How to train a post-processor for tandem mass spectrometry proteomics database search while maintaining control of the false discovery rate

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

Abstract

Decoy-based methods are a popular choice for the statistical validation of peptide detections in tandem mass spectrometry proteomics data. Such methods can achieve a substantial boost in statistical power when coupled with post-processors such as Percolator that use auxiliary features to learn a better-discriminating scoring function. However, we recently showed that Percolator can struggle to control the false discovery rate (FDR) when reporting the list of discovered peptides. To address this problem, we developed a general protocol called "RESET" that is used to determine the list of reported discoveries in a target-decoy competition setup, where each putative discovery is augmented with a list of relevant features. The key idea is that, rather than using the entire set of decoys wins for both training and estimating, a random subset of decoys is chosen for each task. One subset is then used to train a semi-supervised machine learning model, and the other is used for controlling the FDR. We show that, by applying RESET to the Percolator algorithm, we control the FDR both theoretically and empirically, while reporting only a marginally smaller number of discoveries than Percolator. Furthermore, a variant of the protocol that we introduce, which requires generating two decoys per target, maintains theoretical and empirical FDR control, typically reports slightly more discoveries than Percolator, and exhibits less variability than the single-decoy approach.

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

1 Introduction

Tandem mass spectrometry provides a high-throughput method for detecting proteins in a complex sample. The approach involves digesting the proteins to peptides and then running a liquid chromatography tandem mass spectrometry (MS/MS) experiment to generate MS2 fragmentation spectra. Each of those spectra is then searched against a relevant peptide database to find its best matching peptide. Deciding which peptides to report is most commonly done through target-decoy competition (TDC), which involves searching the same spectra against a database of decoy peptides, generated by reversing or randomly shuffling the original ("target") peptides. For each spectrum, the best-scoring target and decoy peptides compete against each other. Thereafter, a relatively simple calculation provides an estimate of the false discovery rate (FDR) among all target peptides that win the competition and whose assigned score exceeds a given threshold [4, 17]. Notably, it has been established that, under natural assumption that are specified below, this TDC procedure theoretically guarantees FDR control [1, 12].

Unfortunately, applying TDC to database search results can become more complicated when those results are first post-processed by a machine learning algorithm. 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 [14], 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 [13], 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.

Despite its popularity, we have recently shown that in practice Percolator can fail to control the FDR [7]. The problem arises because Percolator uses decoys in two fundamentally different ways: first to train a machine learning model to distinguish between correct and incorrect PSMs, and second to estimate FDR using the TDC procedure. Recognizing this challenge, Percolator uses a cross-validation procedure to attempt to prevent information about the target/decoy labels to leak from the training set to the test set [11]. However, we argued that in practice Percolator's training process can indirectly peek at the target/decoy labels of the test set, and hence the overall procedure may fail to control the FDR.

In this paper we offer 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, Pecolator-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. Finally, we note that although this paper is concerned with the specific goal of detecting peptides using fragmentation spectra, the two RESET procedures are more generally applicable.

2 Methods

2.1 Target-decoy competition

Our implementation of the RESET procedure employs a modification of the standard TDC procedure, so we begin with outlining the four steps of PSM-and-peptide, which is the particular protocol of peptide-level TDC that we build on [17]. First, we compete each target PSM with its corresponding decoy PSM, and we keep only the higher scoring of the two (ties are randomly broken throughout). Second, each peptide (target or decoy) is assigned a score, which is that of the maximal scoring PSM associated with the peptide. A key feature of PSM-and-peptide is that it next extends the target-decoy competition to the peptide level: in the third step, each target peptide competes with its paired decoy, and only the higher scoring of the two is kept, together with a label indicating whether the winning score corresponds to the target peptide ("target win") or the decoy peptide ("decoy win").

Finally, PSM-and-peptide proceeds, as all variants of TDC do, to use the decoy wins to estimate the false discovery rate (FDR) among the corresponding target wins, which it subsequently uses to determine which winning target peptides to report. More specifically, TDC begins this step by sorting the winning peptide scores in decreasing order. For each k, it then uses the number of decoy wins in the top k scores as an estimate of the number of falsely detected/discovered target peptides among the top k winning peptides. This approach is justified by TDC's first key assumption, namely, that each target peptide that is absent from the sample is equally likely to win the peptide-level competition as its decoy. Dividing this estimated number of falsely discovered target peptides by the complementary number of target wins in the top k winning peptides gives an estimate of the false discovery rate among those top target wins. To rigorously control the FDR at level α, TDC adds 1 to the number of decoy wins before dividing by the number of target wins, and it looks for the largest k for which this estimate is still $\leq \alpha$. TDC then uses this largest k to report all the target peptides among the top k peptides. This estimating/controlling step of TDC is further described in SeqStep+ (Algorithm 1 with $c = 1/2$ here), which in turn is essentially the Selective SeqStep+ procedure of [1]. It was shown that adding to the first key assumption a second one, which requires that the first assumption holds for each target-decoy paired-peptides independently of all other pairs, as well as of the pair's winning score, TDC rigorously controls the FDR [1, 12].

Importantly, the same approach will control the FDR for *any* ordering of the winning peptides 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.

2.2 Feature-augmented target-decoy competition

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. As in TDC, we assume that each putative discovery is paired with a competing, artificially constructed, or statistically speaking, 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 is that of the maximal PSM associated with it). As an aside, note that in the context of variable selection, where this setup is also applicable, the decoys are called *knockoffs* [1].

Departing now from the canonical TDC setup, we introduce a "feature-augmented" setting, in which 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 each target or decoy peptide that has at least one PSM associated with it can inherit the features of its maximally scoring PSM, such as the ones that Percolator uses: the precursor charge state, peptide length, etc. Peptides with no associated PSMs either lose their paired competition or are dropped if neither of the paired peptides is associated with a PSM.

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 [2, 10]. In the feature-augmented setting, we also assume that the following extended analogue of TDC's key assumptions hold. First, each false target discovery is equally likely to be the higher scoring when competing with its paired decoy. Second, this happens independently of all the winning scores, all features, and the labels of all other discoveries.

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.

2.3 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 Algorithm 2.

1. RESET begins with a TDC-like competition step: from each pair of target and decoy discoveries only the maximal one is kept, together with a label indicating whether this was a target or a decoy win. RESET also records the vector of features 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 1 with $\frac{c}{1-c} = 2$, or $c = 2/3$).

2.4 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 indicating whether the target or one of its two associated decoys is equal to the maximum of the three scores, 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 of the winning discovery. This multiple competition method is essentially the same as the *max method* introduced by Emery et al. [5].
- 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 1 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 below.

2.5 Combining RESET with Percolator

2.5.1 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 3) can be described by applying the following adjustments to the steps described in Section 2.3:

- In Step 1 of Section 2.3, we integrate the PSM-and-peptide protocol of [17] (Section 2.1 and Algorithm 4a) with the dynamic-level competition we introduced in [6]. 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 [3], we employ the Tailor similarity score [20].

  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 (charge, peptide length, delta mass, etc.). 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) [6]. 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 2.5.3 and Algorithms 4a-4c.

- For Step 3 of Section 2.3, 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 1), 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 2.5.4 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 2.5.4).

  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).

2.5.2 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 2.5.1 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 2.5.5 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 5a-5c.)

- 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.

2.5.3 Dynamic-level competition

We recently pointed out that if variable modifications are specified, the decoy database can no longer provide adequate competition against some of the incorrectly identified target peptides [6]. Specifically, TDC relies on the assumption that a false discovery (a peptide that won its competition but is not in the sample) is equally likely to be a target or a decoy win. However, when variable modifications are specified the database will often contain clusters of highly similar peptides corresponding to variable modifications of the same unmodified ("stem") form of a single peptide. The problem is that the canonically constructed decoys (by shuffling or reversing the target peptides) cannot account for mismatches within such clusters. For example, consider a spectrum that was generated by P[10]EPTIDE but because of random noise is optimally matched to PEP[10]TIDE. In this case, the decoys cannot offer a real competition to such a mismatch because, typically, the "closest" decoy to P[10]EPTIDE would still be significantly different from its cluster-neighbor PEP[10]TIDE. This problem can be further exacerbated in a so-called "open" search, in which each peptide is allowed to match to spectra whose associated precursor masses can substanially differ from the mass of the peptide. In an open search setting, in our example P[10]EP[10]TIDE would also belong to the cluster.

