annals of Applied Statistics*, 17(2):1152–1174, 2023.

[25] P. Sulimov and A. Kertész-Farkas. Tailor: A nonparametric and rapid score calibration method for database search-based peptide identification in shotgun proteomics. *Journal of Proteome Research*, 19(4):1481–1490, 2020.