Supplement to "Direct maximization of protein identifications from tandem mass spectra"

Marina Spivak
Department of Machine Learning,
NEC Laboratories America,
4 Independence Way,
Princeton, NJ 08540
marina@nec-labs.com

Jason Weston
Department of Machine Learning,
NEC Laboratories America,
4 Independence Way,
Princeton, NJ 08540
jasonw@nec-labs.com

Michael J. MacCoss
Department of Genome Sciences
1705 NE Pacific Street
Box 355065
University of Washington
Seattle, WA 98195
maccoss@u.washington.edu

William Stafford Noble
Department of Genome Sciences
Department of Computer Science and Engineering
1705 NE Pacific Street
Box 355065
University of Washington
Seattle, WA 98195
william-noble@u.washington.edu

September 24, 2009

1 The Barista model

We are given a set of observed spectra $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_S}\}$ and a database $\mathcal{D}$ of target and decoy proteins against which we perform a database search. The search produces a set of peptide-spectrum matches (PSMs). Denoting the set of peptides as $E_1, \ldots, E_{N_E}$, the PSMs are written as the tuples $(E_i, \mathbf{s}_j) \in \mathcal{M}$, each representing a match of peptide i to spectrum j. Note that, in general, we may opt to retain the single best-scoring peptide for each spectrum, or a small constant number of top-ranked PSMs per spectrum. Each of the identified peptides E_k belongs to one or more proteins, leading to a set of proteins $\mathbf{R}_1, \ldots, \mathbf{R}_{N_R}$ that cover the set of peptides. Thus, $\mathbf{R}$ includes every protein in $\mathcal{D}$ that has at least one identified peptide (i.e. the maximal set of proteins that can explain the observed spectra).

For our algorithm, we define a feature representation $\phi(E, s) \in \mathbb{R}^d$ for any given PSM. Our particular choice for this feature representation, which is described in Supplementary Table 1, contains a variety of scores of the quality of the peptide-spectrum match, as well as features that capture properties of the spectrum and properties of the peptide.

1.1 PSM Scoring Function

We now define the score of a PSM to be a parameterized function of its feature vector $\phi(E, s)$. We consider two possibilities.

Linear Parameterization Previous works used a family of linear functions of the form:

$$f(E, s) = \mathbf{w}^\top \phi(E, s) + b,$$

where $\mathbf{w} \in \mathbb{R}^d$. This is the model chosen by methods such as PeptideProphet [Keller et al., 2002] and Percolator [Käll et al., 2007].

Nonlinear Parameterization We choose a family of nonlinear functions given by two-layer neural networks:

$$f(E,s) = \sum_{i=1}^{\mathcal{HU}} \mathbf{w}_i^O h_i(\phi(E,s)) + b,$$

where $\mathbf{w}^O \in \mathbb{R}^{\mathcal{HU}}$ are the output layer weights for the $\mathcal{HU}$ hidden units, and $h_k(\phi(E,s))$ is the k^{th} hidden unit, defined as:

$$h_k(\phi(E,s)) = \tanh((\mathbf{w}_k^H)^\top \phi(E,s) + b_k),$$

where $\mathbf{w}_k^H \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ are the weight vector and threshold for the k^{th} hidden unit. The number of hidden units $\mathcal{HU}$ is a hyperparameter that can be chosen by cross-validation. This nonlinear function is the improved model used in Q-ranker [Spivak et al., 2009]. Throughout this work, we use a fixed value of 3 hidden units. In preliminary experiments, we observed that 3 or 4 hidden units provided approximately the same performance, whereas using 5 hidden units led to evidence of over-fitting.

1.2 Peptide Scoring Function

A single peptide can have several spectra matching to it (several PSMs). For each distinct peptide we would like to rank the likelihood that they have been matched. Hence, we define the score of a peptide as the maximum score assigned to any of its PSMs:

$$g(E) = \max_{s:(E,s)\in\mathcal{M}} f(E,s)$$

where $(E,s) \in \mathcal{M}$ is the set of PSMs assigned to peptide E. We take the max over the PSMs for each peptide because of the argument presented in [Nesvizhskii et al., 2003], that many spectra matching the same peptide are not an indication of the correctness of the identification.

1.3 Protein Scoring Function

Finally, the score of a protein is defined in terms of the scores of the peptides in that protein as follows:

$$F(R) = \frac{1}{|N(R)|^\alpha} \sum_{E\in N'(R)} g(E) \tag{1}$$

where $N(R)$ is the set of predicted peptides in protein R, $N'(R)$ is the set of peptides in the protein R that were observed during the MS/MS experiment, and α is a hyperparameter of the model. The set $N(R)$ is created by virtually digesting the protein database $\mathcal{D}$ with the protease used to digest the protein mixture for the mass spectrometry experiment. Therefore, the sum of the scores of all the peptides identified during the database seach is used to estimate the accuracy of the protein identification. Dividing by a function of the predicted number of peptides is designed to correct for the number of the peptides not identified during the database search. Setting $\alpha = 1$ penalizes linearly, whereas setting $\alpha < 1$ punishes larger sets of peptides to a lesser degree - for example, this can be used if not all peptides in a protein are observervable. In our results we use the fixed value $\alpha = 0.3$, after selecting it in validation experiments (Supplementary Figure 10).

2 Training the model

The training proceeds as follows (see Supplementary Algorithm 1). Draw a protein R at random and determine its score $F(R)$ based on the scores of its peptides. Because the parameters $\mathbf{w}$ of the PSM scoring function $f(E,s)$ change during training, the scores of all PSMs belonging to the peptides are recalculated, and a max operation is performed each time a protein is drawn.

For each protein $\mathbf{R}_i \in \mathcal{D}$ we also have a label $\mathbf{y}_i \in \pm 1$ indicating whether it is a target (positive) or decoy (negative). Given our set of proteins R and corresponding labels $\mathbf{y}$, the goal is to choose the parameters $\mathbf{w}$ of the discriminant function $F(R)$, such that

$$\begin{aligned} F(R) > 0 \quad &\text{if } \ y_i = 1 \\ F(R) < 0 \quad &\text{if } \ y_i = -1. \end{aligned}$$

To find $F(R)$ we search for the function in the family that best fits the empirical data. The quality of the fit is measured using a loss function $L(F(R), y)$ which quantifies the discrepancy between the values of $F(R)$ and the true labels y. We thus train the weights $\mathbf{w}$ using stochastic gradient descent with the hinge loss function [Cortes and Vapnik, 1995]

$$L(F(R), y) = \max(0, 1 - yF(R))$$

During training, the gradients $\frac{\delta L(F(R),y)}{\delta \mathbf{w}}$ of the loss function are calculated with respect to each weight w, and the weights are updated. After convergence, the final output is a ranked list of proteins, sorted by score.

