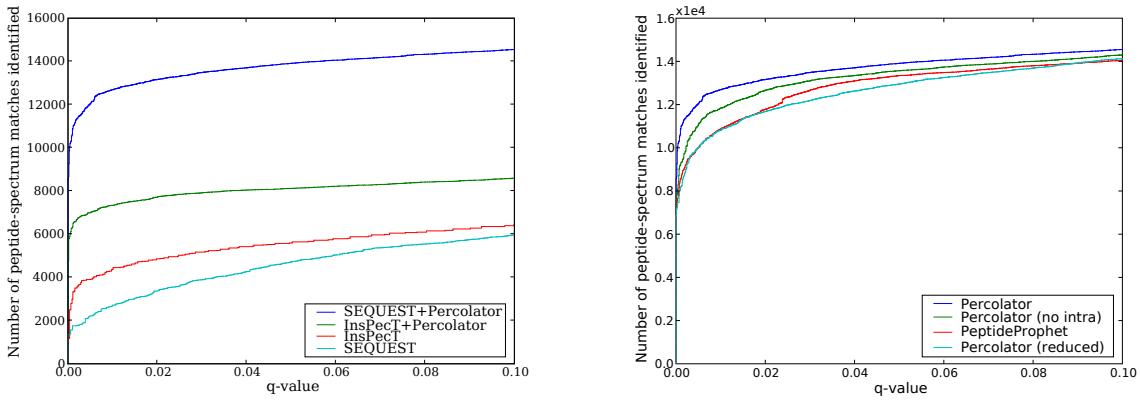
Semi-supervised learning for peptide identification from shotgun proteomics datasets

Lukas Käll, Jesse D Canterbury, Jason Weston, William Stafford Noble & Michael J MacCoss

Supplementary Figures and Text

Supplementary Figure 1 Percolator comparisons.
Supplementary Figure 2 Variation of the Percolator scoring function between data sets.
Supplementary Figure 3 Percolator robustness.
Supplementary Figure 4 Interpretation of a single tandem mass spectrum acquired from two unique peptides.
Supplementary Table 1 Features used to represent peptide spectrum matches.
Supplementary Table 2 Feature analysis.
Supplementary Methods
Supplementary Data Additional experiments.

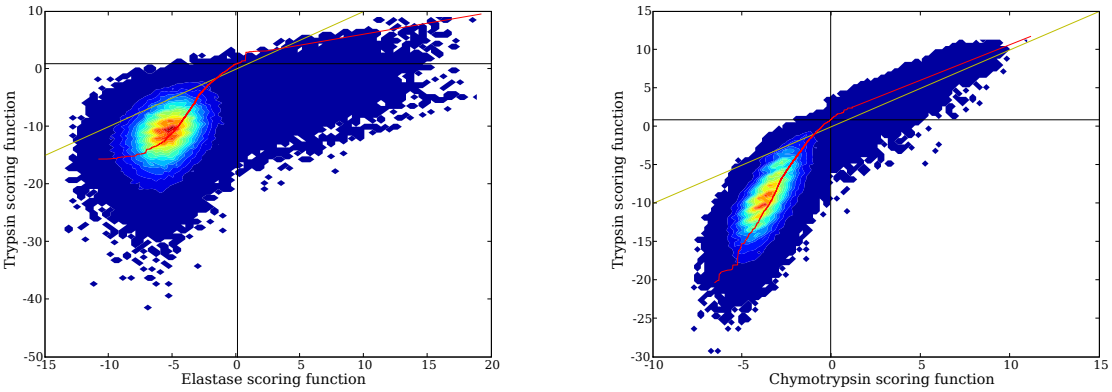
Supplementary Figure 1: Percolator Comparisons



A B

Supplementary Figure 1: **Percolator Comparisons (A) Coupling Percolator with SEQUEST and InsPecT** The figure plots the number of identified PSMs as a function of the false discovery rate on a yeast data set containing 69,705 PSMs. The four series correspond to SEQUEST and InsPecT, with and without post-processing by Percolator. **(B) Performance dependence of features on the tryptic yeast data set.** Percolator's improvement relative to PeptideProphet could be attributable to two differences between the algorithms: (1) Percolator's ability to dynamically fit its model to the given data set, and (2) Percolator's use of a larger feature set. To isolate these differences, we trained a version of Percolator on a reduced set of features, corresponding to the features used in PeptideProphet. We used the yeast data treated with elastase and chymotrypsin as training data. We have plotted the performance of Percolator (blue curve), PeptideProphet 3.0 (red curve), a version of Percolator without the intra-set (protein level) features (green curve), and a version of Percolator trained only on the features used by Peptide Prophet (cyan curve). We see a performance increase over PeptideProphet, whose linear discriminator is trained once for a fixed tryptic data set. However, the version of Percolator that uses a reduced feature set does perform significantly worse than Percolator using the entire collection of features.

