Improved Classification of Mass Spectrometry Database Search Results Using Newer Machine Learning Approaches*

Peter J. Ulintz‡§¶, Ji Zhu‖, Zhaohui S. Qin§**, and Philip C. Andrews‡§

Manual analysis of mass spectrometry data is a current bottleneck in high throughput proteomics. In particular, the need to manually validate the results of mass spectrometry database searching algorithms can be prohibitively time-consuming. Development of software tools that attempt to quantify the confidence in the assignment of a protein or peptide identity to a mass spectrum is an area of active interest. We sought to extend work in this area by investigating the potential of recent machine learning algorithms to improve the accuracy of these approaches and as a flexible framework for accommodating new data features. Specifically we demonstrated the ability of boosting and random forest approaches to improve the discrimination of true hits from false positive identifications in the results of mass spectrometry database search engines compared with thresholding and other machine learning approaches. We accommodated additional attributes obtainable from database search results, including a factor addressing proton mobility. Performance was evaluated using publically available electrospray data and a new collection of MALDI data generated from purified human reference proteins. ***Molecular & Cellular Proteomics 5:497–509, 2006.***

The field of proteomics is driven by the need to develop increasingly high throughput methods for the identification and characterization of proteins. MS is the primary experimental method for protein identification; MS/MS in particular is now the *de facto* standard identification technology, providing the ability to rapidly characterize thousands of peptides in a complex mixture. Instrument development continues to improve the sensitivity, accuracy, and throughput of analysis. Current instruments are capable of routinely generating several thousand spectra per day, detecting subfemtomolar levels of peptide at 10 ppm mass accuracy or better. Such an increase in instrument performance is limited, however, without effective tools for automated data analysis. In fact, the primary bottleneck in high throughput proteomic production "pipelines" is in many cases no longer the rate at which the instrument can generate data, but rather it is in quality analysis and interpretation of the results to generate confident protein assignments. This bottleneck is primarily due to the fact that it is often difficult to distinguish true hits from false positives in the results generated by automated mass spectrometry database search algorithms. All MS database search approaches produce scores describing how well a peptide sequence matches experimental fragmentation, yet classifying hits as "correct" or "incorrect" based on a simple score threshold frequently produces unacceptable false positive/false negative rates. Consequently manual validation is often required to be truly confident in the assignment of a database protein to a spectrum.

Software and heuristics for automated and accurate spectral identification (1–7) and discrimination of correct and incorrect hits (8–16) are thus an ongoing effort in the proteomics community with the ultimate goal being completely automated MS data interpretation. The most straightforward approach to automated analysis is to define specific score-based filtering thresholds as discriminators of correctness, *e.g.* accepting SEQUEST scores of doubly charged fully tryptic peptides with XCorr $>$ 2.2 and ΔCn values of at least 0.1 (17); these thresholds are typically published as the criteria for which correctness is defined. Other efforts have focused on establishing statistical methods for inferring the likelihood that a given hit is a random event. A well known example of this is the significance threshold calculated by the Mascot search algorithm, which by default displays a threshold indicating the predicted probability of an assignment being greater than 5% likely to be a false positive based on the size of the database. Use of a reverse database search to provide a measure of false positive rate is another method frequently used (8, 18). More formally, Sadygov and Yates (12) model the frequency of fragment ion matches from a peptide sequence database matching a spectrum as a hypergeometric distribution, a model also incorporated into the openly available X!Tandem algorithm (6, 13); whereas Geer *et al.* (7) model this distribution as a Poisson distribution.

Several of these approaches have been implemented directly in the scoring calculation of new search algorithms (6, 7, 12, 15). Alternatively external algorithms may be developed that process the output of the more standard search platforms such as Mascot or SEQUEST, classifying results as

From the ‡National Resource for Proteomics and Pathways, §Bioinformatics Program, ‖Department of Statistics, and **Department of Biostatistics, School of Public Health, University of Michigan, Ann Arbor, Michigan 48109

Received, July 27, 2005, and in revised form, November 10, 2005
Published, MCP Papers in Press, November 30, 2005, DOI 10.1074/mcp.M500233-MCP200

either correct or incorrect with an associated probability. Examples of the latter type include PeptideProphet (11) and QScore (8). These tools have the advantage of being able to accommodate results from these existing, well established search engines that may already be in place in a production laboratory; conversely approaches in which the quality measures are built into the search algorithm scoring are arguably more user-friendly in that they eliminate the extra postsearch processing step of having to run a second algorithm.

Keller *et al.* (11) were among the first to implement a generic tool for classifying the results of common search algorithms as either correct or incorrect. Their PeptideProphet tool represents arguably the most well known openly available tool implementing a probabilistic approach to assess the validity of peptide assignments generated by MS database search algorithms. Their approach contains elements of both supervised and unsupervised learning, achieving a much higher sensitivity than conventional methods based on scoring thresholds. One concern with PeptideProphet, however, is the degree to which the supervised component of the model can be generalized to new types of data and the ease with which new potentially useful information can be added to the algorithm.

This work attempts to address these difficulties by applying a set of simple "over the counter" methods to the challenging peptide identification problem. Anderson *et al.* (14) demonstrated that support vector machines could perform well on ion trap spectra searched using the SEQUEST algorithm. In this study, we demonstrated that the latest machine learning techniques for classification, namely tree-based ensemble methods such as boosting and random forest, are more suitable for the peptide classification problem and provide improved classification accuracy. The rationale for the improvements lies in their ability to efficiently combine information from multiple easy-to-get but dependent and weakly discriminatory attributes. Such work will hopefully result in development of software tools that are easily installed in a production laboratory setting that would allow convenient filtering of false identifications with an acceptably high accuracy either as new tools or as a complement to currently existing software. The problem of classification of mass spectrometry-based peptide identification seems well suited to these algorithms and could lead to more readily usable software for automated analysis of the results of mass spectrometry experiments.

EXPERIMENTAL PROCEDURES

Overview of Classification Techniques Used

Mixture Model Approach in PeptideProphet—Among all the methods that have been proposed in the literature for the peptide identification problem, the mixture model approach implemented in the PeptideProphet algorithm (11) is perhaps the most well known. In this method, a discriminant score function, $F(x_1, x_2, \ldots, x_5) = c_0 + c_1x_1 \ldots + c_5x_5$, is defined to combine database search scores $x_1, x_2, \ldots, x_S$ where c_i values are weights. Based on a training dataset, a Gaussian distribution is chosen to model the discriminant scores corresponding to correct peptide assignments, and a Gamma distribution is selected to model the asymmetric discriminant scores corresponding to incorrect peptide assignments. All the scores are therefore represented by a mixture model $p(x) = rf_1(x) + (1 - r)f_2(x)$ where $f_1(x)$ and $f_2(x)$ represent the density functions of the two types of discriminant scores, and r is the proportion of correct peptide identifications. For each new test dataset, the expectation maximization algorithm (19) is used to estimate the probability that the peptide identified is correct. A decision can be made by comparing the probability with a prespecified threshold. When compared with conventional means of filtering data based on SEQUEST scores and other criteria, the mixture model approach achieves much higher sensitivity.

A crucial part of the above approach is the choice of discriminant score function F. In (11), the c_i values are derived to maximize the between- *versus* within-class variation under the multivariate normal assumption using training data. To make this method work, one has to assume that the training data and the test data are generated from the same source. When a new set of discriminant scores is generated and needs to be classified, one has to retrain the c_i weight parameters using a new corresponding training set; in other words, the discriminant function F is data-dependent. In an area such as proteomics in which there is a good amount of heterogeneity in instrumentation, protocol, database, and database searching software, it is fairly common to come across data that display significant differences. It is unclear to what degree the results of a classification algorithm are sensitive to these differences, hence it is desirable to automate the discriminant function training step. Another potential issue is the Normal and Gamma distribution used to model the two types of discriminant scores. There is no theoretical explanation why the discriminant scores should follow these two distributions; in fact, a Gamma distribution rather than a Normal distribution may be appropriate for both positive and negative scores when using the Mascot algorithm.[1] It is possible that for a new set of data generated by different mass spectrometers and/or different search algorithms, the two distributions may not fit the discriminant scores well. Also certain types of data attributes or scores may be more difficult to accommodate into such a model if those attributes significantly alter the shape of the discriminant score distribution. For example, qualitative or discrete attributes may be more difficult to model. As a result, higher classification errors may be produced using this model-based approach.