3 Multi-task training

For the multi-task version of Barista, we train the protein and peptide optimization tasks in parallel using a shared neural network representation. For the protein-level training, we use the the hinge loss to optimize $L_{prot}(F(\mathbf{R}_i), y_i) = \max(0, 1 - y_i F(\mathbf{R}_i))$ and follow the procedure outlined above. For peptide ranking we use a similar procedure: we pick a peptide example, E_i, and we assign this peptide a label based on the target/decoy labels of the corresponding proteins. We then make a gradient step to optimize the hinge loss function on the peptide level: $L_{pep}(g(\mathbf{E}_j), y_j) = \max(0, 1 - y_j g(\mathbf{E}_j))$.

To learn both tasks simultaneously, we optimize $L_{multi} = L_{prot}(F(\mathbf{R}_i), y_i) + L_{pep}(g(\mathbf{E}_j), y_j)$. The training follows the procedure discribed in [Collobert and Weston, 2008]:

1. Select next task.
2. Select a random training example for this task.
3. Update the NN for this task by taking a gradient step with respect to this exampe.
4. Go to 1.

4 Degeneracy

For *degenerate peptides*—peptides that appear in several proteins—our approach is as follows:

1. Merge all proteins that contain a common set of identified peptides into a single meta-protein, and count it as a single protein in all the reported results.
2. Identify proteins whose peptides are completely contained in another protein, and report only the larger protein.
3. For proteins sharing only a portion of their peptides, we propose two solutions: non-parsimonious and parsimonious. By default, Barista returns a non-parsimonious solution, which is simply a ranking of proteins after the two steps above. The parsimonious solution (referred to as *p-Barista*) is as described in [Bern and Goldberg, 2008]: the final protein scores are composed such that if several proteins share at least one peptide, then this peptide is assigned only to the highest-scoring protein in the group and does not contribute to the score of any other protein.

5 Running PeptideProphet, ProteinProphet and IDPicker

We used the versions of PeptideProphet and ProteinProphet from the Trans Proteomic Pipeline version 4.0. Each data set was analyzed using PeptideProphet with the appropriate enzyme specificity and decoy option. The default peptide probability of 0.05 assigned by PeptideProphet was used to filter the input for further analysis by ProteinProphet. IDPicker version 2.0 was run using four different FDR thresholds: 0.01, 0.05, 0.1 and 0.25.

6 Defining a gold standard based on external data sets

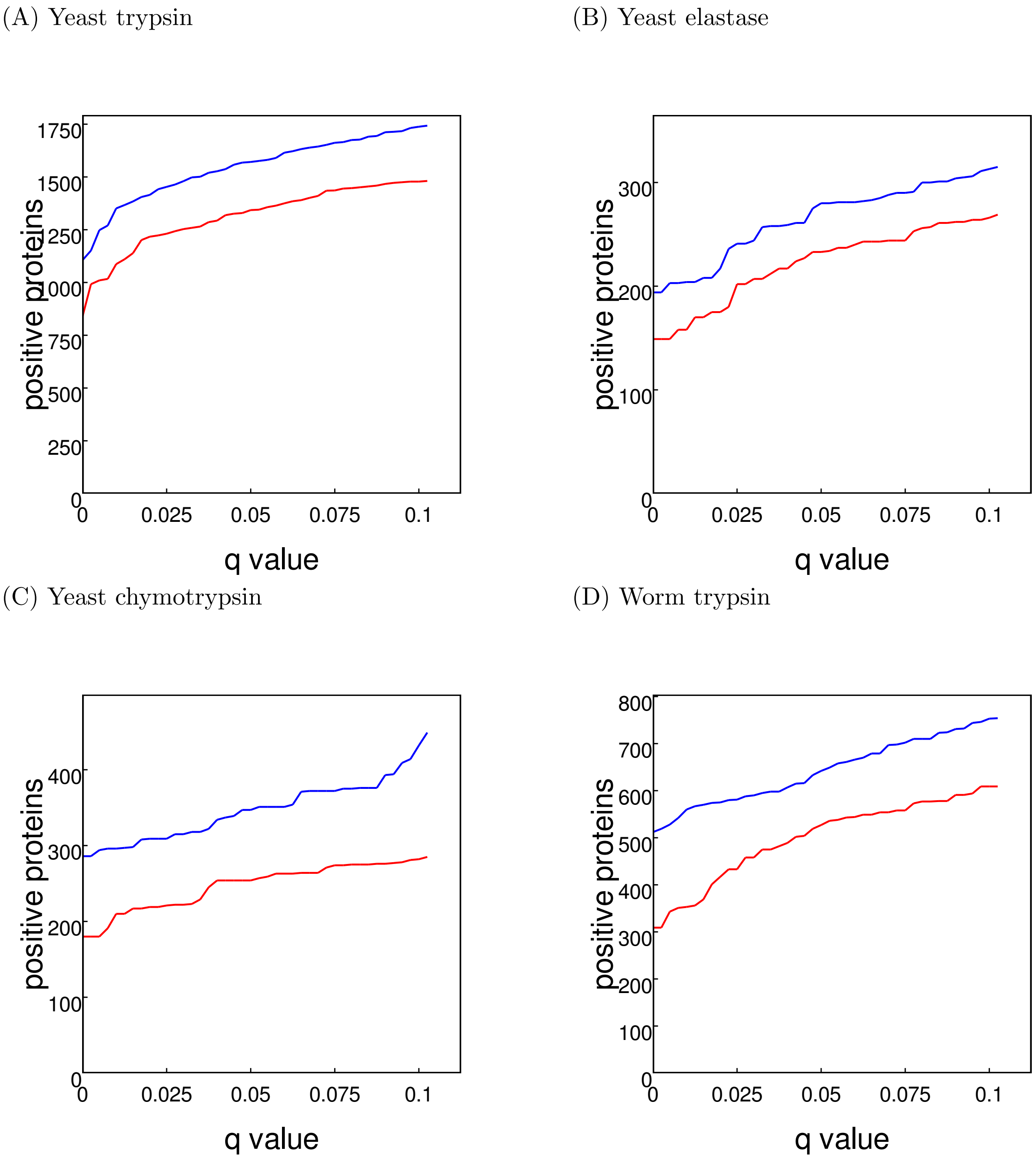
For our ROC analysis, we treated as positives the intersection of the protein sets identified by the mRNA [Holstege et al., 1998] and protein-tagging experiments [Ghaemmaghami et al., 2003]. The following thresholds applied to the datasets: (1) all 1627 proteins whose mRNA copy count was higher than the average copies/cell counts (2.4 copies/cell) were considered as present according to the microarray experiments, and (2) all 3790 proteins detected by both GFP (green flourenscent protein) and TAP (a specific antigen) were considered present according to the protein-tagging experiment. The intersection of these two sets contains 1295 proteins, and was used as an independent gold standard.