To address this problem we employ an adaptation of the dynamic-level competition introduced in [6]: when no variable modifications are used, the second competition takes place at the peptide level. On the other hand, when variable modifications are used and the tandem spectra are searched with a narrow precursor tolerance, then the stem level analysis first clusters peptides with the same residue sequence and modification mass. In our above example, this would amount to clustering the peptides P[10]EPTIDE, PEP[10]TIDE, and any other peptidoform of PEPTIDE with a mass modification that totals 10 Da. We then select from each cluster its highest scoring peptide as

the cluster's representative. Finally, the target-decoy pairing can be readily extended to the clusters, and hence to the clusters' representatives. Therefore, we can execute our second target-decoy competition at this representative level, keeping the higher scoring of the two competing representatives together with its associated features. Note that the competing representatives are paired at the cluster level, and each of them might be paired with other peptides from the cluster of the competing representative. The above procedure also applies when variable modifications are used in combination with an open search, except that the clusters are defined as all peptidoforms of the same stem form (clustering all peptidoforms of PEPTIDE in our example above).

An algorithmic description of this procedure is given in Algorithms 4a-4c, and the extension of the procedure when two decoy databases are considers is outlined in Section 2.5.2 and detailed in Algorithms 5a-5c.

2.5.4 SVM training in Percolator-RESET

If the dataset is sufficiently small, it is possible that the positive training set, determined in Step 3 of Sections 2.5.1 and 2.5.2, will be empty. In this case, we first remove the +1 penalty from SeqStep+ (SeqStep in Algorithm 1) and repeat the procedure without it. If the newly defined positive training set is still empty, then we iteratively increase the training FDR threshold from its initial value of $\alpha_1 = 0.01$ by increments of 0.005 until a non-empty positive training set is obtained.

Because the positive and negative training sets are expected to be unbalanced, we select the class weights $C^+, C^- \in \{0.1, 1, 10\}$ using a three-fold cross validation procedure, as in Percolator. Here we use the average number of discoveries using SeqStep+ with the +1 penalty and the training FDR threshold $\alpha_1 := 0.01$ as the criterion for selection of those hyperparameters. If none of the class weight combinations yields a positive average number of discoveries (which may happen for small data sets), then we follow the procedure described above, first dropping the +1 penalty, and then iteratively increasing the training FDR threshold by 0.005 while the procedure fails to report any discoveries.

2.5.5 Processing the Tailor scores in Percolator-RESET with two decoys

In practice, we obtain search results for the two-decoy Percolator-RESET by applying the Tide search engine twice, once for each of the two randomly generated peptide-shuffled decoy databases. Specifically, in each search, Tide is given the target database and a paired decoy database, and it searches for the best match in each of the two databases. The best target match for a given spectrum does not vary between the two searches, but that optimal target PSM will typically have different Tailor scores across the two searches. This is because the Tailor score normalizes the raw PSM score with respect to the quantiles of all the target and decoy matches to the given spectrum, and the decoys vary between the two searches [20]. Since we use the Tailor score for doing the peptide-level competition in Percolator-RESET, as well as for ordering the winning peptides to define the initial positive set, we address the aforementioned variation by averaging the two Tailor scores assigned for each target PSM.

2.5.6 Reporting PSMs

Both versions of Percolator-RESET promise theoretical FDR control only at the (representative) peptide, because the PSM level suffers from a multiplicity issue that violates the usual independence assumptions [12, 17]. Understandably, some sort of PSM-level analysis is often desirable. For example, when using an open search, multiple matches to the same stem peptide may correspond to multiple types of modifications of that peptide. Hence, similar to CONGA [6], we additionally

In-sample proportion	In-sample peptides	Entrapment peptides	In-sample:entrapment
100%	898	571298	1:636
75%	674	571298	1:848
50%	449	571298	1:1272
25%	225	571298	1:2539

Table 1: **The number of in-sample target peptides and entrapment-peptides for each combined target peptide database used in the entrapment runs.** The table reports the number of in-sample target peptides and entrapment target peptides for each of the four combined target databases The first column reports the proportion of the total in-sample target peptides used in the combined target database, the second is the number of these in-sample peptides used, the third column is the number of entrapment peptides, and the last column is the ratio of in-sample peptides to entrapment peptides.

report a list of PSMs that are associated with the originally discovered peptides. The process is essentially the same as the one used in CONGA: we first define a set of mass bins for each discovered (stem) peptide by a greedy clustering algorithm that takes into account the specified isolation window (Supplementary Algorithm 6). Then the highest (Tailor) scoring one of all the PSMs with the same peptide that fall into the same mass bin is reported. The list of PSMs is enhanced with a label for each PSM, which indicates whether the PSM scored above the SVM score threshold, giving the user additional confidence in those PSMs.

2.5.7 Availability

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).

2.6 Datasets

2.6.1 Entrapment runs

Entrapment runs were performed using data taken from the standard protein mix database, ISB18 [16]. We used the nine `.ms2` spectrum files taken from [17], which were originally sourced from the "Mix 7" spectrum files downloaded from `https://regis-web.systemsbiology.net/PublicDatasets/18_Mix/Mix_7/ORBITRAP/` (this excluded the tenth, smallest spectrum file). These `.ms2` spectrum files were subsequently used as input for Tide.

The in-sample database contained the 18 proteins of the mix and 30 additional hitchhiker proteins downloaded from `https://regis-web.systemsbiology.net/PublicDatasets/database`. We used the castor plant proteome as the entrapment database, as in [17]. Tide-index was used to virtually digest the in-sample and entrapment database. Four random subsets of varying proportions of the in-sample peptides were added to the entrapment peptides to create four combined target peptide databases. Tide-index was then used to create 100 randomly shuffled decoy database indices for each of the four. Param-medic [19] was used to determine any variable modifications, though none were detected. Any entrapment peptide that was identical (up to any leucine/isoleucine substitution) to an ISB18 peptide was removed from the entrapment database. Table 1 lists the resulting numbers of in-sample and entrapment peptides in each combined target peptide database, along with their in-sample-to-entrapment ratios.

Tide-search was used to search the nine `.ms2` spectrum files against each of the combined target and decoy databases with the following settings: for narrow searches we used `--auto-precursor-window warn --auto-mz-bin-width warn --use-tailor-calibration T --pin-output T`,

Project ID	Spectrum file	Var
PXD008920	QEP1_ZADAM_deg1_2_1_170615.mzid_QEP1_ZADAM_deg1_2_1_-170615.MGF	F
PXD008996	A431_01uM_DON_3_2.mgf	F
PXD010504	QX05437.mgf	F
PXD014277	Q06965_MS18-017_37_J9(55).mgf	F
PXD016724	190322-SI-0149-F5-01.mgf	F
PXD019354	Progenesis-Trophoplast-top5-140416.mgf	F
PXD024284	p2830_NaClvsHank_mascot.mgf	F
PXD025130	Q26431_MS20-025_Virus-purif_1.mgf	F
PXD029319	MP_16072020_LvN_S_layer_gps_5_DDA01.mgf	F
PXD030118	Q26756_MS20-013_C15.mgf	F
PXD002470	Mudpit_LC-iTRAQ1.RAW_(F004785).mzid_Mudpit_LC-iTRAQ1.RAW_-(F004785).MGF	T
PXD006856	export_Mzidentml_F134282.mgf	T
PXD010413	F373129.mgf	T
PXD012528	04222014_111_AF_160_22.mgf	T
PXD012611	Temp-3-Speclib-1.mzid_Temp-3-Speclib-1.MGF	T
PXD013274	PDL-2-11.mzid_PDL-2-11.MGF	T
PXD019186	QE1_Chrysi_Exp1_Sample12b.raw.-1.mgf	T
PXD022257	Sitaset1F6.mgf	T
PXD023571	Experiment_new_run_P6-GFP-PBANKA_1028300-Schizont-8hrs_from_OT_-200625_105_MS20-039_01.mzid_Experiment_new_run_P6-GFP__PBANKA_-1028300__Schizont__8hrs_from_OT_200625_105_MS20-039_01.MGF	T
PXD026895	191004_MK_ChRS19_100Z10C_pink_A.mzid_191004_MK_ChRS19 _100Z10C_-pink_A.MGF	T

Table 2: **The PRIDE-20 data set** The list of 20 spectrum files used in the PRIDE-20 data set and their associated project discoveries. The "Var" column indicates whether the data was deemed to include variable modifications.

and for open searches we used `--auto-mz-bin-width warn --precursor-window-type mass --precursor-window 100 --use -tailor-calibration T --pin-output T`. The option `--concat` was set to `T` for searches subsequently used by Percolator and the single-decoy Percolator-RESET and `F` for searches subsequently used by the two-decoy Percolator-RESET. We used Tide in Crux v4.1.6809338 [15].