Machine Learning Techniques—Distinguishing correct from incorrect peptide assignments can be regarded as a classification problem or supervised learning, a major topic in the statistical learning field. Many powerful methods have been developed such as classification and regression tree analysis, support vector machine (SVM),[2] random forest, boosting, and bagging (21). Each of these approaches has some unique features that enable them to perform well in certain scenarios; the SVM, for example, is a good tool for small sample size, large feature space situations. On the other hand, all approaches are quite flexible and have been applied to an array of biomedical problems. In this study, we applied state-of-the-art machine learning

[1] Liu, J., Beaudrie, C. E. H., Yanofsky, C., Carrillo, B., Boismenu, D., Morales, F. R., Bell, A., and Kearney, R. E. (2005) A Statistical Model for Estimating Reliability of Peptide Identifications Using Mascot, Poster WP21389 presented at the 53rd American Society for Mass Spectrometry Conference on Mass Spectrometry and Allied Topics, San Antonio, Texas (June 5–9, 2005).

[2] The abbreviations used are: SVM, support vector machine; N-term, N-terminal; C-term, C-terminal; NTT, number of tryptic termini; MPF, mobile proton factor; SPI, scored peak intensity; BCS, backbone cleavage score; Sp, preliminary score; ROC, receiver operating characteristic.

approaches to the peptide assignment problem.

Boosting—The boosting idea, first introduced by Freund and Schapire (22) with their AdaBoost algorithm, is one of the most powerful learning techniques introduced during the past decade. It is a procedure that combines many "weak" classifiers to achieve a final powerful classifier. Here we give a concise description of boosting in the two-class classification setting. Suppose we have a set of training samples, where x_i is a vector of input variables (in this case, various scores and attributes of an individual MS database search result produced from an algorithm such as SEQUEST) and y_i is the output variable coded as -1 or 1, indicating whether the sample is an incorrect or correct assignment of a database peptide to a spectrum. Assume we have an algorithm that can build a classifier $T(x)$ using weighted training samples so that, when given a new input x, $T(x)$ produces a prediction taking one of the two values $\{-1, 1\}$; the classifier $T(x)$ is typically a decision tree. Then boosting proceeds as follows: start with equal weighted training samples and build a classifier $T_1(x)$. If a training sample is misclassified, *e.g.* an incorrect peptide is assigned to the spectrum, the weight of that sample is increased (boosted). A second classifier $T_2(x)$ is then built with the training samples but, using the new weights, is no longer equal. Again misclassified samples have their weights boosted, and the procedure is repeated M times. Typically one may build hundreds or thousands of classifiers this way. A final score is then assigned to any input x, defined to be a linear (weighted) combination of the classifiers. A high score indicates that the sample is most likely a correctly assigned protein with a low score indicating that it is most likely an incorrect hit. By choosing a particular value of the score as a threshold, one can select a desired specificity or a desired ratio of correct to incorrect assignments.

Random Forests—Similar to boosting, the random forest (23) is also an ensemble method that combines many decision trees. However, there are three primary differences in how the trees are grown. 1) Instead of assigning different weights to the training samples, the method randomly selects, with replacement, n samples from the original training data. 2) Instead of considering all input variables at each split of the decision tree, a small group of input variables on which to split are randomly selected. 3) Each tree is grown to the largest extent possible. To classify a new sample from an input, one runs the input down each of the trees in the forest. Each tree gives a classification (vote). The forest chooses the classification having the most votes over all the trees in the forest. The random forest enjoys several nice features: like boosting, it is robust with respect to input variable noise and overfitting, and it gives estimates of what variables are important in the classification. A discussion of the relative importance of the different attributes used in our analysis of MS search results is given under "Results."

Support Vector Machines—The SVM is another successful learning technique (24). It typically produces a non-linear classification boundary in the original input space by constructing a linear boundary in a transformed version of the original input space. The dimension of the transformed space can be very large, even infinite in some cases. This seemingly prohibitive computation is achieved through a positive definite reproducing kernel, which gives the inner product in the transformed space. The SVM also has a nice geometrical interpretation in the finding of a hyperplane in the transformed space that separates two classes by the biggest margin in the training samples, although this is usually only an approximate statement due to a cost parameter. The SVM has been successfully applied to diverse scientific and engineering problems, including the life sciences (25–27). Anderson *et al.* (14) introduced the SVM to MS/MS spectra analysis, classifying SEQUEST results as correct and incorrect peptide assignments. Their result indicates that the SVM yields less false positives and false negatives compared with other cutoff approaches.

TABLE I
Number of spectra examples for the ESI-SEQUEST dataset and the MALDI-Spectrum Mill dataset

	Training	Testing
ESI-SEQUEST		
Correct	1,930	827
Incorrect	24,001	10,286
Total	25,931	11,113
MALDI-Spectrum Mill		
Correct	731	313
Incorrect	7,504	3,216
Total	8,235	3,529

However, one weakness of the SVM is that it only estimates the category of the classification, whereas the assignment probability $p(x)$ may be of interest itself where $p(x) = P(Y = 1|X = x)$ is the posterior probability of a sample being in class 1 (*i.e.* a correctly identified peptide). Another problem with the SVM is that it is not trivial to select the best tuning parameters for the kernel and the cost. Often a grid search scheme has to be used; this can be time-consuming. In comparison, boosting and the random forest are very robust, and the amount of tuning needed is rather modest compared with the SVM.

Reference Datasets

Two collections of mass spectrometry data were used in this study, representing the two most common protein MS ionization approaches: ESI and MALDI. We benchmarked the performance of boosting and random forest methods in comparison with other approaches using a known, published ESI dataset and then further applied these methods to newer in-house generated MALDI data from an Applied Biosystems Inc. (Foster City, CA) TOF/TOF mass spectrometer (28).

ESI-SEQUEST Dataset—The electrospray dataset was kindly provided by Andy Keller as described in Refs. 11 and 29. These data are combined MS/MS spectra generated from 22 different LC/MS/MS runs on a control sample of 18 known (non-human) proteins mixed in varying concentrations. A ThermoFinnigan ion trap mass spectrometer was used to generate the dataset. In total, the data consist of 37,044 spectra of three parent ion charge states: $[M + H]^+$, $[M + 2H]^{2+}$, and $[M + 3H]^{3+}$. Each spectrum was searched by SEQUEST against a human protein database with the known protein sequences appended. The top scoring peptide hit was retained for each spectrum; top hits against the known 18 proteins were labeled as correct and manually verified by Keller *et al.* (11). All peptide assignments corresponding to proteins other than the 18 in the standard sample mixture and common contaminants were labeled as incorrect. In all, 2757 (7.44%) peptide assignments were determined to be correct. The distribution of hits as used to train and test the methods used in this study are indicated in Table I.

MALDI-Spectrum Mill Dataset—We wished to evaluate performance of the algorithms on data generated using different instrumentation, namely MALDI data. Toward that end, 300 purified recombinant human protein samples were procured from Genway Biotech Inc. (San Diego, CA). Aliquots of these proteins were resolved by one-dimensional SDS-PAGE to confirm purity and protein molecular weight. Plugs from the band on each one-dimensional gel were subjected to in-gel trypsin digestion, serially diluted (in many cases) to generate potentially more "realistic" spectra, and cleaned by C_{18} ZipTip. Resulting digestions for each protein were spotted in four replicates on MALDI target plates, and MS/MS spectra were acquired on an Applied Biosystems Inc. 4700 Proteomics Analyzer ("TOF/TOF"). Spectra were collected from the successive replicate spots by selecting the most abundant ions from each replicate and excluding previously selected peaks until reasonably

TABLE II
SEQUEST attribute descriptions

Attribute names in bold are treated as discrete categorical variables. DB, database.