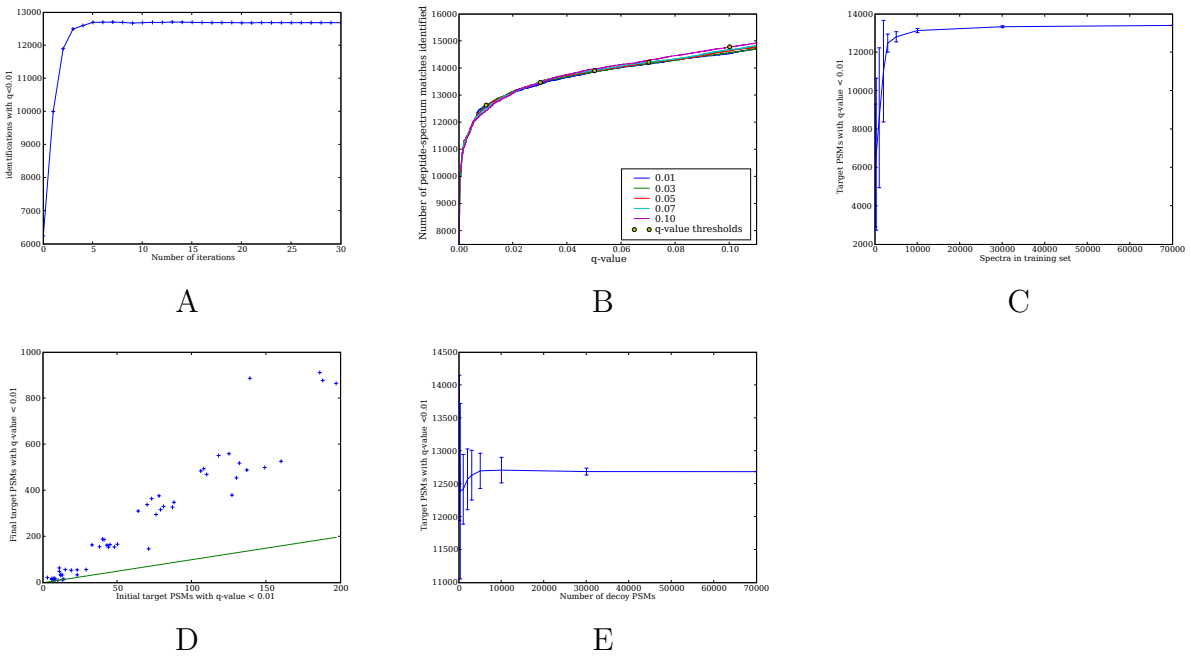
Supplementary Figure 2: Variation of the Percolator scoring function between data sets



(A) Elastase data set (B) Chymotrypsin data set

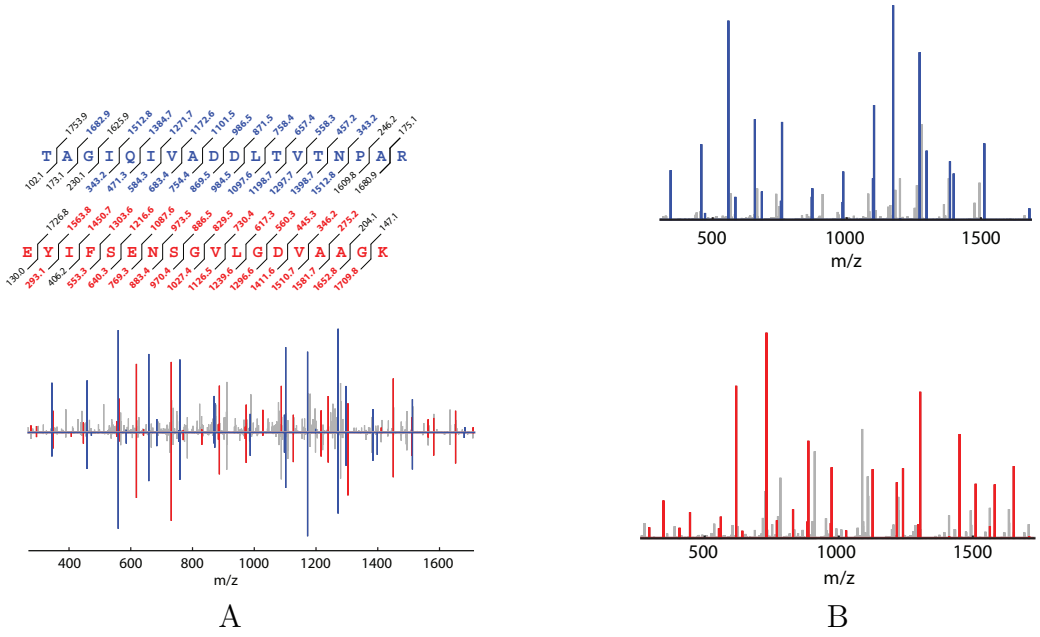
Supplementary Figure 2: **Variation of the Percolator scoring function between data sets.** Superficially, the results in Supplementary Figure 1B suggest that Percolator's performance is largely due to its increased feature set. However, it is important to recognize that using a large feature set only makes sense in the context of a method that adapts to the given data set. As we increase the number of features, the learned classifier begins to fit the training set more closely. The resulting classifier may perform well on the training data but is unlikely to generalize to data that does not resemble the training set. To illustrate this effect, we trained Percolator (using all 20 features) on the tryptic yeast data set and tested its performance on the yeast data treated with elastase and chymotrypsin. For these data sets, we calculated the enzyme specifity features of each PSM (enzN, enzC and enzInt) using the correct cleavage rules for the given enzyme, but we use the SVM feature weights that were learned from the tryptic data set. In the figure, each panel plots the SVM discriminant scores assigned by Percolator to target PSMs in yeast data sets cleaved by (A) elastase (B) chymotrypsin. On the y-axis we use the scoring function from Percolator trained on yeast proteins cleaved by trypsin and on the y-axis we use the scoring function of Percolator trained on the data set at hand. In both plots the score threshold corresponding to a q value of 0.01 is indicated with a black line. Note that, in each case, the enzymatic termini (features "enzN," "enzC" and "enzInt") are computed correctly for the given data set; for the y-axis values, only the SVM weights are taken from the tryptic data set. Yellows lines indicate $y = x$ and red lines indicate equal q values. The results show that the classifier learns to recognize specific characteristics of spectra from tryptic digests, and consequently does not perform as well on other kinds of data. Especially for elastase data, we see a dramatic performance decrease relative to the version of Percolator that has been trained and tested on the elastase data. This experiment shows that we cannot train a static classifier using a large set of features and hope for the classifier to generalize to new types of spectra.