2.6.2 PRIDE-20 searches

We downloaded 20 high-resolution spectrum files from the Proteomics Identifications Database, PRIDE [18]. Seven of these spectrum files were taken from [8] and were originally obtained by randomly selecting a spectrum file from randomly selected PRIDE projects (submitted no earlier than 2018). The remaining 13 spectrum files were similarly obtained by randomly selecting a single spectrum file from a randomly selected PRIDE project (submitted no earlier than 2019). The sampling was constrained to generate a collection of 10 spectrum files that have modifications and 10 spectrum files for which no modifications were detected, as determined in both cases by Parammedic [19]. The protein FASTA database files were also downloaded from the associated PRIDE projects or, in the case of human data, the UniProt database UP000005640 was used (downloaded 9/11/2021). Table 2 reports the list of the 20 spectrum files and PRIDE projects used.

For each of the PRIDE-20 data sets, we used Tide index 10 times to create 10 different decoy peptide databases, each of which was paired with the target to yield 10 target-decoys databases. We used `--auto-modifications T` to call Param-medic and account for variable modifications as detected. For narrow searches using Tide, we used the following options `--auto-precursor-window warn --auto-mz-bin-width warn --use-tailor-calibration T` and for open searches using Tide we used `--auto-mz-bin-width warn --precursor-window-type mass --precursor-window 100 --use-tailor-calibration T`. The option `--concat` was set to `T` for searches subsequently used by Percolator and the single-decoy Percolator-RESET, and to `F` for searches subsequently used by the two-decoy Percolator-RESET. We constructed `pin` files using the option `--pin-output T`, or using the utility function `make-pin`.

2.6.3 Percolator and Percolator-RESET settings

For each `pin` file, we used the default number of top 5 PSMs which allows for generating the feature `deltLCn` (the difference between the XCorr score of the top scoring PSM and the fifth/last ranked PSM, divided by the maximum of the top PSM's XCorr score and 1). In order to filter out hitchhiking neighbors, or peptides that might offer a fairly good match to a spectrum simply because they share many peaks with the top matching peptide, we subsequently kept only the top PSM for each spectrum. Lastly, we removed the `enzInt` feature from consideration (technically we set it to 0 for each PSM), as it was shown that this feature can compromise Percolator's FDR control by allowing it to distinguish between target and decoy PSMs [2]. Overall, the features used for training are `deltLCn, deltCn, Xcorr, TailorScore, PepLen, Charge, enzN, enzC, lnNumSP, dM, absdM` where `Charge` is represented by a one-hot vector indicating the charge state of the precursor. An explanation of each feature is given in Supplementary Table 3.

When running Percolator in Crux, we used `--only-psms T, --tdc F` in the crux percolator command, which reports only the PSMs and does not apply the TDC procedure within Percolator. This allowed us to apply the dynamic-level competition protocol (outlined in Section 2.5.3 and Algorithms 4a-4c) to Percolator's list of PSMs, using the learned percolator score. Furthermore we varied the option `--percolator-seed` during the entrapment experiments. Doing this randomizes the cross-validation folds between each call to Percolator.

Our Python implementation of the two-decoy Percolator-RESET takes in the `pin` files searched against the target and decoy databases separately. Therefore, when running this version of Percolator-RESET we adjusted some of the features so that they were calculated with respect to the combined target and decoy databases instead. These include `lnNumSP` (the number of candidate peptides in the combined database for each spectrum), `deltCn` (the difference between the XCorr score of the two top ranked PSMs with respect to the combined database, divided by maximum of the top PSM XCorr score and 1), and `deltLCn` (the difference between the XCorr score of the top scoring PSM and the fifth/last ranked PSM with respect to the combined database, divided by the maximum of the top PSM's XCorr score and 1).

3 Results

3.1 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, we first prove that both versions of RESET provide theoretical FDR control and then demonstrate this control using real data.

Any proof of FDR control must start by laying out its assumptions. In our case, the RESET procedures provide theoretical guarantee of FDR control if we make analogous assumptions to ones that form the basis of TDC [1, 12]. Specifically, for the single-decoy RESET, we assume that an incorrect target discovery is equally likely to end up winning the competition with its paired decoy (Step 1 of 2.3) as the decoy is, and moreover this happens independently of the winning scores, W, all features $\mathbf{x}$, and the labels L of all other discoveries. Because the enhanced RESET considers two decoys for each target instead of one, the above assumptions must be slightly adjusted to reflect the fact that for an incorrect target and its two derived decoys, each are equally likely to win their triplet competition. That is, we assume that the probability that an incorrect target discovery will win its triplet competition is 1/3 (or a decoy will win with probability 2/3), and again this happens independently given the same conditions as above.

The following theorem, whose proof is given in Supplementary Section A.2, underpins the theoretical basis of FDR control.

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

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.

To provide an empirical assessment of FDR control, we carried out a series of "entrapment" experiments [9]. 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 [7], 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 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, as specified in Section 2.6.1.

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 0.03, with an empirical FDR of 0.0319 (using 100% of the in-sample peptides and using an open-search mode). Given that we averaged the result over 100 decoys, this 6% 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

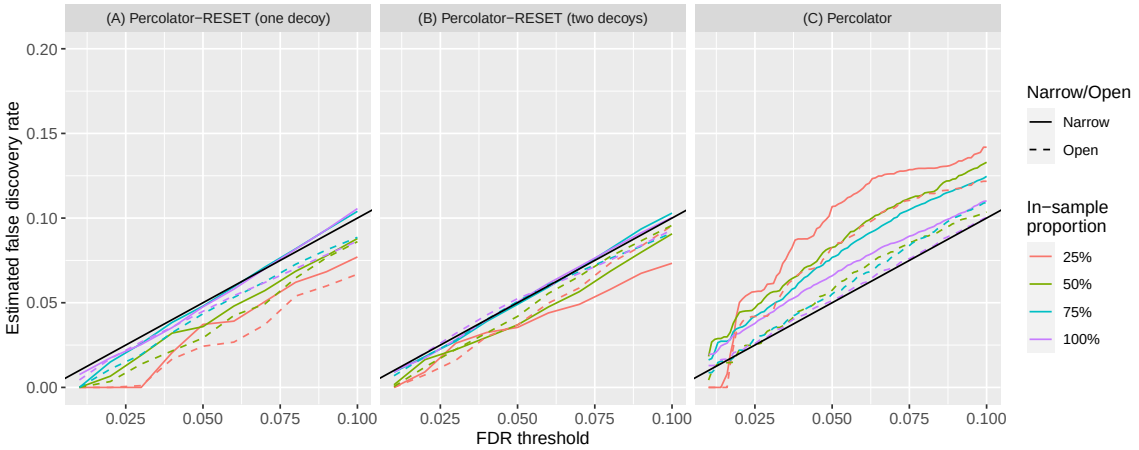

Figure 1: **Empirical FDRs of Percolator Algorithms.** The empirical FDR as estimated from our entrapment experiments on the ISB18 dataset using (A) single-decoy Percolator-RESET, (B) two-decoy Percolator-RESET, (C) Percolator. The FDP is estimated at a range of FDR thresholds ($[0.01, 0.1]$), and its 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.

(based on only 20 decoy databases), that Percolator struggles to control the FDR in these entrapment experiments [7]: 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 failure to control the FDR 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.

3.2 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) [18]. The datasets were essentially selected at random, with the constraint that Param-Medic [19] 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.

The results (Figure 2) 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 99.8% in narrow mode and 99% in open mode. At higher FDR thresholds, this difference in number of discoveries shrinks even more with the median average ratio now at 99.9% in both narrow mode and open mode at the 5% FDR threshold. Moreover, the two-decoy Percolator-RESET appears to marginally edge out Percolator, with median average ratios of 100.3% and 100.6% at a 1% FDR threshold and a median greater

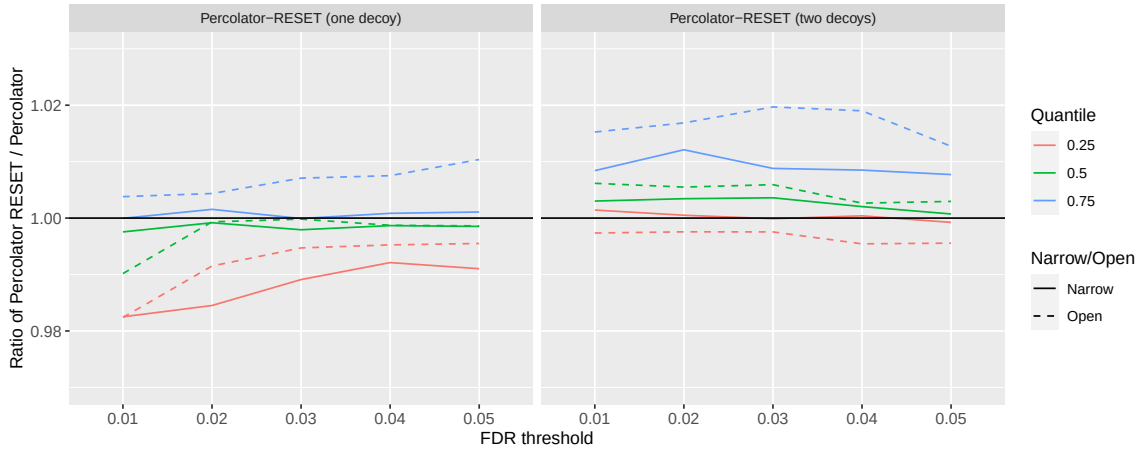

Figure 2: **Percolator-RESET proecdures discover similar number of peptides compared to Percolator.** The left panel displays the quartiles, with respect to the 20 PRIDE-20 datasets, of the mean ratio of the single-decoy Percolator-RESET discovered peptides to Percolator discovered peptides. The right panel similarly covers the mean ratio of the two-decoy Percolator-RESET discovered peptides to Percolator discovered peptides. Searches were conducted using Tide search. For the left panel, the mean ratio is taken with respect to each dataset's 10 decoys. For the right panel, 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.