Attribute group	Attribute name	SEQUEST name	Description
PeptideProphet (I)	Delta MH^+	$(M + H)^+$	Parent ion mass error between observed and theoretical
	Sp rank	Rank/Sp	Initial peptide rank based on preliminary score
	ΔCn	deltCn	1 − Cn: difference in normalized correlation scores between next best and best hits
	XCorr	XCorr	Cross-correlation score between experimental and theoretical spectra
	Length	Inferred from peptide	Length of the peptide sequence
NTT (II)	NTT	Inferred from peptide	Measures whether the peptide is fully tryptic, partially tryptic, or non-tryptic (2, 1, or 0, respectively)
Additional (III)	**Parent charge**	(+1), (+2), (+3)	Charge of the parent ion
	Total intensity	Total inten	Normalized summed intensity of peaks
	DB peptides within mass window	No. matched peptides	Number of database peptides matching the parent peak mass within the specified mass tolerance
	Sp	Sp	Preliminary score for a peptide match
	Ion ratio	Ions	Fraction of theoretical peaks matched in the preliminary score
	C-term residue	Inferred from peptide	Amino acid residue at the C terminus of the peptide (1 = Arg, 2 = Lys, 0 = other)
	Number of prolines	Inferred from peptide	Number of prolines in the peptide
	Number of arginines	Inferred from peptide	Number of arginines in the peptide
Calculated (IV)	Proton mobility factor	Calculated	A measure of the ratio of basic amino acids to free protons for a peptide (described under 'Experimental Procedures')

sized MS peaks could no longer be detected; this process resulted in up to 24 MS/MS spectra for each standard protein. At the time of this analysis, results from 158 of these protein standards had been generated and were used to compose the dataset.

To generate testing and training datasets for this analysis, all MALDI spectra were searched using the Agilent (Palo Alto, CA) Spectrum Mill platform (specifying trypsin as the proteolytic enzyme and accommodating oxidized methionine and pyro-Glu modifications) against a version of the National Center for Biotechnology Information non-redundant (NCBInr) human dataset downloaded March 10, 2005. The NCBInr database was modified by replacing all entries corresponding to the Genway protein standards with annotated entries that include the appropriate N-term affinity tag (either T7 or His_6) and appending the *E. coli* K12 sequences because *E. coli* was the host organism in which the recombinants were produced. Tolerances of 0.7 Da for precursor ion selection and 0.3 Da for fragment ion selection were used in the search with a minimum matched peak intensity of 40%. "Correct/incorrect" labels were assigned to each spectrum in a semiautomated manner by correlating the accession number of the search result with that of the protein digest known to be spotted at the plate location from which each spectrum was derived. The dataset consists of 11,764 search results from 4340 spectra (up to the top five ranking hits for each spectrum). 1044 are "true positive" hits; of these, 111 are non-top ranking hits. The precise distribution of hits as used to test the algorithms are shown in Table I. Note the relatively low fraction of true positives as per what would be expected from this instrument is primarily due to the fact that the top five ranking hits, not just the top hits, were selected from every search result. We wished to include non-top ranking hits to examine whether the machine learning tools could be used to distinguish correct hits among these lower ranked results, a frequent occurrence in MS/MS search data. The lower ranked hits contain potentially valuable protein identifications that would be discarded using many normal approaches.

The complete set of annotated MALDI protein standard data (Aurum) is available for download on www.proteomecommons.org. To our knowledge, this represents the first publicly available annotated MALDI dataset useful for development of these types of algorithms.

Attributes Extracted from Each Dataset

Both *.out search result files from SEQUEST and *.spo result files from Spectrum Mill were parsed into a simple text row/column format suitable for use by pattern classification algorithms using custom modules written in Python (available upon request). For the SEQUEST results, only the top hit for each spectrum was parsed; again, for the MALDI dataset, the top five ranking hits were retained.

SEQUEST—The attributes extracted from SEQUEST assignments are listed in Table II. Attributes include typical scores generated by the SEQUEST algorithm (Sp, Sp rank, ΔCn, and XCorr) as well as other statistics included in a SEQUEST report (total intensity, number of matching peaks, and fragment ion ratio). We include length among the PeptideProphet attributes because PeptideProphet normalizes the XCorr attribute using the length of the peptide. Number of tryptic termini (NTT) is a useful measure for search results obtained by specifying no proteolytic enzyme and is used extensively in Ref. 11. Other attributes include features readily obtainable from the candidate peptide sequence: C-term residue (Lys = "1," Arg = "2," and others = "0"), number of prolines, and number of arginines. A new statistic, the mobile proton factor (MPF), is calculated as follows.

$$\frac{(1.0 \times R) + (0.8 \times K) + (0.5 \times H)}{\text{Charge}} \quad \text{(Eq. 1)}$$

MPF attempts to provide a simple measure of the mobility of protons in a peptide, a theoretical measure of the ease of which a peptide may be fragmented in the gas phase (30–32). A smaller value for MPF is indicative of higher protein mobility, whereas peptides with MPF ≥ 1 can be considered "nonmobile." *R*, *K*, and *H* refer to the number of Arg, Lys, and His residues present in the sequence, reflecting the overall basicity of the peptide. The coefficients for these factors reflect the relative basicity of the three residues normalized to the dissociation constant of arginine (the pK_a values of Arg, Lys, and His are 12.0, 10.0, and 5.9, respectively). *Charge* indicates the charge on the parent peptide, reflecting the number of free protons potentially available for charge-directed fragmentation. We included MPF to

TABLE III
Spectrum Mill attribute descriptions

Attribute name	Attribute description
Delta MH^+	Parent ion mass error between observed and theoretical
Rank	Rank by score and number of unmatched ions of the peptide among all peptides matched for the spectrum
Score	Spectrum Mill score
Percent SPI	Percentage of total peak intensity from observed peaks that match theoretical fragment ion masses
BCS	Number of backbone cleavage events from which a y or a b ion is observed
Unused ion ratio	(Number of observed peaks not matched to fragment ion masses)/(number of observed peaks matched to fragment ion masses)

demonstrate the ease of accommodation of additional information into the classification algorithms, amounting to simply adding an additional data column to the dataset.

Spectrum Mill—Spectrum Mill attributes are indicated in Table III. Spectrum Mill is a search platform based initially on the ProteinProspector set of scripts further developed by Karl Clauser (3) and commercialized by Agilent. The primary Spectrum Mill score is nonprobabilistic, intended as an absolute measure of the information contained in an assignment of a spectrum with a peptide sequence, and is database size-independent. The score increases with peaks matching theoretical fragment ions types (in an instrument-dependent way) with a penalty for unmatched peaks. Intensity of peaks is a factor in the score as is peptide length. The scoring system accommodates all major fragmentation ion types and neutral losses and the presence of internal and immonium ions. Scored peak intensity (SPI) is the second primary Spectrum Mill scoring measure, reflecting the percentage of total peak intensity matching predicted fragment ion masses. Spectrum Mill implements semiautomated spectrum validation and curation tools based on linear thresholds for primary score and SPI. Backbone cleavage score (BCS) indicates the number of cleavage events generating b- or y-ions; unused ion ratio (number of unused ions/number of total ions after peak thresholding) provides a measure of the amount of signal not accounted for in the match; and delta parent mass measures the difference between the observed and experimental peptide precursor masses. Terms such as parent charge were avoided in the Spectrum Mill data due to the fact that the ions are produced by MALDI.