1	XCorr	Cross correlation between calculated and observed spectra
2	ΔC_n	Fractional difference between current and second best XCorr
3	ΔC_n^L	Fractional difference between current and fifth best XCorr
4	Sp	Preliminary score for peptide versus predicted fragment ion values
5	ln(rSp)	The natural logarithm of the rank of the match based on the Sp score
8	Mass	The observed mass $[\mathrm{M+H}]^+$
6	ΔM	The difference in calculated and observed mass
7	abs(ΔM)	The absolute value of the difference in calculated and observed mass
9	ionFrac	The fraction of matched b and y ions
10	ln(NumSp)	The natural logarithm of the number of database peptides within the specified m/z range
11	enzN	Boolean: Is the peptide preceded by an enzymatic (tryptic) site?
12	enzC	Boolean: Does the peptide have an enzymatic (tryptic) C-terminus?
13	enzInt	Number of missed internal enzymatic (tryptic) sites
14	pepLen	The length of the matched peptide, in residues
15–17	charge1–3	Three Boolean features indicating the charge state

Supplementary Table 1: **Features used to represent PSMs.** Each PSM obtained from the search is represented using 17 features. These are the same features used by Percolator, except that three features were removed. These three features—for example, the number of other spectra that match to the same peptide—captured properties of the entire collection of PSMs. We removed them to ensure complete separation between the training set and the test set.

```
Input: labeled proteins $(\mathbf{R}_i, \mathbf{y}_i)$
repeat
  Pick a random protein $(\mathbf{R}_i, \mathbf{y}_i)$
  Compute $F(\mathbf{R}_i)$ given by equation (1).
  if $1 - yF(\mathbf{R}_i) > 0$ then
    Make a gradient step to optimize $L(F(\mathbf{R}_i), \mathbf{y}_i)$
  end if
until convergence
```