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.

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

As suggested in Section 2.5, 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 3A 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 49.1% in both the narrow-search and open-search modes.

In the second experiment, we focused on the variation in the number of discoveries due to the

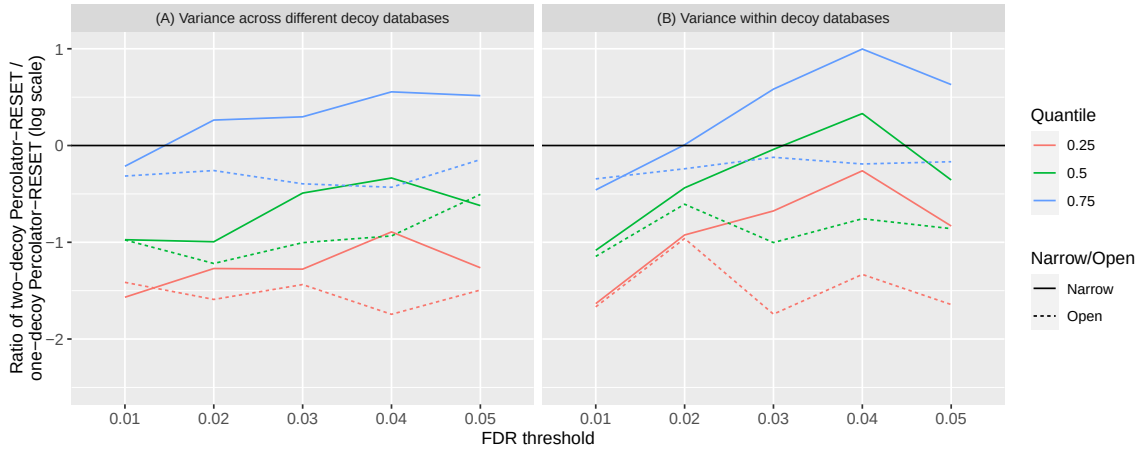

Figure 3: **Comparing the variabilities of Percolator-RESET with and without an extra decoy database.** Both panels display the quartiles, taken over the 20 PRIDE-20 datasets, of the $\log_2$ ratios of variances in the number of discoveries from two-decoy Percolator-RESET (numerator) to one-decoy Percolator-RESET (denominator). Each PRIDE-20 dataset had 10 decoy databases constructed. In (A), we calculated the sample variance for the two-decoy version using 10 applications, with each application using a unique pair selected from the 10 available decoy databases. The sample variance for the one-decoy version was calculated using 10 applications, each using a single decoy database from the 10 available decoy databases. The ratios are calculated as the sample variance of the two-decoy version to the one-decoy version. In (B), for each PRIDE-20 dataset, we preselected two decoy databases from the 10 available. We ran the two-decoy version 20 times using the preselected pair of decoy databases, while randomly varying the training/estimating decoy-splitting, and calculated the sample variance in the reported number of discoveries. We ran the single-decoy Percolator-RESET 10 times for each of the two decoy databases (a total of 20), and calculated the pooled sample variance: the average of the sample variance of the number of discoveries for each of the two sets of 10 applications. The ratios are then calculated as the sample variance of the two-decoy version to the pooled sample variance of the single-decoy version.

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 3B 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 52.8% and 54.8% at the 1% FDR threshold when using a narrow and open search respectively.

4 Discussion

The Percolator software takes as input tandem mass spectrometry search results and performs two distinct tasks: first, it uses machine learning to automatically re-rank the given set of PSMs, and second, it uses TDC to estimate the FDR in the resulting list. Our previous analysis in [7], as well as our analysis in Section 3.1, demonstrate that Percolator can link these two tasks in a problematic fashion, leading to underestimation of the FDR in the context of peptide detection.

Our proposed RESET procedures address this problem by introducing a decoy-splitting proce-

dure, which may be used in conjunction with other 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 [7]. The cross-validation was deemed necessary at the time, because Percolator uses the *entire* set of decoy wins as its negative set [11]. 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 dynamic-level competition is relevant beyond Percolator-RESET. Indeed, it is relevant to any application of TDC as it can be essential in validating TDC's assumption that an incorrect discovery is equally likely to be a target or a decoy win.

A Supplement

A.1 Features used by Percolator-RESET

Feature	Description
deltLCn	The difference between the XCorr score of the top scoring PSM and the fifth/last ranked PSM with respect to the combined database, divided by the maximum of the top PSM's XCorr score and 1
deltCn	The difference between the XCorr score of the two top ranked PSMs with respect to the combined database, divided by maximum of the top PSM XCorr score and 1
Xcorr	The SEQUEST cross-correlation PSM score
TailorScore	A calibrated PSM score (see [20] for details)
PepLen	The length of the matched peptide, in residues
Charge	The charge of the precursor ion (ranging from +1 to + 5)
enzN	A boolean indicating whether the associated peptide is preceded by a enzymatic (tryptic) site
enzC	A boolean indicating whether the associated peptide cotains an enzymatic C-terminus
lnNumSP	The natural logarithm of the number of database peptides within the specified precursor range
dm	The difference between the calculated and observed mass
absdM	The absolute value of the difference between the calculated and observed mass

Table 3: **List of described features used** The list of features considered when applying Percolator-RESET, adapted from `https://crux.ms/file-formats/features.html`.

A.2 Proof of Theorem 1

As noted in Section 2.3, RESET's discovery list is defined by applying SeqStep+ to the list of pseudo-targets (target wins plus the estimating decoys) ordered by RESET's learned scores. Accordingly, we can prove Theorem 1 by verifying that the labels of this list of pseudo-targets satisfy the conditions of Theorem 3 of [1]. The latter theorem deals with sequential hypothesis testing, where the null hypotheses H_j, for $j = 1, \ldots, m$, are sorted in some pre-determined order and it can be restated in our context as follows.

Theorem 3 (**of [1]**)**.** *Suppose that associated with each hypothesis H_j is a label $L_j \in \{-1, 1\}$ so that for any true null H_j, $P(L_j = 1) = c$ and $P(L_j = -1) = 1 - c$ independently of all other L_i for $i \neq j$. Then SeqStep+ (Algorithm 1) controls the FDR in the finite sample setting.*

Remark 1. The original formulation of Barber and Candès does not refer to any labels, but instead refers to p-values p_j that are associated with each hypothesis. Given the p-values we can define the labels as $L_j = \pm 1$ according to $p_j \leq c$ or $p_j > c$, and conversely given the labels we can define p-values of $p_j = c$ or $p_j = 1$ according to $L_j = 1$ or $L_j = -1$.

It follows that to establish our theorem we only need to analyze the probability distribution of the labels L_j of the true null pseudo-targets winning discoveries, i.e., the incorrect target and estimating decoy winning discoveries defined in steps 1-2 of RESET. For ease of reference, below we refer to the two-decoy version of RESET as RESET+, and to the single-decoy version of RESET as simply RESET. We first remind the reader of the necessary assumptions that we make.

Assumption 1. *For RESET, an incorrect target discovery is equally likely to end up winning the competition with its paired decoy as the decoy is, independently of all the winning scores, W, all the features $\mathbf{x}$, and the labels L of all other discoveries. For RESET+, an incorrect target discovery will win its triplet competition with probability 1/3 and a decoy will win it with probability 2/3, independently of all the winning scores, W, all the features $\mathbf{x}$, and the labels L of all other discoveries.*

Let

- N be the set of indices of the true null hypotheses (corresponding to the incorrect target discoveries);
- d be the number of decoys per target ($d = 1$ for RESET and $d = 2$ for RESET+);
- $\tilde{L}_j$ denote whether the jth discovery is a pseudo-target ($\tilde{L}_j = 1$) or a training decoy ($\tilde{L}_j = -1$).