Implementation Specifics

A single training and testing dataset was constructed for each of the ESI-SEQUEST and MALDI-Spectrum Mill datasets by random selection. Random sampling was done separately for correct-labeled and incorrect-labeled data so that both training and testing data contain the same proportions. For all results, evaluation was done on a test set that does not overlap the training set. Two-thirds of all data were used for training, and one-third was used for testing.

The PeptideProphet standalone application used in this analysis was downloaded from peptideprophet.sourceforge.net. PeptideProphet is also available as part of the Trans-Proteomics Pipeline being developed at the Seattle Proteome Center (tools.proteomecenter.org/TPP.php). All SEQUEST *.out result files corresponding to each test set were placed in a separate directory and processed using the out2summary.c script to generate the PeptideProphet html input file. PeptideProphet was run by executing the runPeptideProphet script using default parameters. PeptideProphet was not run on the MALDI-Spectrum Mill dataset.

For the boosting and random forest approaches, we used contributed packages for the R programming language. We use an implementation of the AdaBoost algorithm (22) implemented in the R statistical programming language by Greg Ridgway (rweb.stat.umn.edu/R/library/gbm/html/gbm.html) and the randomForest version 4.5-8 package, an R port by Andy Liaw and Matthew Wiener of the original Fortran algorithm developed by Leo Brieman and Adele Cutler. In general, we did not fine tune the parameters (*i.e.* tree size, number of trees, etc.) of the random forest and boosting implementations for two reasons: classification performances of both the random forest and boosting are fairly robust to these parameters and also because our ultimate goal is to provide a software tool that can be easily used in a production laboratory setting without a significant tuning requirement. For the AdaBoost analysis, we used decision trees with 40 leaves for the weak classifier and fixed the number of boosting iterations (M) equal to 1000. For random forests, the default number of attributes for each tree (one-third of the total number of attributes) was used except for the five-variable case in which the number of attributes was fixed at two. The default number of trees in the forest was 500, and each tree in the forest was grown until the leaf was either pure or had only five samples.

With the support vector machine, we chose a radial kernel to classify the samples as implemented in the libSVM package version 2.7 (www.csie.ntu.edu.tw/~cjlin/libsvm/). The radial kernel is flexible and performed well in preliminary studies. To select the optimal set of tuning parameters for radial kernel, a grid search scheme was adopted using a modified version of the *grid.py* python script distributed with the libSVM package. Optimal parameters are sensitive to specific training sets: for the precise results presented in this study, the optimal parameters were cost = 32768.0 and $\gamma = 3.052e-05$.

RESULTS AND DISCUSSION

Classification Performance—Each of the machine learning approaches used produces an ordering of the collection of examples in the test dataset. With ProteinProspector, the examples are ordered highest to lowest on the basis of a Bayesian posterior probability as described in Ref. 11. For either boosting or random forest, the algorithm returns, in addition to a correct/incorrect classification, an additional "fitness" term. In the case of the random forest, the fitness term can be interpreted as a probability of the identification being correct. A probability score can be generated from the boosting fitness measure as well using a simple transformation. The SVM returns a classification and a measure of the distance to a distinguishing hyperplane in attribute space that can be considered a confidence measure. When examples are ordered in this way, results can be represented as a receiver operating characteristic (ROC) plot, which provides a way of displaying the ratio of true positive classifications (sensitivity) to the fraction of false positives (1 − specificity) as a function of a variable test threshold. The threshold, chosen on the ranked ordering of results produced by the classifier,

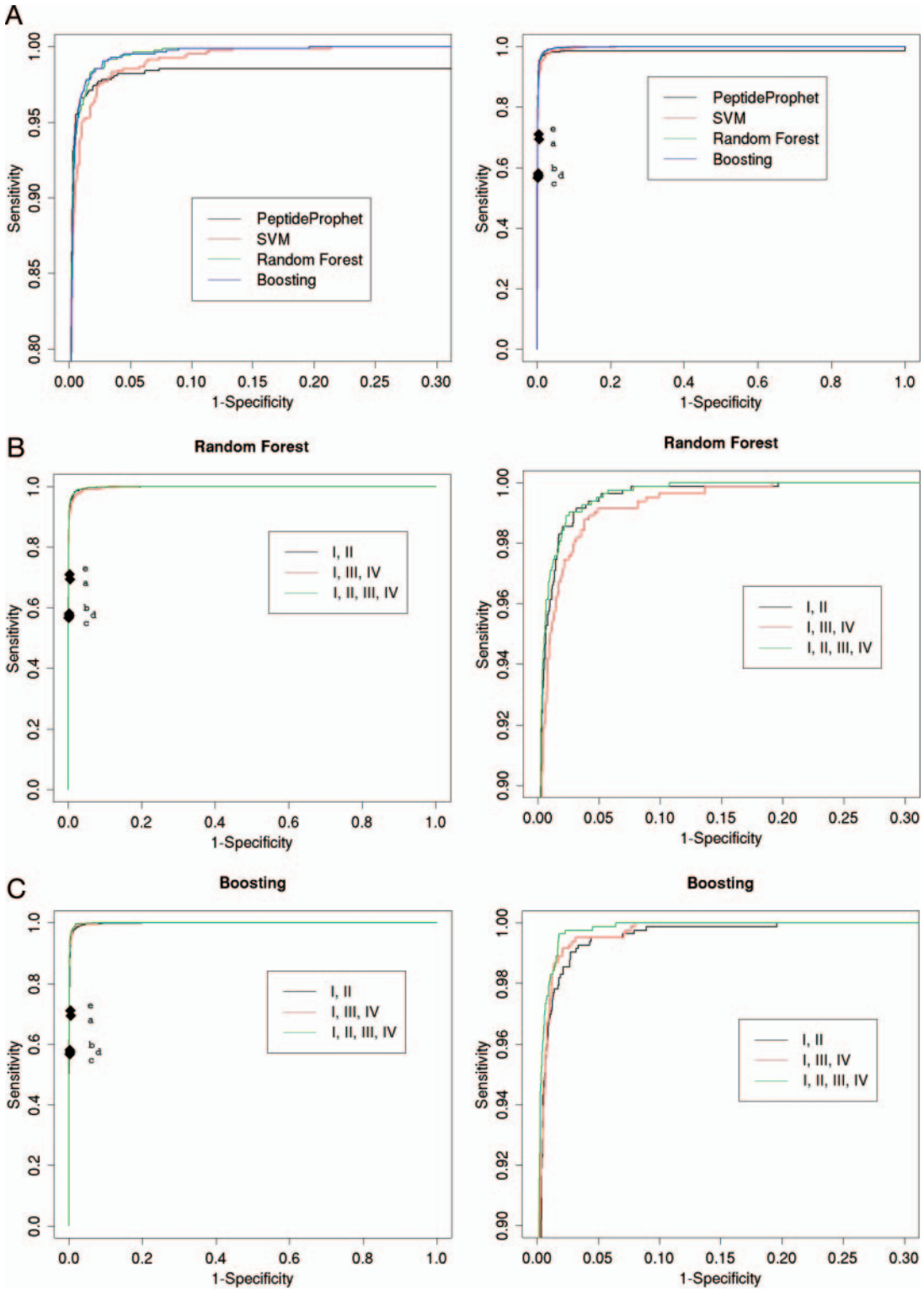

FIG. 1. **Performance of boosting and random forest methods on the ESI-SEQUEST dataset.** *A*, ROC plot of classification of the test set by PeptideProphet, SVM, boosting, and random forest methods using attribute groups I and II. The plot on the *right* is a blowup of the *upper left region* of the figure on the *left*. Also displayed are points corresponding to several sets of SEQUEST scoring statistics used as linear threshold values in published studies. The following criteria were applied for choosing correct hits (the +1, +2, and +3 numbers indicate peptide charge): *a*, +1: XCorr ≥ 1.5, NTT = 2; +2, +3: XCorr ≥ 2.0, NTT = 2 (36); *b*, ΔCn > 0.1, +1: XCorr ≥ 1.9, NTT = 2; +2: XCorr ≥ 3 or 2.2 ≤ XCorr ≤ 3.0, NTT ≥ 1; +3: XCorr ≥ 3.75, NTT ≥ 1 (17); *c*, ΔCn ≥ 0.08, +1: XCorr ≥ 1.8; +2: XCorr ≥ 2.5; +3: XCorr ≥ 3.5 (20); *d*, ΔCn ≥ 0.1, +1: XCorr ≥ 1.9, NTT = 2; +2: XCorr ≥ 2.2, NTT ≥ 1; +3: XCorr ≥ 3.75, NTT ≥ 1 (16); *e*, ΔCn ≥ 0.1, Sp rank ≤ 50, NTT ≥