Supplementary Algorithm 1: Training Barista



(A) Yeast trypsin

(B) Yeast elastase

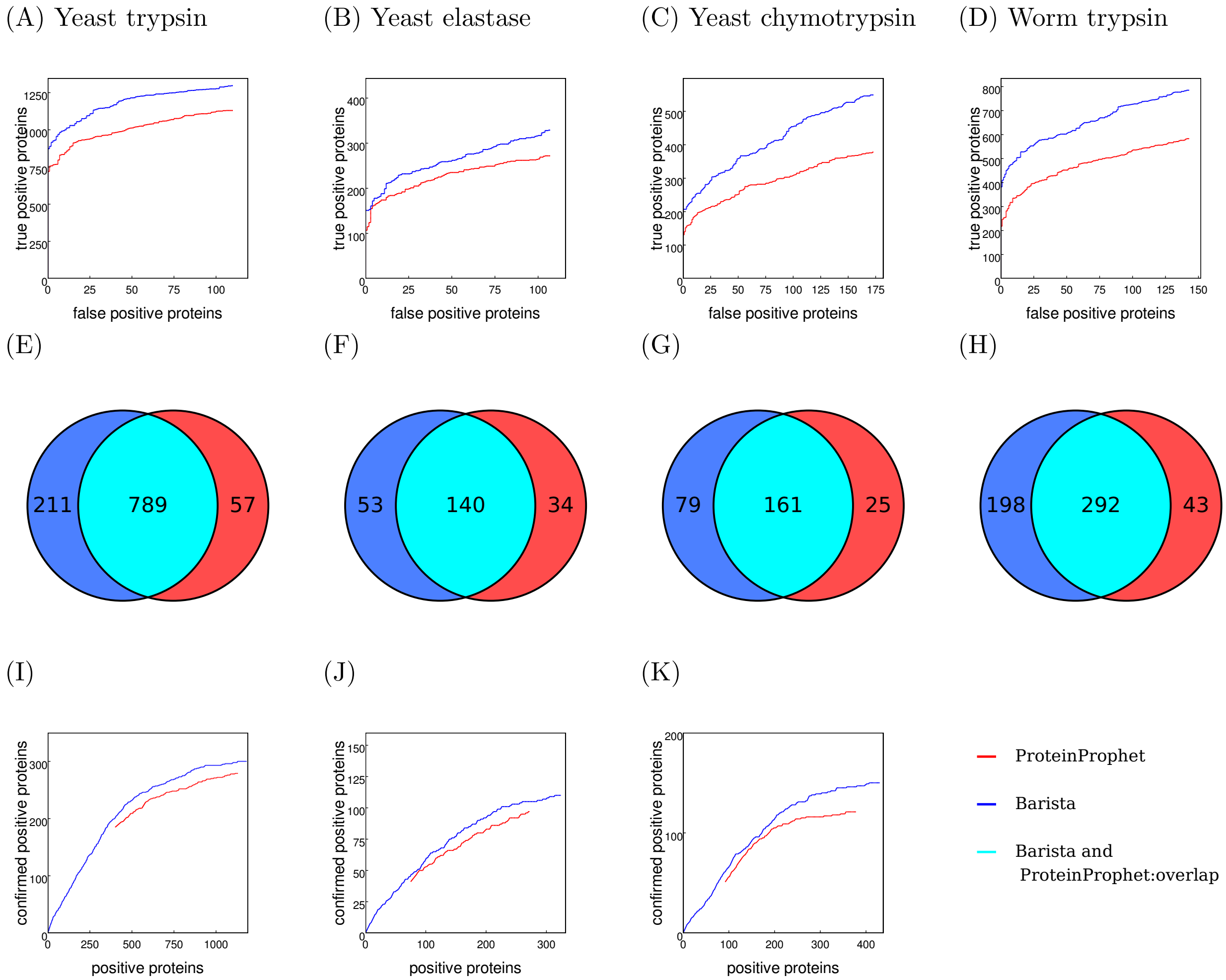
(C) Yeast chymotrypsin

(D) Worm trypsin

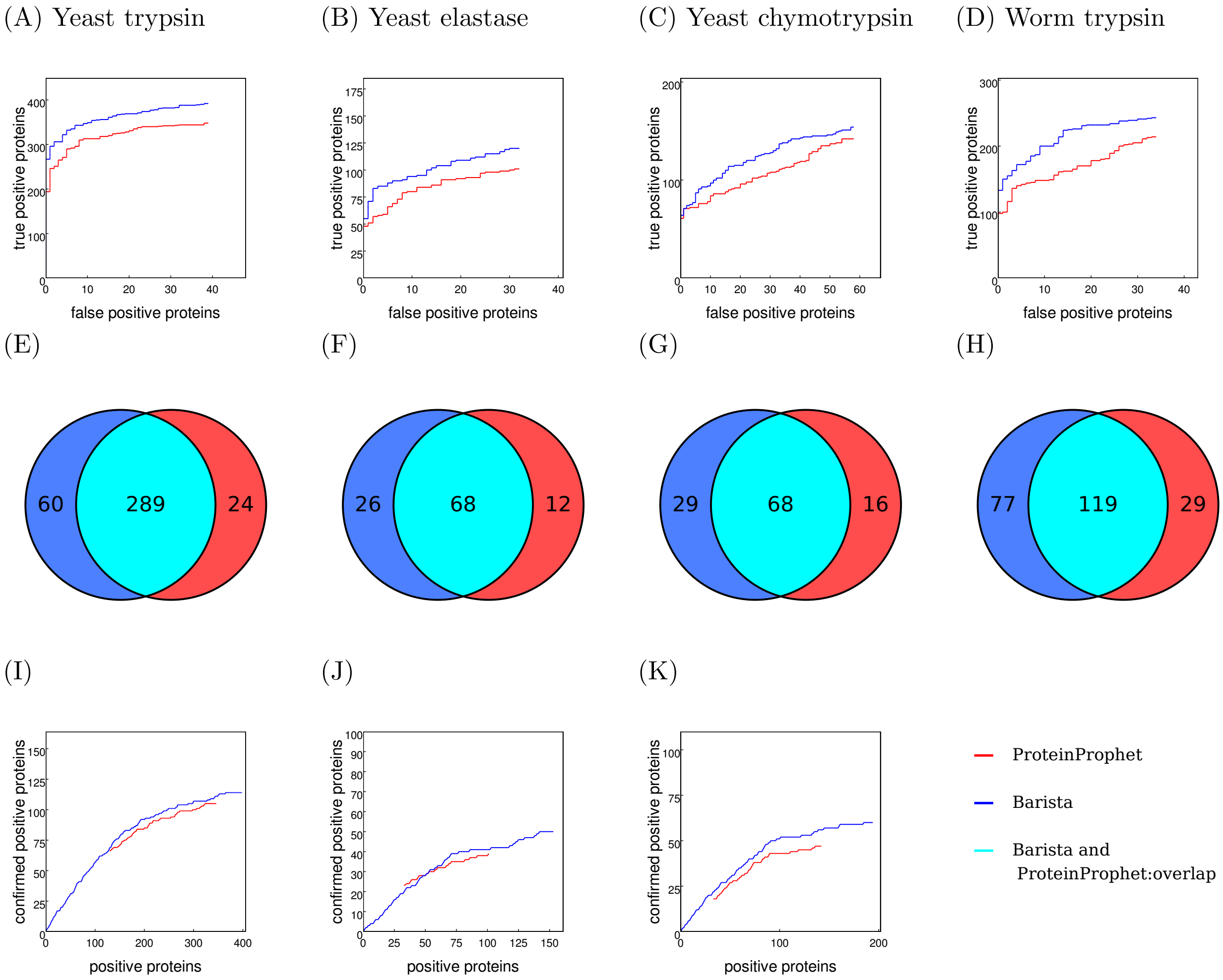
Supplementary Figure 1: **Comparison of Barista and ProteinProphet, using q value thresholds.** This figure is similar to Figure **??**, except that Barista results are reported with respect to a range of q value thresholds plotting on the x-axis, instead of numbers of false positives. The q value is defined as the minimal FDR threshold at which a given score is deemed significant.

Method	PP	Barista	%>PP	IDP	%>IDP
Yeast trypsin	1079	1351	25%	1084	24%
Worm trypsin	271	475	74%	327	45%
Chymotrypsin	289	210	37%	184	57%
Elastase	204	158	29%	144	41%

Supplementary Table 2: **Comparison of protein identification methods at a *q* value threshold of 0.01.** The table lists, for each of the four datasets, the number of proteins identified at $q < 0.01$ by ProteinProphet (PP), Barista and IDPicker (IDP), as well as the improvement provided by Barista relative to the other two methods.



Supplementary Figure 2: **Comparison of Barista (training set) and ProteinProphet.** This figure is similar to Figure **??**, except that Barista results are reported with respect to a training set consisting of approximately 75% of the data.



(A) Yeast trypsin

(B) Yeast elastase

(C) Yeast chymotrypsin

(D) Worm trypsin

(E)

(F)

(G)

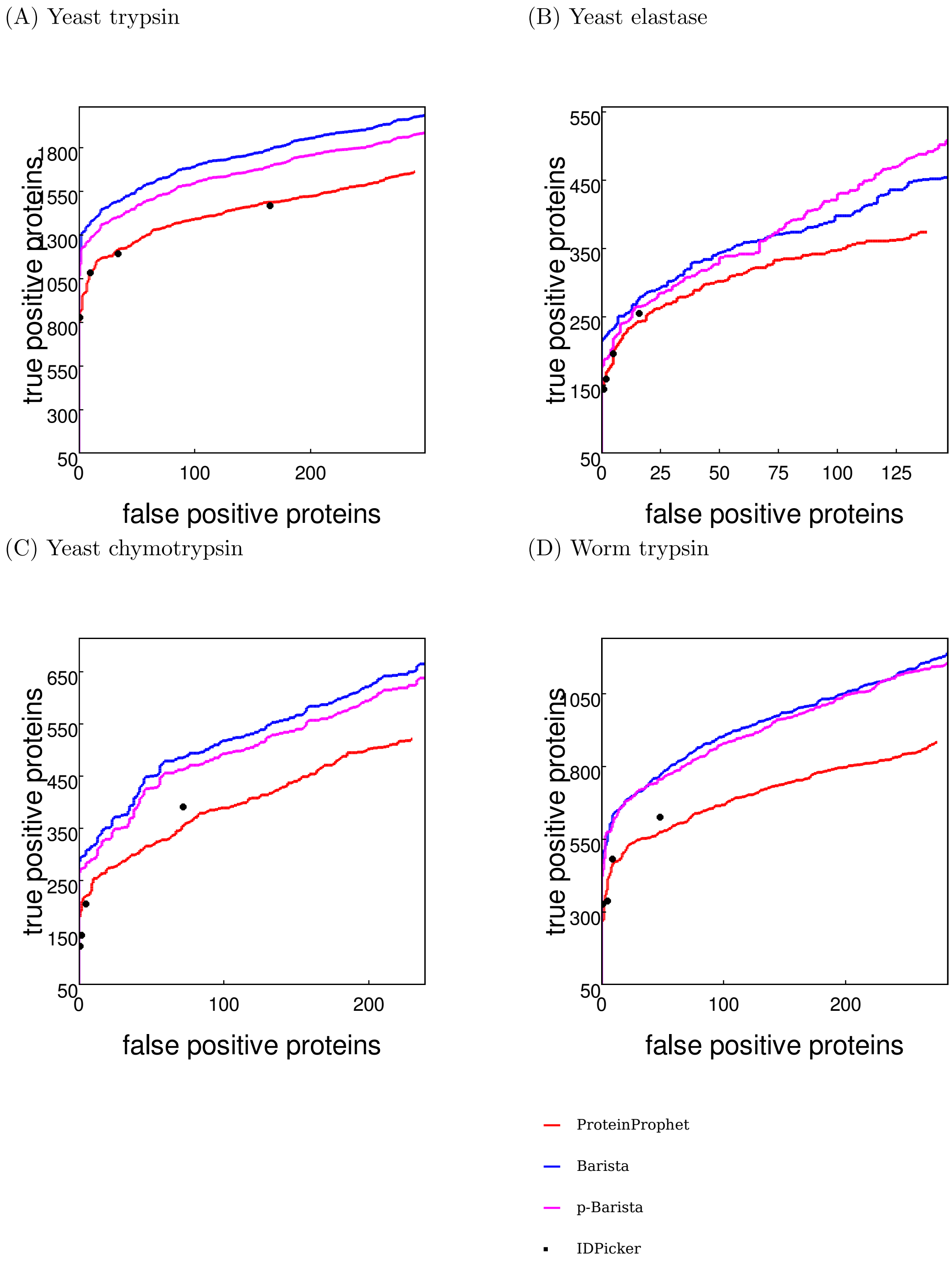
(H)