Supplementary Figure 3: Percolator Robustness



Supplementary Figure 3: **Percolator Robustness (A) Convergence behavior of Percolator.** The figure plots, for the initial yeast data set, the number of peptides identified at a 1% FDR threshold as a function of the number of iterations. **(B) Performance with various FDR thresholds.** The figure plots the number of identified peptides as a function of the FDR threshold. Each series was generated using Percolator with a different FDR threshold. **(C) Learning curve.** The figure plots the average number of PSMs identified at a 1% FDR threshold in the entire data as a function of the number of normal and shuffled PSMs used to generate the training set. Numbers are calculated as an average over 10 experiments. The bars represent the standard error. **(D) Influence of the number of good target PSMs of the performance.** The figure plots the number of PSMs identified at a 1% FDR threshold before and after processing with Percolator. Regardless of the number of target PSMs over threshold Percolator never identified less than the initial number of PSMs. The train and test sets were derived by sub sampling the larger yeast data set treated with trypsin. The green line indicates the level where the number of PSMs identified before and after processing with Percolator is the same. **(E) Influence of the size of the negative test set.** The figure plots the average number of peptides identified at a 1% FDR threshold in the entire data as a function of the number of shuffled PSMs in the test set. Numbers are calculated as an average over 10 experiments. The bars represent the standard error.

Supplementary Figure 4: Interpretation of a single tandem mass spectrum acquired from two unique peptides



A B

Supplementary Figure 4: **Interpretation of a single tandem mass spectrum acquired from the isolation and activation of two unique peptide species using a combination of SEQUEST and Percolator.** A complicating factor for the analysis of complex proteomics mixtures is the interpretation of MS/MS spectra that contain multiple peptide species isolated and activated simultaneously during the precursor ion selection [10, 1]. Some preliminary experiments indicate that as many as 12% of the total MS/MS spectra acquired may be composed of two or more molecular species [**?**]. Optionally, Percolator can treat each of the top five sequences returned by the database searching algorithm independently, calculating a q-value for each. Thus, if a threshold is set to return peptides with q-values less than 0.01 and multiple peptide sequences meet this criterion, then multiple peptide sequences will be assigned to a single spectrum. In the yeast data set, when re-ranking SEQUEST's top five PSMs, Percolator identifies 13,657 PSMs, corresponding to 12,798 spectra. In total, Percolator assigns more than one peptide to 731 spectra. However, many of these apparent double identifications are indistinguishable, *e.g.*, substitutions of leucine for isoleucine. Filtering out these examples, we are left with 399 spectra that are assigned to two distinct peptides, and no spectra assigned to three or more peptides. Figure 4 shows an example of a single spectrum assigned to two different peptide sequences. The blue and red fragment ions in the spectrum indicate the predicted b- and y-ions arising from the peptide sequences TAGIQIVADDLTVTNPAR (Percolator q-value = 0.0, XCorr=3.86) and EYIFSENSGVLGDVAAGK (Percolator q-value = $3.4 \cdot 10^{-4}$, XCorr=4.15), respectively. **(A)** Percolator interpreted the SEQUEST results and assigned two peptide sequences to the single spectrum. The fragment ions from the peptides TAGIQIVADDLTVTNPAR and EYIFSENSGVLGDVAAGK are displayed in blue and red, respectively. The mirrored spectrum is the combination of two spectra acquired from the two respective synthetic peptides. **(B)** This panel shows the two synthetic peptides that are superimposed in panel (A).

Supplementary Table 1: Features used to represent PSMs

Supplementary Table 1: **Features used to represent PSMs.**

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 $[M+H]^+$
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
18	ln(numPep)	Number of PSMs for which this is the best scoring peptide.
19	ln(numProt)	Number of times the matched protein matches other PSMs.
20	ln(pepSite)	Number of different peptides that match this protein.

Note: The first ten features are computed by SEQUEST. For numProt, if more than one protein matches the spectrum, then the protein most frequently matching other spectra is selected. For pepSite, if more than one protein matches the spectrum, then the protein with the most such peptide sites is selected. Supplementary Algorithm 2 gives a detailed description of how features 18-20 are calculated. Percolator standardizes each feature to have a mean of zero and a variance of one across the entire collection of target and decoy PSMs.