By Assumption 1 and the definition of the training labels $\tilde{L}_j$, for each $j \in N$, independently of all the winning scores W, the features $\mathbf{x}$, and all other labels L_i for $i \neq j$ we have:

$$\begin{aligned}
P(L_j = 1, \tilde{L}_j = 1) &= P(L_j = 1) = 1/(d+1), \\
P(L_j = -1, \tilde{L}_j = 1) &= P(L_j = -1) \cdot P(\tilde{L}_j = 1 | L_j = -1) = d/(d+1) \cdot (1-s), \\
P(\tilde{L}_j = 1) &= P(L_j = 1, \tilde{L}_j = 1) + P(L_j = -1, \tilde{L}_j = 1) \\
&= 1/(d+1) + d/(d+1) \cdot (1-s) = [1 + d(1-s)]/(d+1), \\
P(L_j = 1 | \tilde{L}_j = 1) &= \frac{1/(d+1)}{[1+d(1-s)]/(d+1)} = \frac{1}{1+d(1-s)}, \\
P(L_j = -1 | \tilde{L}_j = 1) &= \frac{d(1-s)}{1+d(1-s)}.
\end{aligned} \tag{1}$$

Moreover, because the assignment of the pseudo-target/training decoy labels is determined for each hypothesis independently, and because we are only considering $\tilde{L}_j = 1$ above, the above probabilities are also independent of all the training labels $\tilde{L}$.

Let Π be any permutation, or reordering of the m hypotheses that is chosen based on information gleaned from all the winning scores W, the features $\mathbf{x}$, and all the training labels $\tilde{L}$. The above analysis shows that the last two lines of (1) hold under the new ordering, independently of all other labels L_i for $i \neq j$.

Because RESET's reordering is based exactly on W, $\mathbf{x}$, and the labels $\tilde{L}$, it follows that with $c = 1/[1 + d(1-s)]$ our labels satisfy the conditions of Theorem 3 above, and hence applying SeqStep+ to them controls the FDR.

A.3 Algorithms

Note that the detailed description of Percolator-RESET in Algorithm 3 in this section *omits step 1* from its outline in Section 2.5.1. The reason is, that this initial PSM-and-peptide dynamic-level competition is dependent on the size of the precursor tolerance (narrow or open search), whether variable modifications are considered, and whether one or two decoy-databases are used. Thus, it is conceptually easier to detail this step externally to Percolator-RESET, which we do in Algorithms 4a-4c for the single-decoy version, and Algorithms 5a-5c for the two-decoy version.

Algorithm 1 SeqStep / SeqStep+ (adopted from Selective Sequential Step+ of [1])

Input:
- $(W_i, L_i)_{i=1}^m$ the list of winning scores, and labels ($L_i = \pm 1$ indicating a target/decoy win);
- $c \in (0, 1)$ - the probability of a null target win;
- $\alpha \in (0, 1)$ - the FDR threshold;

Output: A discovery list R

sort the paired (W_i, L_i) in decreasing order of W_i ▷ ties are randomly broken
$D_k \leftarrow \#\{i \leq k : L_i = -1\}$ ▷ number of decoy wins in top k scores
$T_k \leftarrow \#\{i \leq k : L_i = 1\}$ ▷ number of target wins in top k scores
if SeqStep **then**
    $k_0 \leftarrow \max\{k : \frac{D_k}{T_k \vee 1} \cdot \frac{c}{1-c} \leq \alpha\}$
else if SeqStep+ **then**
    $k_0 \leftarrow \max\{k : \frac{D_k+1}{T_k \vee 1} \cdot \frac{c}{1-c} \leq \alpha\}$
end if
return $R \leftarrow \{i :$ the pre-sorted W_i is in the top k_0 ranks when sorted$\}$

Algorithm 2 RESET in feature-augmented TDC

Input:
- $\{(S_i^T, S_i^D, \mathbf{x}_i^T, \mathbf{x}_i^D) : i = 1, \ldots, m\}$ - the list of target scores, decoy scores, target features and decoy features;
- α - FDR threshold for the discovery list;
- s - the probability of assigning a decoy to the training set (default: $s = 1/2$);
- $S : \mathbb{R}^2 \to \mathbb{R}$ - a symmetric score function such that $S(x, y) = S(y, x)$;
- f - a semi-supervised machine learning model;
- RESET+ - a boolean indicating whether the two decoy version of RESET is used;

Output: A discovery list R

for $i = 1, \ldots, m$ **do** ▷ if $S_i^T = S_i^D$, randomly break ties
  if $S_i^T > S_i^D$ **then**
    $(W_i, \mathbf{x}_i, L_i) \leftarrow (S(W_i^T, W_i^D), \mathbf{x}_i^T, 1)$ ▷ e.g. $S(x, y) := x \vee y$ or $S(x, y) := |x - y|$
  else
    $(W_i, \mathbf{x}_i, L_i) \leftarrow (S(W_i^T, W_i^D), \mathbf{x}_i^D, -1)$
  end if
end for
$I \leftarrow$ a subset of the decoy win indices, $\{i : L_i = -1\}$, determined by randomly including each one with probability s
$\tilde{L}_i \leftarrow -1$ for $i \in I$ ▷ training decoys
$\tilde{L}_i \leftarrow 1$ for $i \in J := I^c$ ▷ pseudo-targets
$(\tilde{W}_i)_{i=1}^m \leftarrow f\left((W_i, \mathbf{x}_i, \tilde{L}_i)_{i=1}^m\right)$ ▷ where f is any machine learning model
if RESET+ **then** ▷ use the estimating decoys to define the list of discoveries in the two-decoys version
  $R \leftarrow \text{SeqStep+}((\tilde{W}_i, L_i)_{i \in J}, c = \frac{1}{3-2s}, \alpha)$
else ▷ same for single-decoy
  $R \leftarrow \text{SeqStep+}((\tilde{W}_i, L_i)_{i \in J}, c = \frac{1}{2-s}, \alpha)$
end if
return R

Algorithm 3 Percolator-RESET (one decoy)/Percolator-RESET+ (two decoys)

Input:
- $\{(W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - the list of winning scores, features and labels obtained from one of the PSM-and-peptide variants for Percolator-RESET (Algorithm 4a-4c) or for Percolator-RESET+ (Algorithm 5a-5c);
- α - FDR threshold for the discovery list;
- α_1 - FDR threshold for training (default: $\alpha_1 = 0.01$);
- `total_iter` - the number of SVM iterations (default: `total_iter` $= 5$);
- s - the probability of assigning a decoy to the training set (default: $s = 1/2$);
- RESET+ - a boolean indicating whether the two decoy version of RESET is used;

Output: A discovery list R

$I \leftarrow$ a subset of the decoy win indices, $\{i : L_i = -1\}$, determined by randomly including each one with probability s
$\tilde{L}_i \leftarrow -1$ for $i \in I$ ▷ training decoys
$\tilde{L}_i \leftarrow 1$ for $i \in J := I^c$ ▷ pseudo-targets
$\tilde{W}_i \leftarrow W_i$ ▷ the original winning scores are the initial SVM scores
if $RESET+$ **then**
  $c_t \leftarrow \frac{3-2s}{3}$
else
  $c_t \leftarrow \frac{2-s}{2}$
end if
for `iter` $\in \{1, \ldots, \texttt{total_iter}\}$ **do**
  $T^+ \leftarrow \text{SeqStep+}((\tilde{W}_i, \tilde{L}_i)_{i \in J}, c_t, \alpha_1)$ ▷ indices of positive training set determined by Algorithm 1
  if $T^+ = \emptyset$ **then**
    $T^+ \leftarrow \text{SeqStep}((\tilde{W}_i, \tilde{L}_i)_{i \in J}, c_t, \alpha_1)$ ▷ try to drop the +1 penalty if set is empty
  end if
  $\eta \leftarrow 0.005$
  while $T^+ = \emptyset$ **do**
    $T^+ \leftarrow \text{SeqStep}((\tilde{W}_i, \tilde{L}_i)_{i \in J}, c_t, \alpha_1 + \eta)$
    $\eta \leftarrow \eta + 0.005$ ▷ keep relaxing the FDR threshold until you get a non-empty set
  end while
  $T^- \leftarrow I$ ▷ indices of negative training set
  $(\tilde{W}_i)_{i=1}^m \leftarrow \texttt{applySVM}((W_i, \mathbf{x}_i, \tilde{L}_i)_{i=1}^m, T^+, T^-)$ ▷ apply SVM to rescore all peptides
end for
if $RESET+$ **then**
  $c_e \leftarrow \frac{1}{3-2s}$
else
  $c_e \leftarrow \frac{1}{2-s}$
end if
$R \leftarrow \text{SeqStep+}((\tilde{W}_i, L_i)_{i \in J}, c_e, \alpha)$ ▷ use the estimating decoys to define the list of discoveries
return R

Algorithm 4a PSM-and-peptide adapted from [17] allowing it to pass through extra features (unused here) and omitting its final step of applying SeqStep+ (Algorithm 1).