(I)

(J)

(K)

Supplementary Figure 3: **Comparison of Barista (test set) and ProteinProphet.** This figure is complementary to the previous figure: Barista results are reported with respect to a test set consisting of approximately 25% of the data.

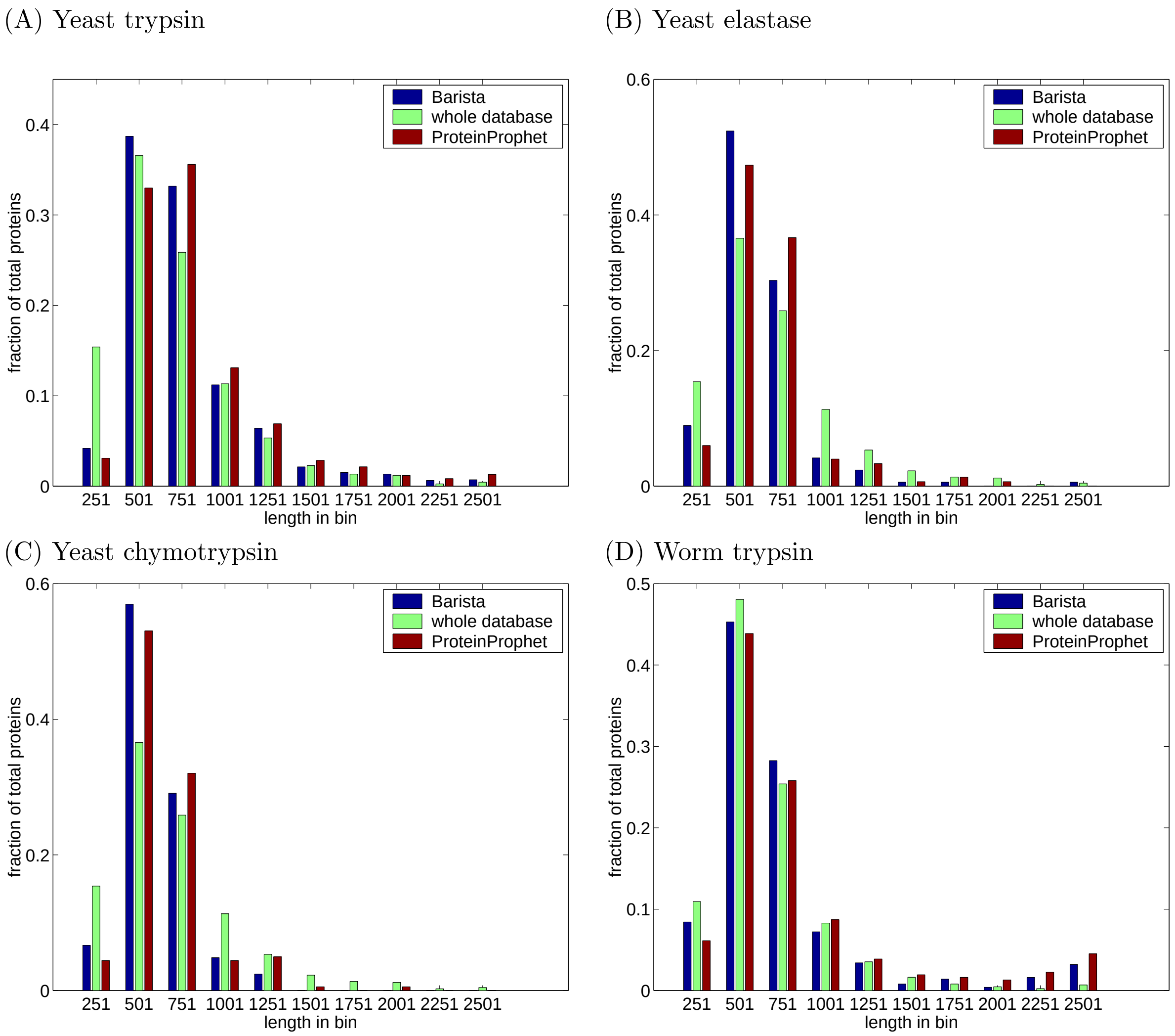

(A) Yeast trypsin

(B) Yeast elastase

(C) Yeast chymotrypsin

(D) Worm trypsin

Supplementary Figure 4: **Comparison of methods, including parsimonious Barista.** This figure is similar to Figure **??**, except that p-Barista is also included.