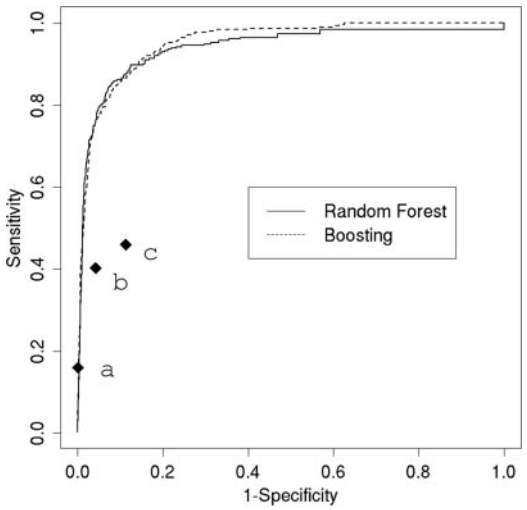

FIG. 2. **Performance of boosting and random forest methods on the MALDI-Spectrum Mill dataset.** The variables used are: rank, Spectrum Mill score, SPI, delta parent mass (delta M + H), BCS, and unused ion ratio. Points corresponding to standard guideline scoring threshold values from the Spectrum Mill documentation are displayed: *a*, Spectrum Mill score >15, SPI > 70% (outstanding); *b*, Spectrum Mill score >10, SPI > 70% (good); *c*, Spectrum Mill score >5, SPI > 70% (modest). (Note that the Spectrum Mill tool is intended to facilitate manual curation, and as such these threshold levels are only guidelines and are dataset-dependent not solid rules or recommended publication standards.)

represents a trade-off between being able to select the true positives without selecting too many false positives. If we set our scoring threshold very high, we can minimize or eliminate the number of false positives but at the expense of missing a number of true positives; conversely as we lower the scoring threshold, we select more true positives, but more false positives will be included as well. The slope at any point in the ROC plot is a measure of the degree to which one group is included at the expense of the other.

The ESI-SEQUEST dataset allowed us to compare all four classification approaches: boosting, random forests, PeptideProphet, and the SVM. ROC plots showing the results of classifying correct *versus* incorrect peptide assignments of the ESI-SEQUEST dataset using these methods are shown in Fig. 1*A*. All methods performed well on the data. As can be seen, the boosting and random forest methods provided a slight performance improvement over PeptideProphet and the SVM classification using the same six attributes. At a false positive rate of roughly 0.05%, the boosting and random forest achieved a sensitivity of 99%, whereas PeptideProphet and SVM provided a 97–98% sensitivity. We note that, although a systematic difference of 1–2% can be seen in these results, this corresponds to a relatively small number of total spectra. Also indicated in Fig. 1 are points corresponding to well known attribute thresholds from several literature citations. Each point shows the sensitivity and specificity that would be obtained on the test dataset by applying these published thresholds to the SEQUEST attributes charge, Xcorr, ΔCn, and NTT.

Of interest is the fact that the boosting, random forest, and SVM results asymptotically approached 1.0 sensitivity, whereas PeptideProphet approached a sensitivity of about 0.98 for most of the length of the ROC curve (the PeptideProphet results did achieve 1.0 sensitivity at the very end). These results point to a set of spectra that the tools learn to discriminate differently; in the case of the test set described here, the discrepancy corresponds to 14 spectra of the total 11,113. Eleven of these spectra are results annotated as correct hits that PeptideProphet assigns as incorrect and three incorrect results that PeptideProphet assigns as correct. The 11 spectra of the former category all represent singly charged spectra with very small SEQUEST ΔCn values; PeptideProphet appears to have some difficulty with instances of +1 spectra in which the second highest hit has a score very close to the top hit. The other three spectra represent an interesting case. Because the LCQ instrument on which the spectra were generated lacks the resolution to discriminate between doubly and triply charged parent ions, typical peak list extraction protocols produce two identical peak lists for non-singly charged precursor masses, one each for the doubly and triply charged case. The database search for only one of these two peak lists should produce a correct result under normal circumstances. The cases that PeptideProphet "missed" here are an exception to this rule. They are cases in which two peptides containing identical sequence from the same protein (a larger peptide with a +3 charge and one or more missed tryptic cleavage sites and a smaller peptide (a subset of the first) with a +2 charge) have the same apparent mass. For example, in one instance a parent precursor with a *m/z* of 1109 Da was selected, corresponding to a peptide from the CAH2_BOVINE protein. The CAH2_BOVINE peptide SSQQMLKFRTLNFNAEGEPELLMLANWR has a mass of 3324.82 Da. This peptide has two missed trypsin cleavage sites, a Lys at position 7 and an Arg at position 9. The peptide resulting from cleaving after position 9 is TLNFNAEGEPELLMLANWR with a mass of 2220 Da. The longer peptide was hit by SEQUEST for the +3 peak list, and the shorter one was hit for the +2 peak list: 3326/3 ≅ 2219/2 ≅ 1109 Da. Following the rule that only one of the two identical peak lists generated from the 1109-Da precursor mass spectrum can be

1, +1: not included; +2: XCorr ≥ 2.0; +3: XCorr ≥ 2.5 (11). It can be seen that all machine learning approaches provide a significant improvement over linear scoring thresholds. *B*, results of the random forest method using various sets of attributes. The *black line* represents the result of the random forest using six attributes defined in Table II as groups I and II: the SEQUEST XCorr, Sp rank, ΔCn, delta parent mass, length, and NTT. The *red line* is the result using 14 attributes, groups I, III, and IV (no NTT). The *blue line* represents the result using all attribute groups I–IV, all 15 variables. *C*, ROC plot of the boosting method using attribute groups I and II (*black*); I, III, and IV (*red*); and I–IV (*green*).