Input:
- $(\mathcal{T}, \mathcal{D})$ - the target and decoy databases;
- $\{(\sigma, \pi_T^\sigma, \pi_D^\sigma, S_T^\sigma, S_D^\sigma, \mathbf{x}_T^\sigma, \mathbf{x}_D^\sigma) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated scores, and their vector of features;
- `TD_pairs` - a mapping that pairs every target peptide to its associated decoy;

Output: $WP = \{(\pi_i, W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - a list of winning peptides, scores, features and labels ($L_i = \pm 1$ indicating a target/decoy win)

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  if $S_T^\sigma > S_D^\sigma$ **then** ▷ if $S_T^\sigma = S_D^\sigma$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \mathbf{x}_T^\sigma)$
  else
    $(\pi^\sigma, S^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^\sigma, S_D^\sigma, \mathbf{x}_D^\sigma)$
  end if
end for
for $\pi \in \mathcal{T} \cup \mathcal{D}$ **do** ▷ obtaining the best score associated for each peptide π
  $S_\pi \leftarrow \max\{S^\sigma \mid \pi^\sigma = \pi\}$ ▷ where $\max(\emptyset) = -\infty$
  $\mathbf{x}_\pi \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid \pi^\sigma = \pi, S^\sigma = S_\pi\}$ ▷ NULL if $S_\pi = -\infty$
end for
$WP \leftarrow \emptyset$ ▷ a list characterizing the winning target/decoy peptides
for $\pi \in \mathcal{T}$ **do** ▷ peptide-level competition
  if $S_\pi > S_{\texttt{TD_pairs}(\pi)}$ **then** ▷ if $S_\pi = S_{\texttt{TD_pairs}(\pi)}$, randomly break ties
    $WP \leftarrow WP \cup (\pi, S_\pi, \mathbf{x}_\pi, 1)$ ▷ add a target winning peptide
  else
    $WP \leftarrow WP \cup (\texttt{TD_pairs}(\pi), S_{\texttt{TD_pairs}(\pi)}, \mathbf{x}_{\texttt{TD_pairs}(\pi)}, -1)$ ▷ add a decoy winning peptide
  end if
end for
return WP

Algorithm 4b PSM-and-peptide adapted for variable modifications in narrow search

Input:
- $(\mathcal{T}^s, \mathcal{D}^s)$ - the target and decoy databases containing only the **stem** form peptides;
- $\{(\sigma, \pi_T^\sigma, \pi_D^\sigma, S_T^\sigma, S_D^\sigma, \mathbf{x}_T^\sigma, \mathbf{x}_D^\sigma) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated scores and their vector of features;
- `TD_pairs` - a mapping that pairs every target peptide to its associated decoy;
- `remove_mods` - a mapping that takes a peptide and returns its stem form;
- `sum_mods` - a mapping that takes a peptide and returns the total mass from each modification;

Output: $WS = \{(\pi_i, W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - a list of winning (representative) peptides, scores, features and labels ($L_i = \pm 1$ indicating a target/decoy win)

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  if $S_T^\sigma > S_D^\sigma$ **then** ▷ if $S_T^\sigma = S_D^\sigma$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \delta^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \texttt{remove_mods}(\pi_T^\sigma), \texttt{sum_mods}(\pi_T^\sigma), \mathbf{x}_T^\sigma)$
  else
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \delta^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^\sigma, S_D^\sigma, \texttt{remove_mods}(\pi_D^\sigma), \texttt{sum_mods}(\pi_D^\sigma), \mathbf{x}_D^\sigma)$
  end if
end for
for $\mathring{\pi} \in \mathcal{T}^s$ **do**
  $M_{\mathring{\pi}} \leftarrow \{\delta^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi} \text{ or } \mathring{\pi}^\sigma = \texttt{TD_pairs}(\mathring{\pi})\}$
  $M_{\texttt{TD_pairs}(\mathring{\pi})} \leftarrow M_{\mathring{\pi}}$
end for
for $\mathring{\pi} \in \mathcal{T}^s \cup \mathcal{D}^s$ **do** ▷ obtaining the best score with each stem peptide $\mathring{\pi}$ and mass mod δ
  for $\delta \in M_{\mathring{\pi}}$ **do**
    $S_{\mathring{\pi},\delta} \leftarrow \max\{S^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi}, \delta^\sigma = \delta\}$ ▷ where $\max(\emptyset) = -\infty$
    $\pi_{\mathring{\pi},\delta} \leftarrow$ randomly select from $\{\pi^\sigma \mid S^\sigma = S_{\mathring{\pi},\delta}, \mathring{\pi}^\sigma = \mathring{\pi}, \delta^\sigma = \delta\}$ ▷ NULL if $S_{\mathring{\pi},\delta} = -\infty$
    $\mathbf{x}_{\mathring{\pi},\delta} \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid S^\sigma = S_{\mathring{\pi},\delta}, \mathring{\pi}^\sigma = \pi_{\mathring{\pi},\delta}, \delta^\sigma = \delta\}$ ▷ NULL if $S_{\mathring{\pi},\delta} = -\infty$
  end for
end for
$WS \leftarrow \emptyset$ ▷ tuples characterizing the winning target/decoy stems
for $\mathring{\pi} \in \mathcal{T}^s$ **do** ▷ stem-level (w/modification mass) competition
  for $\delta \in M_{\mathring{\pi}}$ **do**
    if $S_{\mathring{\pi},\delta} > S_{\texttt{TD_pairs}(\mathring{\pi}),\delta}$ **then** ▷ if $S_{\mathring{\pi},\delta} = S_{\texttt{TD_pairs}(\mathring{\pi}),\delta}$, randomly break ties
      $WS \leftarrow WS \cup (\pi_{\mathring{\pi},\delta}, S_{\mathring{\pi},\delta}, \mathbf{x}_{\mathring{\pi},\delta}, 1)$ ▷ add a target winning representative peptide
    else
      $WS \leftarrow WS \cup (\pi_{\texttt{TD_pairs}(\mathring{\pi}),\delta}, S_{\texttt{TD_pairs}(\mathring{\pi}),\delta}, \mathbf{x}_{\texttt{TD_pairs}(\mathring{\pi}),\delta}, -1)$ ▷ add a decoy winning representative peptide
    end if
  end for
end for
return WS

Algorithm 4c PSM-and-peptide adapted for variable modifications in open search

Input: • $(\mathcal{T}^s, \mathcal{D}^s)$ - the target and decoy databases containing only the **stem** form peptides;
• $\{(\sigma, \pi_T^\sigma, \pi_D^\sigma, S_T^\sigma, S_D^\sigma, \mathbf{x}_T^\sigma, \mathbf{x}_D^\sigma) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated scores and their vector of features;
• `TD_pairs` - a mapping that pairs every target peptide to its associated decoy;
• `remove_mods` - a mapping that takes a peptide and returns its stem form;

Output: $WS = \{(\pi_i, W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - a list of winning (representative) peptides, scores, features and labels ($L_i = \pm 1$ indicating a target/decoy win)

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  if $S_T^\sigma > S_D^\sigma$ **then** ▷ if $S_T^\sigma = S_D^\sigma$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \texttt{remove_mods}(\pi_T^\sigma), \mathbf{x}_T^\sigma)$
  else
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^\sigma, S_D^\sigma, \texttt{remove_mods}(\pi_D^\sigma), \mathbf{x}_D^\sigma)$
  end if
end for
for $\mathring{\pi} \in \mathcal{T}^s \cup \mathcal{D}^s$ **do** ▷ obtaining the best score for each stem peptide $\mathring{\pi}$
  $S_{\mathring{\pi}} \leftarrow \max\{S^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi}\}$ ▷ where $\max(\emptyset) = -\infty$
  $\pi_{\mathring{\pi}} \leftarrow$ randomly select from $\{\pi^\sigma \mid S^\sigma = S_{\mathring{\pi}}, \mathring{\pi}^\sigma = \mathring{\pi}\}$ ▷ NULL if $S_{\mathring{\pi}} = -\infty$
  $\mathbf{x}_{\mathring{\pi}} \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid S^\sigma = S_{\mathring{\pi}}, \mathring{\pi}^\sigma = \pi_{\mathring{\pi}}\}$ ▷ NULL if $S_{\mathring{\pi}} = -\infty$
end for
$WS \leftarrow \emptyset$ ▷ tuples characterizing the winning target/decoy stems
for $\mathring{\pi} \in \mathcal{T}^s$ **do** ▷ stem-level competition
  if $S_{\mathring{\pi}} > S_{\texttt{TD_pairs}(\mathring{\pi})}$ **then** ▷ if $S_{\mathring{\pi}} = S_{\texttt{TD_pairs}(\mathring{\pi})}$, randomly break ties
    $WS \leftarrow WS \cup (\pi_{\mathring{\pi}}, S_{\mathring{\pi}}, \mathbf{x}_{\mathring{\pi}}, 1)$ ▷ add a target winning representative peptide
  else
    $WS \leftarrow WS \cup (\pi_{\texttt{TD_pairs}(\mathring{\pi})}, S_{\texttt{TD_pairs}(\mathring{\pi})}, \mathbf{x}_{\texttt{TD_pairs}(\mathring{\pi})}, -1)$ ▷ add a decoy representative peptide
  end if
end for
return WS