Supplementary Table 2: Feature analysis

Supplementary Table 2: **(Top) Feature weights for scoring functions trained on different yeast data sets. (Bottom) Importance of different features types.**

Feature	Trypsin	Elastase	Chymotrypsin
ln(rSp)	-0.179	-0.193	-0.121
ΔC_n^L	0.734	0.87	0.64
ΔC_n	6.23	10.6	6.91
Xcorr	0.861	0.975	0.637
Sp	-3.54e-05	0.00111	-0.000136
IonFrac	0.182	0.863	1.33
Mass	0.00184	0.00171	-0.000148
PepLen	-0.139	-0.187	0.0467
Charge1	0.309	2.31	0.703
Charge2	0.169	0.3	-0.0127
Charge3	-0.182	-0.373	-0.00364
enzN	3.1	0.744	1.1
enzC	4.2	0.272	0.913
enzInt	-1.47	-0.00287	-0.149
ln(numSp)	0.313	3.99	0.068
ΔM	0.126	-0.0318	0.0595
abs(ΔM)	-0.36	-0.346	-0.223
ln(numPep)	-0.26	-0.593	-0.446
ln(numProt)	3.36	4.32	2.49
ln(pepSite)	-2.92	-2.36	-1.79

Removed Features	Number of positives	Drop in performance
None	12,691	-
Charge1, Charge2, Charge3	12,671	0.2%
enzN, enzC, enzInt	8,466	33%
ln(numPep), ln(numProt), ln(pepSite)	11,820	6.9%
Xcorr, ΔC_n, ΔC_n^L, Sp, ln(rSp)	11,231	12%
All but PeptideProphet's features	10,716	16%

In the "All but PeptideProphet's features" experiment, we used only Xcorr, ΔC_n, ln(rSp), abs(ΔM), pepLen, Charge1-3, enzN and enzC.

1 Supplementary Methods

1.1 The Percolator algorithm

Percolator's goal is to rank a collection of candidate PSMs to maximize the number of peptides identified at a target false discovery rate. Our method, which we call Percolator, proceeds in three phases (see Algorithm 1). Initially, we run an existing peptide identification algorithm on the spectra twice, using one unshuffled and one shuffled sequence database. While we have chosen to demonstrate Percolator using a decoy derived from shuffled sequences, our software can use any type of decoy, including decoys generated from a reversed database. For each spectrum, we store the top-scoring PSM against each database. We refer to these as target and decoy PSMs, respectively. For each target and decoy PSM, we compute a vector of 20 features, summarized in Table 1. These features remain fixed for the duration of the algorithm. We randomly divide the set of decoy PSMs in half, using one half in phase two, and the remainder in phase three. At the end of the algorithm, a subset of the target PSMs will be identified as correct.

The second phase is iterative, and each iteration consists of three steps: (1) selecting a subset of high-confidence target PSMs to serve as a positive training set, (2) training an SVM to discriminate between the positive and the decoy PSMs, and (3) re-ranking the entire set of PSMs using the trained classifier. To select the positive PSMs, we rank the target and decoy PSMs by the SEQUEST XCorr, and we set a threshold to achieve a user-specified target q value. The target PSMs above the threshold comprise the positive training set, and all of the decoy PSMs comprise the negative training set. We then train a linear SVM to discriminate between positive and negative PSMs, using a modified finite Newton l_2-SVM solver [2, 5]. This training is very fast: training the classifier on 70,000 PSMs takes approximately 2 s on an Athlon MP Opteron 842 CPU. In subsequent iterations, the ranking is produced by our discriminative classifier, rather than by XCorr. The algorithm terminates after a fixed number of iterations. Empirical evidence (Supplementary Figure 3) suggests that ten iterations is sufficient to achieve a stable set of PSMs, and that the algorithm performs very similarly, regardless of the user-specified q value threshold.

In the third phase, we apply the final SVM to the entire set of target PSMs, as well the second set of decoy PSMs. The resulting ranked list gives us an unbiased estimate of the q value for each target PSM [6], *i.e.*, of the minimal false discovery rate threshold required to form a set of positive identifications which includes the PSM.

Percolator is implemented in C++, using SVM optimization code from SVMlin [2]. The software, including source code, can be downloaded from `http://noble.gs.washington.edu/proj/percolator`.

1.1.1 Spectra containing multiple peptides

Another complicating factor for the analysis of complex proteomics mixtures is the interpretation of MS/MS spectra that contain multiple peptide species isolated and activated simultaneously during the precursor ion selection [10, 1]. Some preliminary experiments indicate that as many as 12% of the total MS/MS spectra acquired may be composed of two or more molecular species [**?**]. Optionally, Percolator can treat each of the top five sequences

returned by the database searching algorithm independently, calculating a q value for each. Thus, if a threshold is set to return peptides with q values less than 0.01 and multiple peptide sequences meet this criterion, then multiple peptide sequences will be assigned to a single spectrum. In the yeast data set, when re-ranking SEQUEST's top five PSMs, Percolator identifies 13,657 PSMs, corresponding to 12,798 spectra. In total, Percolator assigns more than one peptide to 731 spectra. However, many of these apparent double identifications are indistinguishable, *e.g.*, substitutions of leucine for isoleucine. Filtering out these examples, we are left with 399 spectra that are assigned to two distinct peptides, and no spectra assigned to three or more peptides. An example of a single spectrum assigned to two different peptide sequences is shown in the supplement.