(A) Yeast trypsin

(B) Yeast elastase

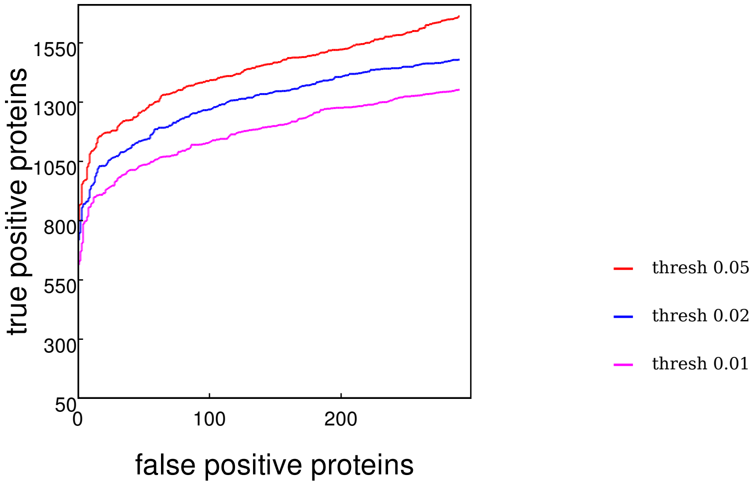
(C) Yeast chymotrypsin

(D) Worm trypsin

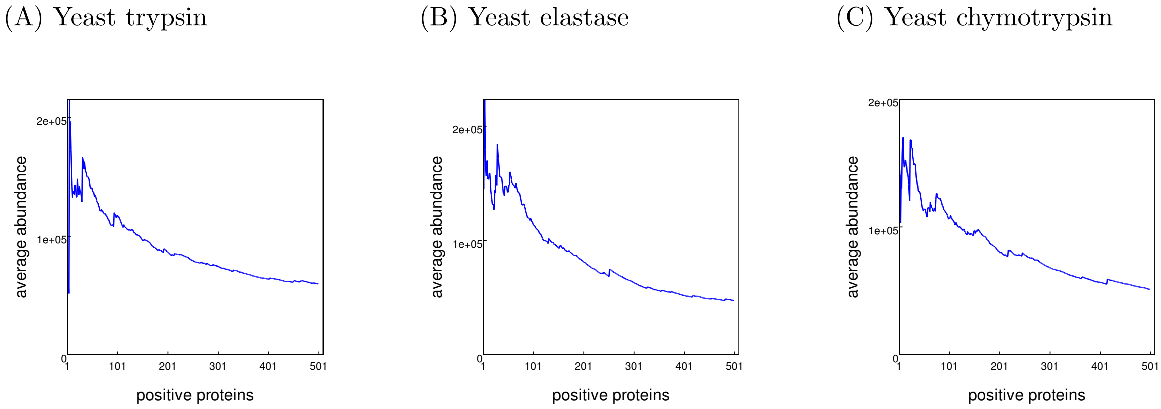
Supplementary Figure 5: **Lengths of proteins identified by Barista and ProteinProphet.** The figure shows histograms of normalized protein counts within different protein length ranges (bins). The protein counts are normalized by the total numbers of proteins in the sample. Proteins are selected using a threshold of 10 false positives.

Data/Method	FP	Barista	ProteinProphet	Overlap	Only Barista	Only ProtProphet
Yeast	0	512	569	555	401	853
Predicted	5	513	574	546	398	924
	10	514	576	544	402	964
	50	523	582	547	431	957
Yeast	0	556	569	574	497	496
Observed	5	552	574	565	496	669
	10	551	576	563	491	690
	50	555	582	568	495	718
Elastase	0	357	389	378	325	505
Predicted	5	356	418	376	320	740
	10	356	441	378	313	762
	50	372	483	378	364	997
Elastase	0	336	389	380	269	495
Observed	5	349	418	378	306	767
	10	355	441	371	327	927
	50	381	483	383	377	983
Chymotrypsin	0	357	389	378	325	505
Predicted	5	356	418	376	320	740
	10	356	441	378	313	762
	50	372	483	378	364	997
Chymotrypsin	0	336	389	380	269	495
Observed	5	349	418	378	306	767
	10	355	441	371	327	927
	50	381	483	383	377	983
Worm	0	565	668	645	468	825
Predicted	5	553	629	613	466	721
	10	530	621	582	446	814
	50	508	652	559	425	1012
Worm	0	680	668	735	595	465
Observed	5	653	629	695	591	460
	10	626	621	657	576	537
	50	575	652	627	483	710

Supplementary Table 3: **Average lengths of identified proteins.** The table reports, for each data set, the average length of the proteins identified at various thresholds. Results for two variants of Barista are reported, using the standard protein score normalization ("predicted") and using normalization based on the number of matched peptides ("observed").



Supplementary Figure 6: **Performance of ProteinProphet as a function of threshold.** This figure is similar to Figure **??**, using the "yeast trypsin" dataset, except that ProteinProphet was run with PeptideProphet thresholds of 0.01, 0.02 and 0.05.