Algorithm 5a PSM-and-peptide+

Input: • $(\mathcal{T}, \mathcal{D}_1, \mathcal{D}_2)$ - the target and two decoy databases;
• $\{(\sigma, \pi_T^\sigma, \{\pi_D^{\sigma,i}\}_{i=1}^2, \{S_T^{\sigma,i}\}_{i=1}^2, \{S_D^{\sigma,i}\}_{i=1}^2, \mathbf{x}_T^\sigma, \{\mathbf{x}_D^{\sigma,i}\}_{i=1}^2) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated (Tailor) scores and their vector of features;
• $\{\texttt{TD_match}_i\}_{i=1}^2$ - a set of mappings that takes every target peptide to its associated ith decoy;

Output: $WP = \{(\pi_i, W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - a list of winning peptides, scores, features and labels ($L_i = \pm 1$ indicating a target/decoy win)

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  $S_T^\sigma \leftarrow \frac{1}{2}\left(S_T^{\sigma,1} + S_T^{\sigma,2}\right)$ ▷ average the two (Tailor) target scores
  if $S_T^\sigma > S_D^{\sigma,i}$ for $i = 1, 2$ **then** ▷ if any $S_T^\sigma = S_D^{\sigma,i}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \mathbf{x}_T^\sigma)$
  else
    $i_0 \leftarrow \arg\max_i S_D^{\sigma,i}$ ▷ if $S_D^{\sigma,1} = S_D^{\sigma,2}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^{\sigma,i_0}, S_D^{\sigma,i_0}, \mathbf{x}_D^{\sigma,i_0})$
  end if
end for
for $\pi \in \mathcal{T} \cup \mathcal{D}_1 \cup \mathcal{D}_2$ **do** ▷ obtaining the best score associated for each peptide π
  $S_\pi \leftarrow \max\{S^\sigma \mid \pi^\sigma = \pi\}$ ▷ where $\max(\emptyset) = -\infty$
  $\mathbf{x}_\pi \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid \pi^\sigma = \pi, S^\sigma = S_\pi\}$ ▷ NULL if $S_\pi = -\infty$
end for
$WP \leftarrow \emptyset$ ▷ tuples characterizing the winning target/decoy peptides
for $\pi \in \mathcal{T}$ **do** ▷ peptide-level competition
  if $S_\pi > S_{\texttt{TD_match}_i(\pi)}$ for $i = 1, 2$ **then** ▷ if any $S_\pi = S_{\texttt{TD_match}_i(\pi)}$, randomly break ties
    $WP \leftarrow WP \cup (\pi, S_\pi, \mathbf{x}_\pi, 1)$ ▷ add a target winning peptide
  else
    $i_0 \leftarrow \arg\max_i S_{\texttt{TD_match}_i(\pi)}$ ▷ randomly break ties
    $WP \leftarrow WP \cup (\texttt{TD_match}_{i_0}(\pi), S_{\texttt{TD_match}_{i_0}(\pi)}, \mathbf{x}_{\texttt{TD_match}_{i_0}(\pi)}, -1)$ ▷ add a decoy winning peptide
  end if
end for
return WP

Algorithm 5b PSM-and-peptide+ for variable modifications in narrow search

Input:
- $(\mathcal{T}^s, \mathcal{D}_1^s, \mathcal{D}_2^s)$ - the target and two decoy databases containing only the **stem** form peptides;
- $\{(\sigma, \pi_T^\sigma, \{\pi_D^{\sigma,i}\}_{i=1}^2, \{S_T^{\sigma,i}\}_{i=1}^2, \{S_D^{\sigma,i}\}_{i=1}^2, \mathbf{x}_T^\sigma, \{\mathbf{x}_D^{\sigma,i}\}_{i=1}^2) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated (Tailor) scores and their vector of features;
- $\{\texttt{TD_match}_i\}_{i=1}^2$ - a set of mappings that maps every target peptide to its associated ith decoy;
- `remove_mods` - a mapping that takes a peptide and returns its stem;
- `sum_mods` - a mapping that takes a peptide and returns the total mass from each modification;

Output: $WS = \{(\pi_i, W_i, \mathbf{x}_i, L_i) : i = 1, \ldots, m\}$ - a list of winning (representative) peptides, scores, features and labels ($L_i = \pm 1$ indicating a target/decoy win)

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  $S_T^\sigma \leftarrow \frac{1}{2}\left(S_T^{\sigma,1} + S_T^{\sigma,2}\right)$ ▷ average the two (Tailor) target scores
  if $S_T^\sigma > S_D^{\sigma,i}$ for $i = 1, 2$ **then** ▷ if any $S_T^\sigma = S_D^{\sigma,i}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \delta^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \texttt{remove_mods}(\pi_T^\sigma), \texttt{sum_mods}(\pi_T^\sigma), \mathbf{x}_T^\sigma)$
  else
    $i_0 \leftarrow \arg\max_i S_D^{\sigma,i}$ ▷ if $S_D^{\sigma,1} = S_D^{\sigma,2}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \delta^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^{\sigma,i_0}, S_D^{\sigma,i_0}, \texttt{remove_mods}(\pi_D^{\sigma,i_0}), \texttt{sum_mods}(\pi_D^{\sigma,i_0}), \mathbf{x}_D^{\sigma,i_0})$
  end if
end for
for $\mathring{\pi} \in \mathcal{T}^s$ **do**
  $M_{\mathring{\pi}} \leftarrow \{\delta^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi} \text{ or } \mathring{\pi}^\sigma = \texttt{TD_pairs}_1(\mathring{\pi}) \text{ or } \mathring{\pi}^\sigma = \texttt{TD_pairs}_2(\mathring{\pi})\}$
  $M_{\texttt{TD_pairs}_i(\mathring{\pi})} \leftarrow M_{\mathring{\pi}}$ for $i = 1, 2$
end for
for $\mathring{\pi} \in \mathcal{T}^s \cup \mathcal{D}_1^s \cup \mathcal{D}_2^s$ **do** ▷ obtaining the best score for each stem $\mathring{\pi}$ and mass mod δ
  for $\delta \in M_{\mathring{\pi}}$ **do**
    $S_{\mathring{\pi},\mu} \leftarrow \max\{S^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi}, \delta^\sigma = \delta\}$ ▷ where $\max(\emptyset) = -\infty$
    $\pi_{\mathring{\pi},\delta} \leftarrow$ randomly select from $\{\pi^\sigma \mid S^\sigma = S_{\mathring{\pi},\delta}, \mathring{\pi}^\sigma = \mathring{\pi}, \delta^\sigma = \delta\}$ ▷ NULL if $S_{\mathring{\pi},\delta} = -\infty$
    $\mathbf{x}_{\mathring{\pi},\delta} \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid S^\sigma = S_{\mathring{\pi},\delta}, \mathring{\pi}^\sigma = \pi_{\mathring{\pi},\delta}, \delta^\sigma = \delta\}$ ▷ NULL if $S_{\mathring{\pi},\delta} = -\infty$
  end for
end for
$WS \leftarrow \emptyset$ ▷ tuples characterizing the winning target/decoy stems
for $\mathring{\pi} \in \mathcal{T}^s$ **do** ▷ stem-level (w/modification mass) competition
  for $\delta \in M_{\mathring{\pi}}$ **do**
    if $S_{\mathring{\pi},\delta} > S_{\texttt{TD_match}_i(\mathring{\pi}),\delta}$ for $i = 1, 2$ **then** ▷ if any $S_{\mathring{\pi},\delta} = S_{\texttt{TD_match}_i(\mathring{\pi}),\delta}$, randomly break ties
      $WS \leftarrow WS \cup (\pi_{\mathring{\pi},\delta}, S_{\mathring{\pi},\delta}, \mathbf{x}_{\mathring{\pi},\delta}, 1)$ ▷ add a target winning representative peptide
    else
      $i_0 \leftarrow \arg\max_i S_{\texttt{TD_match}_i(\mathring{\pi}),\delta}$ ▷ randomly break ties
      $WS \leftarrow WS \cup (\pi_{\texttt{TD_match}_{i_0}(\mathring{\pi}),\delta}, S_{\texttt{TD_match}_{i_0}(\mathring{\pi}),\delta}, \mathbf{x}_{\texttt{TD_match}_{i_0}(\mathring{\pi}),\delta}, -1)$ ▷ add a decoy winning representative peptide
    end if
  end for
end for
return WS