1.1.2 Estimation of q values

Denote the scores of target PSMs (*i.e.*, matches between the spectra and peptides from the unshuffled database) $f_1, f_2, \ldots, f_{m_f}$ and the scores of decoy PSMs $d_1, d_2, \ldots, d_{m_d}$. For a given score threshold t, the number of positives is $P(t) = |\{f_i > t; i = 1, \ldots, m_f\}|$. The estimated number of false positives among the positives is given by $E(FP(t)) = \pi_0 \frac{m_f}{m_d} |\{d_i > t; i = 1, \ldots, m_d\}|$, where π_0 is the estimated proportion of target PSMs that are incorrect. We can then estimate the FDR at a given threshold t as

$$E\{FDR(t)\} = \frac{\pi_0 \frac{m_f}{m_d} |\{d_i > t; i = 1, \ldots, m_d\}|}{|\{f_i > t; i = 1, \ldots, m_f\}|} \tag{1}$$

In this work, we conservatively set $\pi_0 = 0.9$, except when re-ranking false negatives. In this case, because we have five times as many PSMs but roughly the same number of correct PSMs, we set $\pi_0 = 0.98$. Once the FDR levels are established, the q value associated with a given PSM with score t can be calculated as $q(t) = \min_{t' \leq t} E\{FDR(t')\}$. Throughout the text we calculate q values at the PSM level; *i.e.*, the same peptide can be reported as a target or decoy identification multiple times.

1.1.3 SVM training

The SVM algorithm has a single, user-specified regularization parameter C, which controls the magnitude of the penalty assigned to misclassified examples. We expect that our positive and negative sets of PSMs will contain different numbers of errors; therefore, we charge different penalties, C^+ and C^-, for misclassification of positive and negative PSMs. The values of these hyperparameters are selected via internal three-fold cross-validation within the training set. At each iteration, we search a three-by-three grid of values $C^+ \in \{0.1, 1, 10\}$ and the fraction $C^-/C^+ \in \{1, 3, 10\}$, selecting the pair of values that yield the largest number of positive identifications at the user-specified FDR threshold t. We then retrain the SVM on the entire training set using the selected values of C^+ and C^-.

1.1.4 Re-ranking procedure

For each spectrum in our set, the re-ranking procedure reads in the five PSMs with highest XCorr against the target database as well as the five PSMs with highest XCorr against the

shuffled database. ΔC_n is calculated as the fractional difference between the XCorr of the current PSM and the XCorr of the second ranked PSM. This calculation results in ΔC_n values of zero for second ranked PSMs and negative values for lower ranked PSMs. One additional Boolean feature is introduced for the re-ranking procedure, indicating whether this PSM was ranked first by SEQUEST. In the subsequent processing, we use the Percolator algorithm exactly as described above.

Algorithm 1 The percolator algorithm. The input variables are defined as follows: $S =$ a set of spectra; $D =$ a peptide database; $t =$ the desired FDR threshold; $I =$ the number of iterations. SEQUEST returns, for a given set of spectra, a corresponding set of top-ranked peptides and the respective scores.

procedure PERCOLATOR(S, D, t, I)
  $(P_r, X_r) \leftarrow$ SEQUEST(S, D) ▷ Compute target PSMs.
  $(P_d, X_d) \leftarrow$ SEQUEST(S, shuffle(D)) ▷ Compute decoy PSMs.
  $F_r \leftarrow$ computeFeatures(S, P_r) ▷ Compute the corresponding feature vectors.
  $F_d \leftarrow$ computeFeatures(S, P_d)
  for $i \leftarrow 1 \ldots I$ **do**
    $F_r^+ \leftarrow$ selectByFDR(t, F_r, X_r, F_d, X_d) ▷ Select the positive PSMs.
    $W \leftarrow$ trainSVM(F_r^+, F_d) ▷ Train the classifier.
    $X_r \leftarrow classify(W, F_r)$ ▷ Re-rank the PSMs.
    $X_d \leftarrow classify(W, F_d)$
  end for
  $(P_d, X_d) \leftarrow$ SEQUEST(S, shuffle(D)) ▷ Compute new decoy PSMs and features.
  $F_d \leftarrow$ computeFeatures(S, P_d)
  return (selectByFDR(t, F_r, X_r, F_d, X_d)
end procedure

1.2 Alternative peptide identification methods

The Washburn *et al.* [8] criteria were as follows: charge 1 PSMs with XCorr ≥ 1.9 and two tryptic termini; charge 2 PSMs with XCorr ≥ 2.2 and at least one tryptic terminus or XCorr ≥ 3; and charge 3 PSMs with XCorr ≥ 3.75 and at least one tryptic terminus. Furthermore, all PSMs were required to have $\Delta C_n \geq 0.1$ and were allowed any number of internal missed cleavage sites. The DTASelect default thresholds were XCorr ≥ 1.8 for charge 1 spectra, XCorr ≥ 2.5 for charge 2 spectra, and XCorr ≥ 3.5 for charge 3. For all charges, $\Delta C_n \geq 0.08$ was required. We ran PeptideProphet 3.0 with default parameters, except for the elastase and chymotrypsin data, where we used the appropriate enzyme-specificity options.