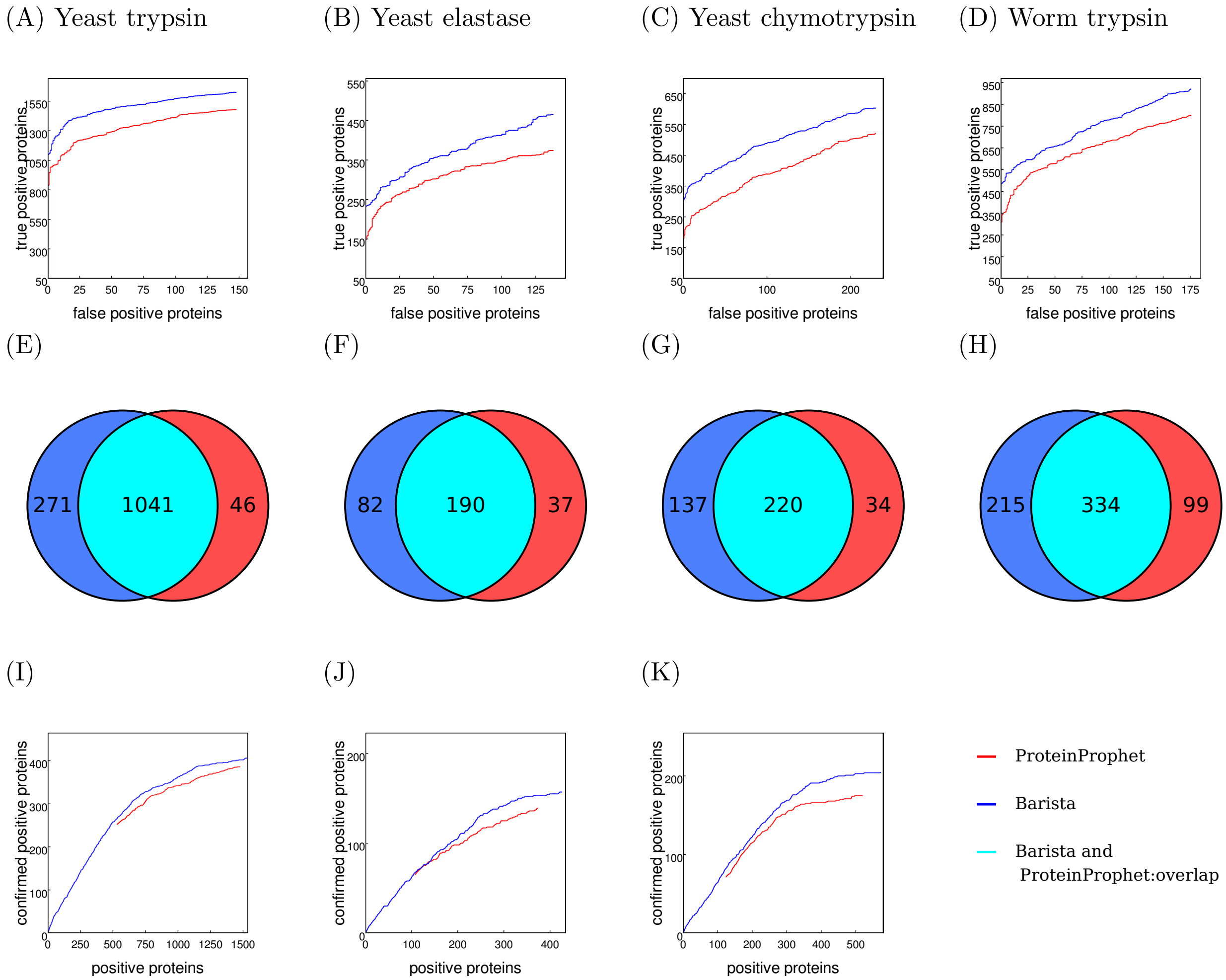


(A) Yeast trypsin

(B) Yeast elastase

(C) Yeast chymotrypsin

Supplementary Figure 7: **Abundances of proteins identified by Barista.** The figure plots average protein abundance of the top n proteins, as a function of n. Protein abundances are taken from [Ghaemmaghami et al., 2003].


(A) Yeast trypsin

(B) Yeast elastase

(C) Yeast chymotrypsin

(D) Worm trypsin

(E)

(F)

(G)

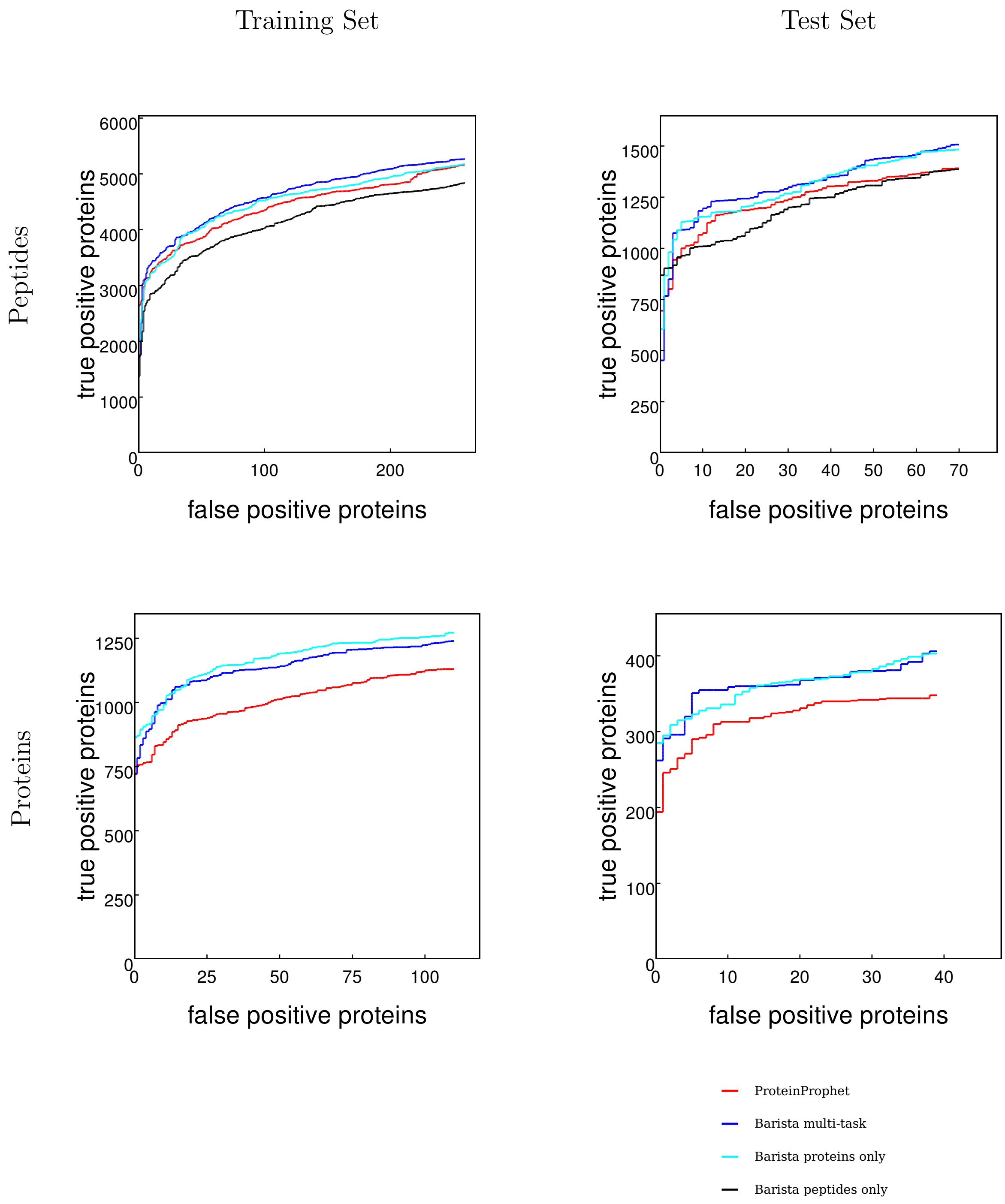
(H)

(I)