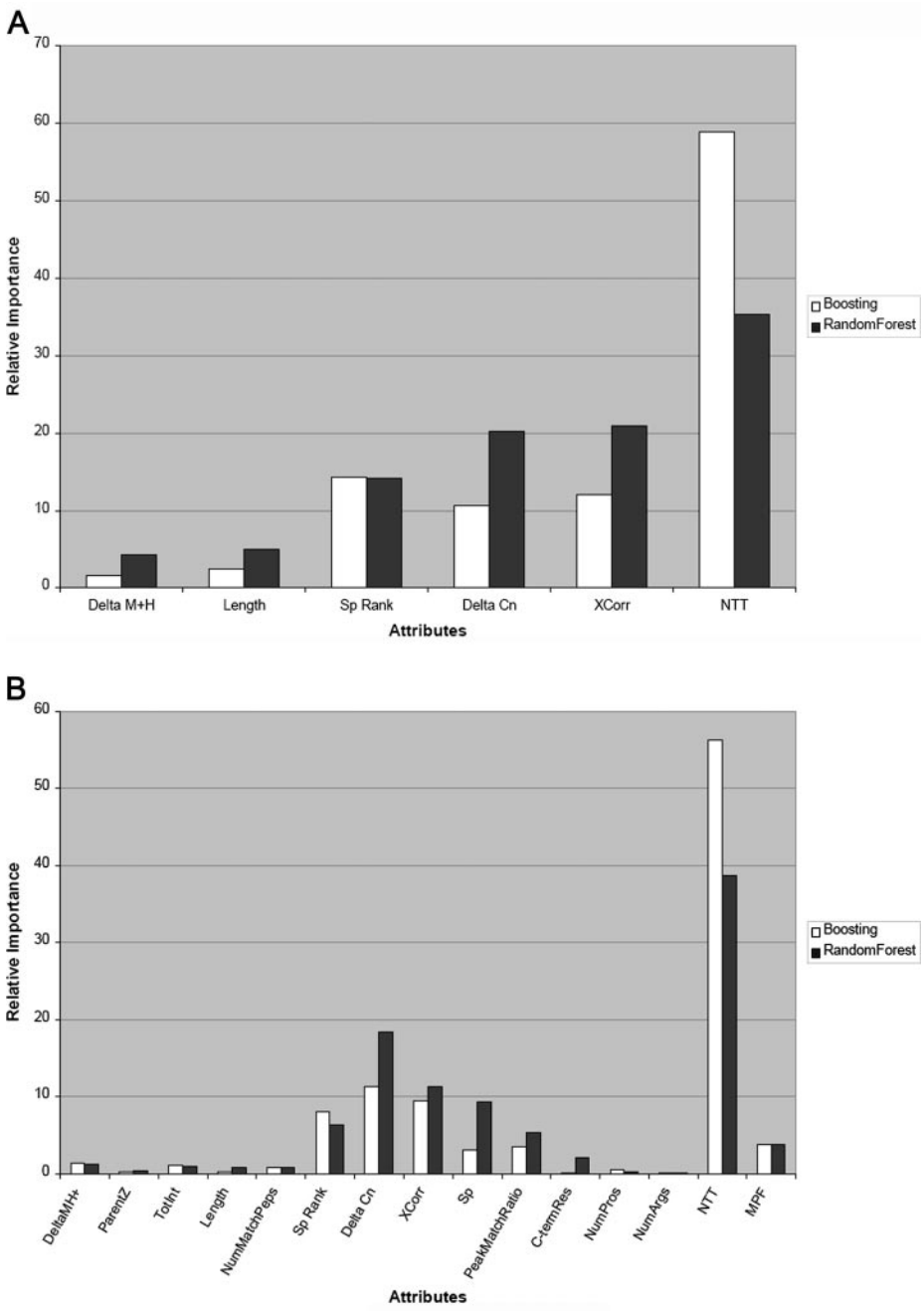

FIG. 3. **Relative importance of data attributes used for classification by boosting and random forest methods.** *A*, SEQUEST attribute importance from boosting and random forest classification using attribute groups I and II. *B*, SEQUEST attribute importance using all attributes in random forest and boosting methods. *C*, SEQUEST attribute importance using attribute groups I, III, and IV. *D*, Spectrum Mill attribute importance using random forest and boosting methods. *TotInt*, total intensity; *NumMatchPeps*, number of matching peptides; *NumPros*, number of prolines; *NumArgs*, number of argininines; *Res*, residue.

correct, the annotation provided with the ESI-SEQUEST dataset assigned the +3 spectrum as correct and the +2 spectrum as incorrect. The incorrect annotation to the search result of the +2 peak lists is misleading in these cases, however. PeptideProphet distinguished these cases, accurately providing a correct classification for the +2 spectra. The other three completely supervised algorithms learned to classify these cases as incorrect based on similar examples in the training dataset.

Fig. 1, *B* and *C*, compares the performance of the boosting and random forest methods using different sets of input attributes as shown in Table II. The panels contain the results of these algorithms using three combinations of features: 1) attribute groups I and II: the six attributes used by the PeptideProphet algorithm (SEQUEST XCorr, ΔCn, Sp rank, delta parent mass, length, and NTT), 2) attribute groups I, III, and IV (all attributes except NTT), and 3) attribute groups I–IV (all 15 variables shown in Table II). Overall it can be seen that both machine learning approaches provided improvement over the scoring thresholds described in the literature. The best performance was obtained by including all 15 variables, indicating that accommodation of additional information is benefi-

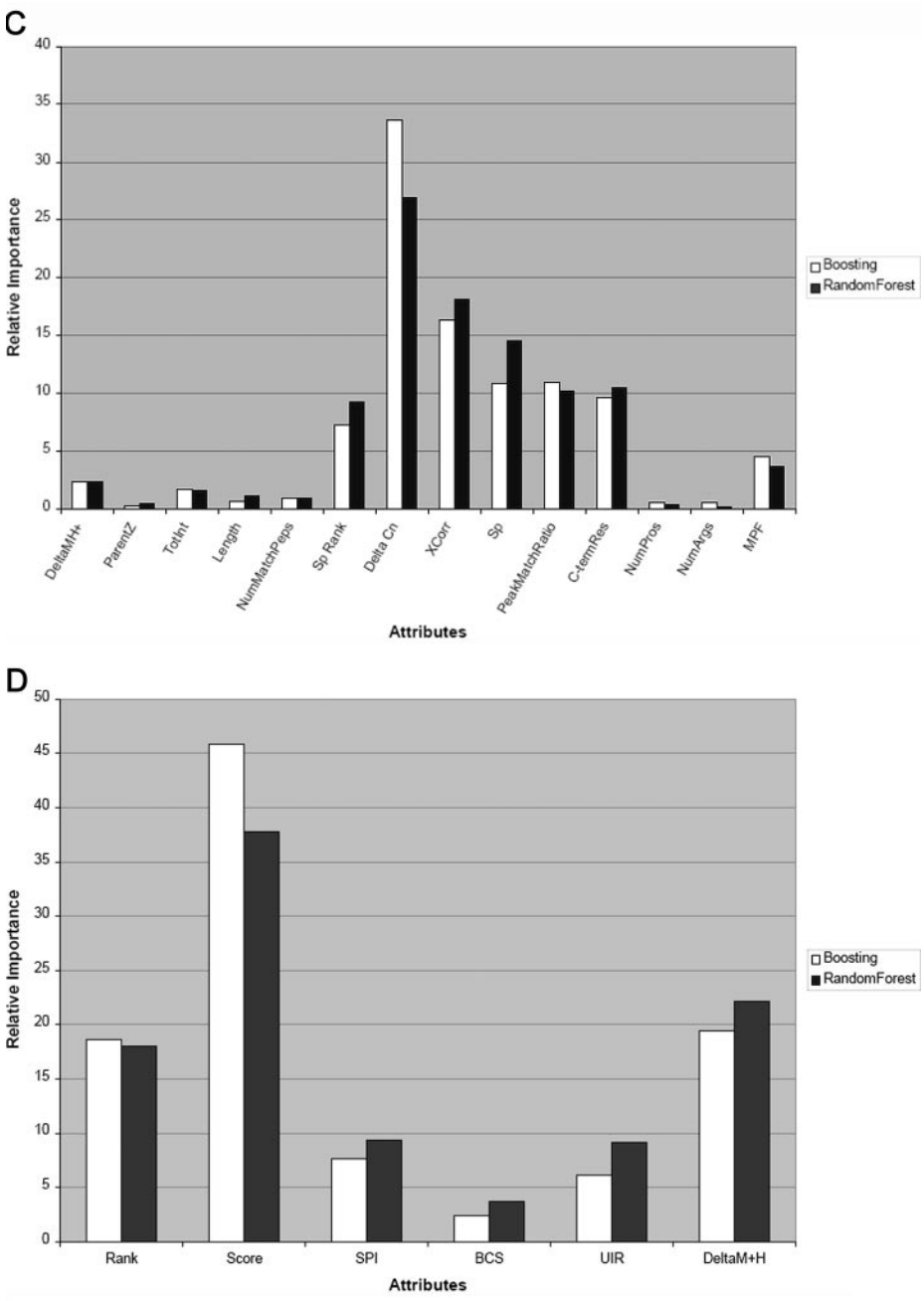

FIG. 3—*continued*

cial. The random forest appeared to be slightly more sensitive to the presence of the NTT variable than boosting. Of note is the fact that effective classification was attained by the boosting and random forest tools even in the explicit absence of the NTT variable as demonstrated by feature combination 2 despite the fact that the ESI dataset was generated using the "no enzyme" feature of SEQUEST. No enzyme specificity in the database search is often time-prohibitive in routine production work; it is much more common to restrict searches to tryptic peptides (or any other proteolytic enzyme used to digest the protein sample). Restricting to trypsin restricts results to having an NTT = 2, rendering the attribute non-discriminatory. It must be noted, however, that in this analysis the C-term residue attribute was not completely independent of NTT in that it contains residue information on one of the termini. If trypsin specificity is turned on in a search, in addition to distinguishing between Lys- and Arg-terminated peptides, C-term residue will discriminate tryptic and semitryptic peptides, the latter being possible if the peptide is the C-terminal peptide of a protein. If trypsin specificity is not used in the search (as in these data), although the C-term residue variable cannot predict the NTT value of a peptide, it can