When processing InsPecT results, Percolator used the values MQScore, TotalPRMScore, MedianPRMScore, FractionY, FractionB, Intensity, p-value, F-Score, DeltaScore, DeltaScoreOther as as defined in the online documentation of InsPecT. In addition, we calculated the features pepLen, charge1-3, enzN, enzC, enzInt, numProt, numPep, pepSite according to the definitions in Table 1. The target database and the two decoy databases were searched with InsPecT version 20070613 in three separate runs, with no protease specificity. The p-values and F-Scores were calculated in a post processing step combining the three results files.

Algorithm 2 The algorithm for calculating intra set features. The input variable P is a set of PSMs. The function findProteinMatches($Peptide$) returns the set of proteins containing the peptide $Peptide$.

```
 1: procedure IntraSetFeatures(P)
 2:     for (Peptide, Spectrum) ∈ P do                          ▷ Initialize variables.
 3:         numberPeptides[Peptide] ← 0
 4:         for Protein ∈ findProteinMatches(Peptide) do
 5:             numberProteins[Protein] ← 0
 6:             uniqPep[Protein] ← {}
 7:         end for
 8:     end for
 9:     for (Peptide, Spectrum) ∈ P do                          ▷ Set up frequency hashes.
10:         numberPeptides[Peptide] ← numberPeptides[Peptide] + 1
11:         for Protein ∈ findProteinMatches(Peptide) do
12:             numberProteins[Protein] ← numberProteins[Protein] + 1
13:             if Peptide ∉ uniqPep[Protein] then
14:                 uniqPep[Protein] ← uniqPep[Protein] ∪ {Peptide}
15:             end if
16:         end for
17:     end for
18:     I ← {}                                                  ▷ Initialize the set of return values.
19:     for (Peptide, Spectrum) ∈ P do
20:         numPep ← numberPeptides[Peptide]
21:         numProt ← 0
22:         pepSite ← 0
23:         for Protein ∈ findProteinMatches(Peptide) do
24:             numProt ← max(numProt, numberProteins[Protein])
25:             pepSite ← max(pepSite, |uniqPep[Protein]|)
26:         end for
27:         I ← I ∪ {(Peptide, Spectrum, numPep, numProt, pepSite)}
28:     end for
29:     return (I)
30: end procedure
```

1.3 Sample Preparation

Yeast (*Saccharomyces cerevisiae* strain S288C) was cultured in YPD media and grown to mid log phase at 30 °C. Cells were lysed and the membrane vesicles enriched by ultracentrifugation. The resulting pellet was solubilized in 0.1% RapiGest (in 50 mM NH_4HCO_3, pH 7.8) using several pulses from an immersion sonicator. Protein disulfide bonds were reduced by incubation with 5 mM dithiolthreitol for 30 min at 60 °C. After cooling to room temperature, the protein free thiols were alkylated with the addition of iodoacetamide to a final concentration of 75 mM for 30 min at room temperature in the dark. Reduced and alkylated proteins were digested by adding modified trypsin (Promega) at a 1:50 enzyme:substrate ratio and incubating at 37 °C for 4 hours with constant mixing. The above digestion protocol was repeated two additional times where the enzymes elastase and chymotrypsin (Roche) were substituted in the described procedure for trypsin. After digestion, the proteolysis was quenched and the RapiGest hydrolyzed by adding HCl to a final concentration of 200 mM and incubating at 37 °C for 45 minutes. The samples were centrifuged at 14,000 RPM using a microcentrifuge to remove any insoluble material and the supernatant stored at -80 °C until analysis by μLC-MS/MS using as described below.

C. elegans (Bristol N2 strain) were cultured at 20 °C on agarose plates containing E. coli (strain OP50) using standard techniques. Mixed stage worms were washed off the plates with M9 buffer and sucrose floated to remove bacterial contaminants. Worms were then pelleted, washed, resuspended in lysis buffer (310 mM NaF, 3.45 mM NaVO3, 50 mM Tris, 12 mM EDTA, 250 mM NaCl, 140 mM dibasic sodium phosphate pH 7.6), and lysed using immersion sonication. Cell debris and unbroken cells were removed by a low speed spin at 2,000 RPM. The supernatant from the low speed spin was collected and spun again at 14,000 RPM. The supernatant was mixed 1:1 with 0.2% RapiGest in 50 mM NH4HCO3, pH 7.8. The protein was then reduced, alkylated, and digested with trypsin as described above for yeast proteins. The resulting peptides were stored at -80 °C until analysis by μLC/μLC/MS/MS as described below.