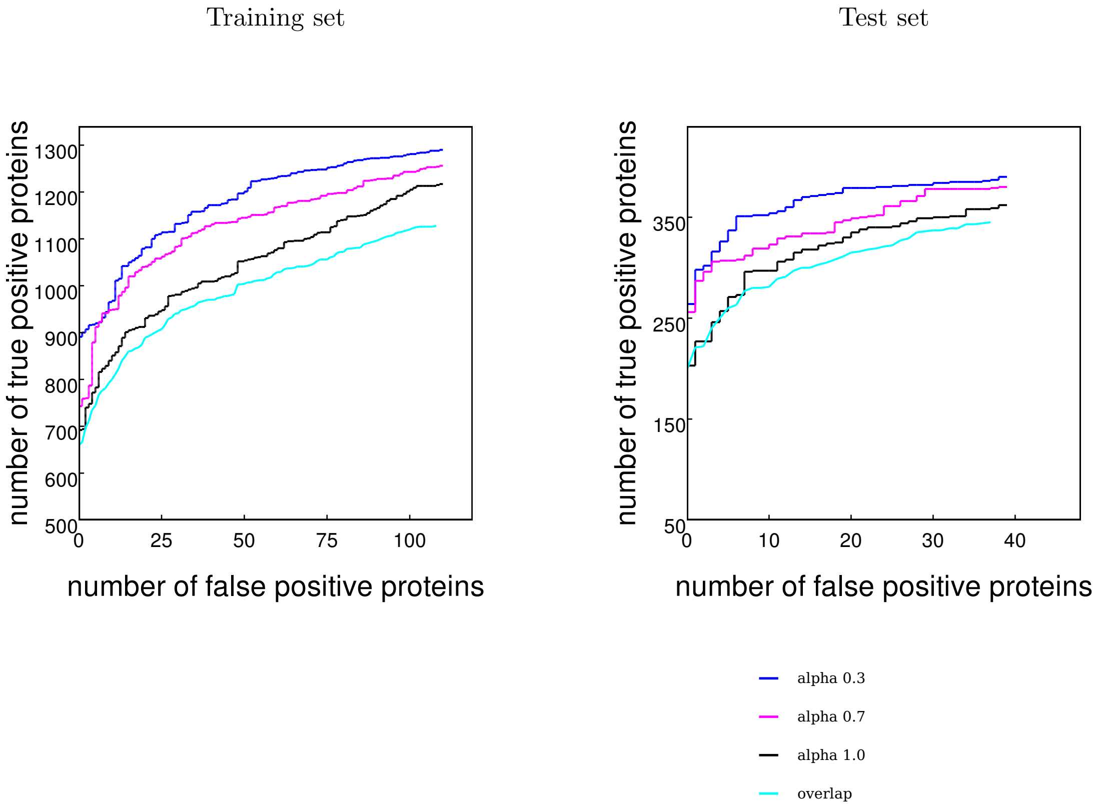
(J)

(K)

Supplementary Figure 8: **Comparison of Barista (with modified protein normalization) and ProteinProphet.** This figure is similar to Figure **??**, except that the number of matched ("observed") peptides was used as the normalization factor in the protein scoring function.



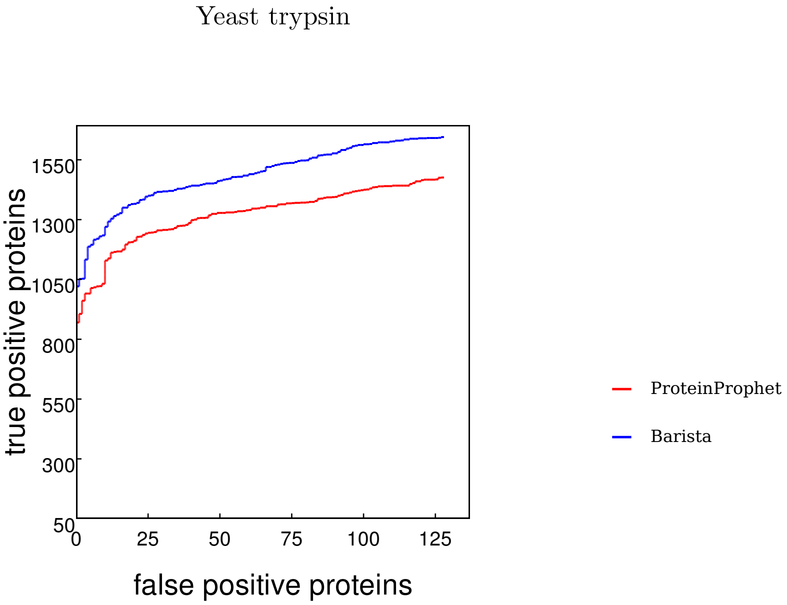
Supplementary Figure 9: **Multi-task optimization of protein and peptide ranking.** The figure shows, for the tryptic yeast data set, the performance of ProteinProphet, PeptideProphet and three variants of Barista. Each panel plots the number of distinct peptides (top) or proteins (bottom) as a function of the number of false positives. ProteinProphet is evaluated at the protein level and PeptideProphet at the peptide level. Barista is trained on the protein ranking task, on the peptide ranking task, or both.



Supplementary Figure 10: **Hyperparameter selection.** Barista uses a hyperparameter α when normalizing for the number N of peptides per protein. The figure shows, for the "yeast trypsin" data set, the performance on the training set and test set for different choices of α. Based on this analysis, we used a fixed value of $\alpha = 0.3$ for all subsequent experiments. The plot also shows ("overlap") the size of the protein set that was identified by all three runs.

	trypsin	elastase	chymotrypsin
Barista true positives	1256	259	318
ProteinProphet true positives	1087	227	254
overlap	992	176	212
Barista only	264	84	107
ProteinProphet only	95	51	42
Barista-only confirmed	18%	32%	49%
ProteinProphet-only confirmed	10%	21%	37%

Supplementary Table 4: **Comparison of protein sets identified by Barista and ProteinProphet.** The table describes the overlap between proteins identified by the two methods, and provides the percentage of proteins identified by a single method that are confirmed by the external gold standard. All of the measurements were done at 10 false positives.



Supplementary Figure 11: **Barista results using reversed decoys** This figure is similar to Figure **??**A, except that the decoys were generated by reversing the proteins in target database, instead of shuffling each protein.

References

M. Bern and D. Goldberg. Improved ranking functions for protein and modification-site identifications. *Journal of Computational Biology*, 15:705–719, 2008.

R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

S. Ghaemmaghami, W. K. Huh, K. Bower, R. W. Howson, A. Belle, N. Dephoure, E. K. O'Shea, and J. S. Weissman. Global analysis of protein exression in yeast. *Nature*, 425:737–741, 2003.

F. C. P. Holstege, E. G. Gennings, J. J. Wyrick, T. I. Lee, C. J. Hengartner, M. R. Green, T. R. Golub, E. S. Lander, and R. A. Young. Dissecting the regulatory circuitry of eukaryotic genome. *Cell*, 95:717–728, 1998.

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.

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.

A. I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, 75:4646–4658, 2003.

M. Spivak, J. Weston, L. Bottou, L. Käll, and W. S. Noble. Improvements to the percolator algorithm for peptide identification from shotgun proteomics data sets. *Journal of Proteome Research*, 8(7):3737–3745, 2009.