discriminate between cases. If the C terminus of the peptide is tryptic, the peptide may be either fully or partially tryptic; if it is not, it can either be partially tryptic or non-tryptic.

The results of classifying the MALDI-Spectrum Mill data using boosting and random forest are shown in Fig. 2. The functionality necessary to run PeptideProphet on Spectrum Mill data would require customization of the tool and was therefore not used. We chose not to run the SVM on the MALDI-Spectrum Mill dataset because the superior performance of the boosting and random forest methods were already demonstrated on the ESI-SEQUEST dataset. Overall the two classifiers performed similarly with boosting outperforming random forests slightly when the false positive rate was above 15%. Both methods showed dramatic improvement over thresholding combinations based on default recommended combinations of Spectrum Mill score and SPI values. Spectrum Mill documentation suggests three guideline threshold combinations for "outstanding", "good," and "modest" hits, indicated in Fig. 2. The Outstanding threshold effectively discriminates results with a low false positive rate but discards a majority of true positives. The learning approaches are able to pull a much greater number of true positives at a similar false positive rate: at a false positive rate of 5%, the learners classify roughly 70% of the true positives. The MALDI-Spectrum Mill dataset was generated by selecting the top five hits from search results because 11% of the correct results were non-top ranking hits. It does not appear to be the case that the machine learning algorithms were able to effectively select these cases out, however (data not shown).

The Impact of Individual Attributes on the Final Prediction Accuracy—It is interesting to examine the relative importance of the various attributes used by the boosting and random forest algorithms to classify search results. The relative importance of each attribute is determined by noting nodes in the ensemble of trees in which the individual attribute appears and summing the relative information content or loss of entropy that each node containing the attribute provides. Attributes that provide the greatest combined discrimination among all the nodes in which it appears thus have a higher importance.

Fig. 3 displays the relative importance of each attribute from SEQUEST and Spectrum Mill search results using the boosting and random forest methods. Results for classification of the ESI-SEQUEST dataset incorporating the six attributes used by PeptideProphet are shown in Fig. 3*A*. All six attributes show a contribution to the discrimination with the most important contribution from the NTT variable. PeptideProphet incorporates only the first five attributes in the calculation of the discriminant function, introducing NTT distributions using a separate joint probability calculation. The coefficients for their discriminant score weight XCorr highest followed by ΔCn with much lower contributions due to delta M + H and Sp rank. Again length is used by PeptideProphet to correct for the well known peptide length dependence of the XCorr variable. Our results indicate a roughly equivalent contribution from ΔCn and XCorr with a significant contribution from Sp rank. Delta M + H and length showed a much more moderate contribution. The six attributes display a high importance when used in conjunction with the other nine attributes from groups III and IV as indicated in Fig. 3*B*. Of these additional nine, Sp score showed a surprising contribution as this scoring measure is rarely used for discrimination in popular usage. Also significant were the PeakMatchRatio measure and the MPF. For those attributes that are in common, these results are in agreement with Fisher's discriminant scores calculated by Anderson *et al.* (14) with the exception of delta M + H, which showed very little contribution using their SVM approach. The number of arginine and proline measures as well as parent charge, length, and number of matched peptides appear to provide very little discriminative value.

The NTT variable provides by far the most important contribution, particularly for the boosting approach, but is informative only for the non-enzyme-specific searches. The results above indicate, however, that the machine learning approaches perform quite well even in the absence of this variable. The relative importance of the other measures in the absence of this variable is shown in Fig. 3*C*. In this scenario, the ΔCn measure provides the most important contribution. A comparison of Fig. 3, *B* and *C*, suggests that in the absence of NTT, the C-term residue variable contributes much more significantly to the discrimination. As discussed above, although not as discriminatory as NTT, C-term residue does contain some of the same information and may be useful as a partial replacement for NTT in situations in which NTT is prohibitively time-consuming to obtain.

The relative importance of different attributes for the Spectrum Mill data are shown in Fig. 3*D*. Not surprisingly, rank and score features are very important for discrimination. It is interesting, however, that rank and score do not duplicate each other because the ranking of results is primarily based on score. One surprising feature is the importance of delta MH^+ in the final discrimination. This attribute is relatively unimportant in the ESI-SEQUEST results; this may demonstrate a difference between the Spectrum Mill and SEQUEST scoring schemes but can more likely be explained by the greater mass accuracy of the TOF/TOF instrument compared with the LCQ. The percent SPI attribute is analogous to the "fraction matched MSMS total ion count" variable published by Anderson *et al.* (14) and is one of the primary criteria used for judging the correctness of a hit in the Spectrum Mill package. Its low importance measure in the context of the other attributes is of interest. The backbone cleavage score and unused ion ratio attributes calculated by Spectrum Mill appear relatively less important in this context as well. Note that NTT was not calculated for the MALDI-Spectrum Mill dataset due to difficulties in performing a no-enzyme specificity search against the NCBInr human dataset using Spectrum Mill.

Unsupervised Learning and Generalization/Comparison with PeptideProphet—In general, machine learning defines two primary types of models to address the classification problem, generative approaches and discriminative approaches. Algorithms such as the boosting and random forests methods discussed in this study are discriminative, nonparametric approaches in that they do not rely on an explicit distribution of the data and model the posterior probability $p(y|x)$ directly (see Ref. 24 for a general description). A generative method assumes a model for the distributions of the attributes given the class label and learns the distribution $p(x|y)$ and then uses Bayes rule to infer the posterior probability and the classification rule. In effect, these models incorporate assumed prior information in the form of probability distributions for each of the classes being modeled and use these distributions to calculate the probability that a data point belongs to each class. PeptideProphet is a generative, parametric method, modeling correct identifications as a Gaussian distribution and incorrect identifications as a Gamma distribution. If this model fits the observed data well, *i.e.* the distributions describing the different classes in the problem accurately reflect the physical processes by which the data are generated, the generative approach works well even for a small amount of training data. On the other hand, if the data diverge from the modeled distributions in a significant way, classification errors proportional to the degree of divergence result. Therefore, although straightforward, the performance of the parameter-based generative approaches tends to be sensitive to the assumed model. For the peptide classification problem discussed in this study, there is little scientific evidence that supports a particular distribution assumption; one has to rely on past experience to make the decision. Discriminative approaches, on the other hand, are a less risky option in that they do not rely on knowledge of the distributions of classes of the data. They become increasingly safe, approaching optimality, as data size increases. Keller *et al.* (11, 35) demonstrate that, for their data, the distributions described in their mixture model fit the data well. Whether these distributions are appropriate for all types of instruments and MS search engines and whether they are optimal are research questions. It may be the case that the addition of new attributes, which alter the discriminant score and thus the shape of the score distribution, may be problematic for a generative tool. Due to their flexibility, our approaches are expected to generalize well to other types of data obtained from various instruments, search engines, and experimental conditions. We believe this is a particularly attractive feature.