1.4 Microcapillary liquid chromatography tandem mass spectrometry

Fused silica capillary tubing (75 μm I.D.; Polymicro Technologies) was pulled to a tip of $\sim$5 μm at one end and packed with 60 cm of Jupiter Proteo reversed phase chromatography material (Phenomonex, Torrance, CA). The column was then placed in-line with an Agilent 1100 HPLC system and an LTQ ion trap mass spectrometer. Peptides from 5 μg of total protein were loaded onto the microcapillary column from the autosampler as described previously [3]. Peptides were then separated using an automated 4 hour HPLC program. The effluent from the column was electrosprayed into the LTQ using a distal voltage (2.2 kV) applied directly to the solvent. MS/MS spectra were acquired using data-dependent acquisition with a single MS survey scan triggering five MS/MS scans. Precursor ions were isolated using a 2 m/z isolation window and activated with 35% normalized collision energy. The automatic gain control was set to 30,000 and 2,000 charges for MS and MS/MS spectra respectively.

1.5 2D-liquid chromatography tandem mass spectrometry (μLC/μLC/MS/MS) of *C. elegans* peptides

A triphasic column was constructed of 100 μm I.D. fused silica capillary tubing pulled to a tip. The column was packed first with 8 cm of Luna C18 chromatography material, second with 4 cm of strong cation exchange material (SCX; Whatman), and finally with an additional 4 cm of Luna C18. The column was equilibrated in 95% acetonitrile, 5% water, and 0.1% formic acid for 30 minutes and then peptides from 100 μg of *C. elegans* total protein was loaded directly onto the column using a loading bomb pressurized with 1,000 PSI of helium gas. Peptides were separated using a 12-step MudPIT (multidimensional protein identification technology) program as described previously[9]. MS/MS spectra were acquired on an LTQ mass spectrometer as described above for the yeast peptides.

1.6 Charge state determination

The charge state of each spectrum was estimated by a simple heuristic that distinguishes between singly charged and multiply charged peptides using the fraction of the measured signal above and below the precuror m/z [4]. No attempt to distinguish between 2+ or 3+ spectra were made other than limiting the database search to peptides with a calculated M+H mass of 700 to 4,000 Da. Thus, of the 35,236 spectra, 737 were searched at 1+ charge state, 30 were searched at 2+ charge state, and the remaining (30,469) were searched at both 2+ and 3+ charge states.

2 Supplementary Data: Additional experiments

2.1 Analysis of *C. elegans* data set

We investigated Percolator's behavior on a larger data set, a 24 hour MudPIT analysis of *C. elegans* proteins containing 207,804 spectra. The analysis was performed with 12 salt steps from the strong cation exchange resin. We processed 202,586 spectra both for charge 2+ and 3+, yielding a total of 410,390 PSMs. Percolator's analysis of the spectra took 26 minutes on an Athlon MP Opteron 842 CPU and identified 70,152 PSMs at a q value of 0.01, corresponding to 12,252 unique peptides and 3,219 proteins. Percolator identifies 15% more PSMs than PeptideProphet (61,186 PSMs) and 7.5% more unique peptides (11,400 peptides). When the *C. elegans* data was analyzed using the method of Washburn *et al.* [8], 55,739 PSMs were identified with a q value of 0.085, corresponding to 13,197 unique peptides and 4,307 proteins. At this more relaxed threshold, Percolator identifies 48% more PSMs (82,516) corresponding to 18,394 unique peptides from 5,671 proteins.

2.2 Control experiments

We performed a variety of control experiments to verify Percolator's performance. First, as a negative control, we attempted to use Percolator to identify correct PSMs in a collection of decoy PSMs. In this experiment, the SVM tried to distinguish between two collections of decoy PSMs. We then used a third set of decoy PSMs to estimate q values. We repeated the procedure ten times using the yeast data set. Percolator identified an average of 47 PSMs (minimum of 0 and maximum of 170). As a second negative control, we adjusted the SEQUEST settings to score peptides using amino acid masses that were increased by 11 Daltons from their true masses. The purposely erroneous settings should render only false identifications. Percolator found no PSMs with q values less than 0.01 under these conditions.

As a more realistic test we created a hybrid data set consisting of the original collection of target PSMs from the yeast data set plus an equal number of decoy PSMs. Across ten repetitions, Percolator identifies on average 11,293 target PSMs and 64 decoy PSMs in this hybrid data set. The small (11%) decrease in the number of identifications of target PSMs is not surprising, given the large number of decoys that we added to the data set. Furthermore, because we use a q value threshold of 0.01, we expect the complete set of 11,357 identified PSMs to contain approximately 113 incorrect identifications, including $113/(1 + 0.9) = 59$ decoy PSMs and $113 - 59 = 54$ target PSMs. Thus, identifying on average 64 decoy PSMs is reasonable.

Finally, to verify that our method is not over-fitting to our particular data set, we performed three additional mass spectrometry analyses from the same biological sample. The three data sets contain 74,113, 69,901 and 70,173 and PSMs. Percolator identifies 13,304, 12,139 and 12,428 PSMs with q value less than 0.01 in each of these data sets. We then trained Percolator on each of these technical replicates, and tested its ability to identify positives in the original data set. In this case, Percolator identifies 12,619, 12,482 and 12,608 positives, which is comparable to the 12,672 positives identified initially. From these technical replicates, 12,261 of the identifications are shared among all four sets.