Algorithm 5c PSM-and-peptide+ for variable modifications in open search

Input:
- $(\mathcal{T}^s, \mathcal{D}_1^s, \mathcal{D}_2^s)$ - the target and two decoy databases containing only the **stem** form peptides;
- $\{(\sigma, \pi_T^\sigma, \{\pi_D^{\sigma,i}\}_{i=1}^2, \{S_T^{\sigma,i}\}_{i=1}^2, \{S_D^{\sigma,i}\}_{i=1}^2, \mathbf{x}_T^\sigma, \{\mathbf{x}_D^{\sigma,i}\}_{i=1}^2) : \sigma \in \Sigma\}$ - the set of experimental spectra, their target and decoy matches, their associated (Tailor) scores and their vector of features;
- $\{\texttt{TD_match}_i\}_{i=1}^2$ - a set of mappings that takes every target peptide to its associated ith decoy;
- `remove_mods` - a mapping that takes a peptide and returns its stem;

Output: WS - a list of winning representative peptides, scores, features and labels

for $\sigma \in \Sigma$ **do** ▷ PSM-level competition
  $S_T^\sigma \leftarrow \frac{1}{2}\left(S_T^{\sigma,1} + S_T^{\sigma,2}\right)$ ▷ average the two (Tailor) target scores
  if $S_T^\sigma > S_D^{\sigma,i}$ for $i = 1, 2$ **then** ▷ if any $S_T^\sigma = S_D^{\sigma,i}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_T^\sigma, S_T^\sigma, \texttt{remove_mods}(\pi_T^\sigma), \mathbf{x}_T^\sigma)$
  else
    $i_0 \leftarrow \arg\max_i S_D^{\sigma,i}$ ▷ if $S_D^{\sigma,1} = S_D^{\sigma,2}$, randomly break ties
    $(\pi^\sigma, S^\sigma, \mathring{\pi}^\sigma, \mathbf{x}^\sigma) \leftarrow (\pi_D^{\sigma,i_0}, S_D^{\sigma,i_0}, \texttt{remove_mods}(\pi_D^{\sigma,i_0}), \mathbf{x}_D^{\sigma,i_0})$
  end if
end for
for $\mathring{\pi} \in \mathcal{T}^s \cup \mathcal{D}_1^s \cup \mathcal{D}_2^s$ **do** ▷ obtaining the best score for each stem peptide $\mathring{\pi}$
  $S_{\mathring{\pi}} \leftarrow \max\{S^\sigma \mid \mathring{\pi}^\sigma = \mathring{\pi}\}$ ▷ where $\max(\emptyset) = -\infty$
  $\pi_{\mathring{\pi}} \leftarrow$ randomly select from $\{\pi^\sigma \mid S^\sigma = S_{\mathring{\pi}}, \mathring{\pi}^\sigma = \mathring{\pi}\}$ ▷ NULL if $S_{\mathring{\pi}} = -\infty$
  $\mathbf{x}_{\mathring{\pi}} \leftarrow$ randomly select from $\{\mathbf{x}^\sigma \mid S^\sigma = S_{\mathring{\pi}}, \mathring{\pi}^\sigma = \pi_{\mathring{\pi}}\}$ ▷ NULL if $S_{\mathring{\pi}} = -\infty$
end for
$WS \leftarrow \emptyset$ ▷ tuples characterizing the winning target/decoy stems
for $\mathring{\pi} \in \mathcal{T}^s$ **do** ▷ stem-level competition
  if $S_{\mathring{\pi}} > S_{\texttt{TD_match}_i(\mathring{\pi})}$ for $i = 1, 2$ **then** ▷ if any $S_{\mathring{\pi}} = S_{\texttt{TD_match}_i(\mathring{\pi})}$, randomly break ties
    $WS \leftarrow WS \cup (\pi_{\mathring{\pi}}, S_{\mathring{\pi}}, \mathbf{x}_{\mathring{\pi}}, 1)$ ▷ add a target winning stem
  else
    $i_0 \leftarrow \arg\max_i S_{\texttt{TD_match}_i(\mathring{\pi})}$ ▷ randomly break ties
    $WS \leftarrow WS \cup (\pi_{\texttt{TD_match}_{i_0}(\mathring{\pi})}, S_{\texttt{TD_match}_{i_0}(\mathring{\pi})}, \mathbf{x}_{\texttt{TD_match}_{i_0}(\mathring{\pi})}, -1)$ ▷ add a decoy winning stem
  end if
end for
return WS

Algorithm 6 Precursor mass clustering (`precursor_clustering`)

Input: • $\{(\rho_i, c_i) : i = 1, \ldots, n\}$ - a set of precursor masses and charge states associated with a given stem peptide;
• (w_l, w_u) - the experimental isolation window given as a pair of values;

Output: $\{P_j\}$ - the clustered precursors;

```
Reorder the precursors so that ρ_1 ≤ ··· ≤ ρ_n
j ← 1
P_j ← {1}
for i ∈ {2, ..., n} do
    if ρ_i ≤ ρ_{i−1} + max{c_{i−1} w_u, c_i w_l} then
        P_j ← P_j ∪ {i}
    else
        j ← j + 1
        P_j ← {i}
    end if
end for
return {P_j}
```

Algorithm 7 Generating the auxiliary list of target PSMs with variants of the discovered peptides

Input: • $\{\pi_i^* : i = 1, \ldots, M\}$ - the set of peptide discoveries;
• $\{(\sigma_i, \pi_i, \rho_i, c_i, S_i, \tilde{S}_i) : i = 1, \ldots, m\}$ - a set of tuples characterizing the m target PSMs where σ_i =spectrum, π_i = peptide, ρ_i =precursor mass, c_i =precursor charge, S_i =PSM score, $\tilde{S}_i$ =SVM score;
• (w_l, w_u) - the experimental isolation window given as a pair of values;
• S - the score threshold used to define the FDR-controlled list of discovered peptides;

Output: • $\{(\sigma_i, \pi_i, \rho_i, c_i, S_i, \tilde{S}_i)\}_{i \in J}$ - the augmented list of PSMs to be reported;
• $\{a_i : i \in J\}$ - labels indicating if the corresponding PSM is greater than or equal to the SVM score threshold ($a_i = 1$) or if it is below ($a_i = 0$);

```
a_i ← 0 for all i = 1, ..., m
J ← ∅                                   ▷ J is the indices of the PSMs to be returned
for π* ∈ {π*_1, ..., π*_M} do
    I ← {i | π* = π_i}                  ▷ all PSMs matched with the discovered peptide π*
    Cs ← precursor_clustering({(ρ_i, c_i)}_{i∈I}, (w_l, w_u))   ▷ applying Algorithm 6
    for C ∈ Cs do
        S_x ← max{S_j | j ∈ C}          ▷ maximal score for the considered cluster of precursor masses
        if S_x > −∞ then                                          ▷ max{∅} = −∞
            l ← randomly select from {i | S_i = S_x}
            J ← J ∪ {l}                 ▷ add a maximal scoring PSM of the considered cluster
            if S̃_l ≥ S then
                a_l ← 1                 ▷ the SVM-score of this maximal PSM is above the discovery threshold
            end if
        end if
    end for
end for
return {(σ_i, π_i, ρ_i, c_i, S_i, S̃_i, a_i)}_{i∈J}
```

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] Yulia Danilova, Anastasia Voronkova, Pavel Sulimov, and Attila Kertész-Farkas. Bias in false discovery rate estimation in mass-spectrometry-based peptide identification. *Journal of proteome research*, 18(5):2354–2358, 2019.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] J. Freestone, W. S. Noble, and U. Keich. Re-investigating the correctness of decoy-based false discovery rate control in proteomics tandem mass spectrometry. In preparation, 2023.

[8] J. Freestone, T. Short, W. S. Noble, and U. Keich. Group-walk: a rigorous approach to group-wise false discovery rate analysis by target-decoy competition. *Bioinformatics*, 38(Supplement 2):ii82–ii88, 09 2022.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. Kertesz-Farkas, F. L. Nii Adoquaye Acquaye, K. Bhimani, J. K. Eng, W. E. Fondrie, C. Grant, M. R. Hoopmann, A. Lin, Y. Y. Lu, R. L. Moritz, M. J. MacCoss, and W. S. Noble. The Crux Toolkit for Analysis of Bottom-Up Tandem Mass Spectrometry Proteomics Data. *Journal of Proteome Research*, 22(2):561–569, Feb 2023.

[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] 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.

[18] 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.

[19] 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.

[20] 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.