Percolator classifies PSMs using a vector of 20 features. The weights that Percolator assigns to these features are summarized in Supplementary Table 2, for each of the yeast data sets that we analyzed. Interpreting these coefficients is difficult because the SVM is a discriminative method. Consequently, the model makes no explicit independence assumptions, and the model parameters have no designated semantics. For example, two highly correlated features may receive a large combined weight, but the model may arbitrarily divide this weight among the two features. Nonetheless, Supplementary Table 1 shows several clear trends. In all three cases, the ΔC_n score receives the highest weight, suggesting that this feature is the most useful discriminator in the set. For the tryptic data set, having enzymatic termini (enzN and enzC) is important, whereas these features are much less important for the chymotrypsin or elastase data sets. Also, all three models assign a large weight to the "numProt" feature (which counts the number of other PSMs that match to this protein), suggesting that this type of information is valuable.

In addition to examining the feature weights directly, we can estimate the relative importance of a feature by removing it and measuring the resulting change in Percolator's performance. However, once again, the discriminative nature of the SVM complicates this analysis, because removing an important feature might not lead to a performance decrease if the feature set contains a redundant feature. Consequently, we performed our feature removal analysis on collections of related features, summarized in Table 2. For the initial yeast data set, we ran Percolator, eliminating one subset of features at a time. For each run, the table lists the number of PSMs identified at a q value of 0.01, as well as the percentage decrease in identified PSMs relative to using all 20 features. Surprisingly, removing the three charge-state features results in almost no change in performance, probably because the charge state information is implicit in some of the other features. In contrast, removing features related to enzyme specificity causes a significant performance decrease (33%), and removing intra-set features causes a smaller but still significant performance decrease of 12%. We also removed all of the score-related features (XCorr, ΔC_n, etc.), and the performance dropped by 16%. However, in this case information about XCorr is still implicitly included, because the original PSMs are selected by this metric.

Percolator dynamically adjusts its scoring function in order to account for differences among data sets. As a direct illustration of the change in the discriminant function, Figure 2 plots the SVM discriminant scores for the elastase and chymotrypsin data sets. In each panel, the x-axis is the discriminant from the SVM produced using the tryptic data set, and the y-axis is the discriminant from the SVM produced using the given data set. Again, in each case, the enzyme specificity features are computed correctly with respect to the given data set. In the figure, many points deviate significantly from the line $y = x$, and the largest deviation is in the region of greatest interest—the PSMs that achieve a q value greater than 0.01.

2.3 Analysis using InsPecT

Thus far, all of our experiments have involved post-processing candidate PSMs generated by SEQUEST. To demonstrate Percolator's generality, we re-analyzed our initial data set of 69,705 yeast PSMs using the InsPecT algorithm [7], and then used Percolator to rank the resulting collection of PSMs. For this analysis, we used a collection of ten features computed

by InsPecT (see Supplementary Methods), plus the ten additional features that are computed by Percolator. Figure 1A shows that Percolator achieves a comparable level of improvement over either SEQUEST or InsPecT. At a q value of 0.01 the combination of InsPecT and Percolator finds 7,334 PSMs while SEQUEST and Percolator finds 12,673 PSMs, corresponding to 5,425 and 8,198 unique peptides, respectively. Furthermore, because most of our analyses thus far have focused on coupling Percolator with SEQUEST, it is likely that a richer or more finely tuned collection of features would yield a greater performance improvement with InsPecT. Finally, we note that the two methods overlap for 5,331 PSMs and 4,765 unique peptides. This result suggests that SEQUEST and InsPecT PSMs are complementary, and that using both search algorithms on a given data set might be beneficial.

Supplementary References

[1] B. E. Frewen, G. E. Merrihew, C. C. Wu, W. S. Noble, and M. J. MacCoss. Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. *Analytical Chemistry*, 78(5678–5684), 2006.

[2] S. Keerthi and D. DeCoste. A modified finite Newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6:341–361, 2005.

[3] A. A. Klammer and M. J. MacCoss. Effects of modified digestion schemes on the identification of proteins from complex mixtures. *Journal of Proteome Research*, 5(3):695–700, 2006.

[4] A. A. Klammer, C. C. Wu, M. J. MacCoss, and W. S. Noble. Peptide charge state determination for low-resolution tandem mass spectra. In *Proceedings of the Computational Systems Bioinformatics Conference*, pages 175–185, 2005.

[5] V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear SVMs. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 477–484, New York, NY, USA, 2006. ACM Press.

[6] J. Storey and R. Tibshirani. Statistical significance for genome-wide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100:9440–9445, 2003.

[7] S. Tanner, H. Shu, A. Frank, Ling-Chi Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna. Inspect: Identification of posttranslationally modified peptides from tandem mass spectra. *Analytical Chemistry*, 77:4626–4639, 2005.

[8] M. P. Washburn, D. Wolters, and J. R. Yates, III. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature Biotechnology*, 19:242–247, 2001.

[9] Christine C Wu, Michael J MacCoss, Kathryn E Howell, and John R 3rd Yates. A method for the comprehensive proteomic analysis of membrane proteins. *Nat Biotechnol*, 21(5):532–538, May 2003.

[10] N. Zhang, X. J. Li, M. Ye, S. Pan, B. Schwikowski, and R. Aebersold. ProbIDtree: an automated software program capable of identifying multiple peptides from a single collision-induced dissociation spectrum collected by a tandem mass spectrometer. *Proteomics*, 5:4096–4106, 2005.