The boosting/random forest methods are supervised approaches, relying on training data for their functionality. We note that an unsupervised learning component has been added to PeptideProphet. PeptideProphet uses training data to learn coefficients in the calculation of the discriminate score; it subsequently uses these scores to establish the basic shape of the probability distributions modeling correct and incorrect search hits as a function of parent peptide charge. For each unique dataset, when additional test data arrive, the distribution parameters are refined using an expectation maximization algorithm such that a refined classification procedure can be performed. This unsupervised component can function to compensate for a less-than-optimal fit of observed data to the model distributions. How to combine the unsupervised learning techniques such as clustering with out-of-the-box classification tools such as the ensemble tree methods discussed here is a challenge that needs to be addressed. Our approach provides a framework for performing the supervised aspect of the problem in a more general way using established out-of-the-box functionality. This approach can be coupled with an unsupervised component to provide more flexible functionality, assuming appropriate training datasets are available that match the input data. Here we have described one such dataset, potentially useful as a distributed resource for training purposes. The degree to which an individual training dataset provides adequate parameterization for a particular test set is an open question. Certainly training sets will need to be search algorithm-specific, and we intend to extend this work to other algorithms such as Mascot and X!Tandem in future work, but whether instrument-specific datasets are necessary is an area of investigation.

Having a tool that generates a truly accurate probability of correctness is an attractive feature of all algorithms in this domain. The fitness scores generated by the random forest method are probability estimates, and the boosting fitness scores can be directly converted into probability estimates via a logit transformation. True probability estimation is a difficult problem, however. It must be clearly noted that, as with other tools in this domain, these estimates must be considered approximate. Various methods have been developed for converting these types of scores into accurate probability estimates (33, 34). These methods are referred to as probability calibration methods in the literature. On the other hand, accurate classification of results does not require accurate probability estimates. For example, in a simple two-class classification setting such as ours, one may only need to know whether the probability is bigger or smaller than some selected value to achieve an accurate classification rule.

As a final note, the work described here addresses the problem of generating rankings and confidence measures for identification of peptides using mass spectrometry database search algorithms. This step is typically not the end goal of data analysis: the peptide measures can be combined to generate confidence measures for the presence of a protein in a sample. This problem of combining peptide results to generate accurate protein identifications has been addressed in other algorithms, such as ProteinProphet (35). Improvement in peptide identification will increase the sensitivity and specificity of downstream protein identifications, and the results from the algorithms described in this work can be used di-

rectly as inputs into protein level calculations.

Conclusions—In a production proteomics laboratory, researchers are often faced with the challenge of curating large lists of protein identifications based on various confidence statistics generated by the search engines. The common methodology for selecting true hits from false positives is based on thresholding. These approaches can lead to a large number of false positives (using a more promiscuous threshold) or a large number of false negatives (using a relatively stringent threshold). Machine learning approaches such as boosting and random forest methods provide a more accurate method for classification of the results of MS/MS search engines as either correct or incorrect. Additionally newer scoring criteria continue to be published that could improve the ability of automated tools to better discriminate true search results and can complement the standard scoring measures generated by popular search engines. Flexible methods that allow for accommodation of these new scoring measures are necessary to allow them to be easily incorporated into production use. Modern machine learning approaches such as the ensemble methods described here can perform very well out of the box with very little tuning. Improved results could very likely be obtained by tuning these tools to particular data sets, *i.e.* by making use of class prior probabilities to accommodate the imbalanced sizes of the correct and incorrect datasets. These approaches can additionally be used to generate measures of relative importance of scoring variables and may be useful in the development of new scoring approaches.

Acknowledgments—We thank all members of the National Resource for Proteomics and Pathways who contributed to this work, namely Angela Walker for TOF/TOF mass spectrometry, Donna Veine and John Strahler for sample processing of the purified Genway proteins, and Tom Blackwell and Jayson Falkner for invaluable feedback on the manuscript. We thank Andy Keller and Alexey Nesvizhskii for making the SEQUEST data available and for the open release of the PeptideProphet software. We also thank Daniela Eggle for contributions to the original support vector machine testing of the SEQUEST dataset.

* This work was supported in part by the National Resource for Proteomics and Pathways funded by National Center for Research Resources Grant P41 RR 18627-01 (to P. C. A). The costs of publication of this article were defrayed in part by the payment of page charges. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

We dedicate this work to the memory of Leo Breiman for outstanding contributions to the study of statistics and machine learning.

¶ To whom correspondence should be addressed: University of Michigan, 300 North Ingalls Bldg., Rm. 1196, Ann Arbor, MI 48109. Tel. and Fax: 734–647-0951; E-mail: pulintz@umich.edu.

REFERENCES

1. Eng, J., McCormack, A., and Yates, J. (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* **5,** 976–989
2. Perkins, D., Pappin, D., Creasy, D., and Cottrell, J. (1997) Probability-based protein identification by searching sequence databases using mass-spectrometry data. *Electrophoresis* **20,** 3551–3567
3. Clauser, K. R., Baker, P., and Burlingame A. L. (1999) Role of accurate mass measurement (±10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Anal. Chem.* **71,** 2871–2882
4. Bafna, V., and Edwards, N. (2001) SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics* **17,** Suppl. 1, S13–S21
5. Havilio, M., Haddad, Y., and Smilansky, Z. (2003) Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem.* **75,** 435–444
6. Craig, R., and Beavis, R. C. (2004), TANDEM: matching proteins with tandem mass spectra. *Bioinformatics* **20,** 1466–1467
7. Geer, L. Y., Markey, S. P., Kowalak, J. A., Wagner, L., Xu, M., Maynard, D. M., Yang, X., Shi, W., and Bryant, S. H. (2004) Open mass spectrometry search algorithm. *J. Proteome Res.* **3,** 958–964
8. Moore, R. E., Young, M. K., and Lee, T. D. (2002) Qscore: an algorithm for evaluating SEQUEST database search results. *J. Am. Soc. Mass Spectrom.* **13,** 378–386
9. MacCoss, M. J., Wu, C. C., and Yates, J. R., III (2002) Probability-based validation of protein identifications using a modified SEQUEST algorithm. *Anal. Chem.* **74,** 5593–5599
10. Zhang, N., Aebersold, R., and Schwikowski, B. (2002) ProbID: a probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics* **2,** 1406–1412
11. Keller, A., Nesvizhskii, A. I., Kolker, E., and Aebersold, R. (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* **74,** 5383–5392
12. Sadygov, R. G., and Yates, J. R., III (2003) A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal. Chem.* **75,** 3792–3798
13. Fenyo, D., and Beavis, R. C. (2003) A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.* **75,** 768–774
14. Anderson, D. C., Li, W., Payan, D. G., and Noble, W. S. (2003) A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *J. Proteome Res.* **2,** 137–146
15. Eriksson, J., and Fenyo, D. (2004) Probity: a protein identification algorithm with accurate assignment of the statistical significance of the results. *J. Proteome Res.* **3,** 32–36
16. Sun, W., Li, F., Wang, J., Zheng, D., and Gao, Y. (2004) AMASS: software for automatically validating the quality of MS/MS spectrum from SEQUEST results. *Mol. Cell. Proteomics* **3,** 1194–1199
17. Washburn, M. P., Wolters, D., and Yates, J. R., III (2001) Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat. Biotechnol.* **19,** 242–247
18. Peng, J., Elias, J. E., Thoreen, C. C., Licklider, L. J., and Gygi, S. P. (2003) Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. *J. Proteome Res.* **2,** 43–50
19. Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) Maximum likelihood from incomplete data via EM algorithm. *J. R. Stat. Soc. Series B* **39,** 1–38
20. Graumann, J., Dunipace, L. A., Seol, J. H., McDonald, W. H., Yates, J. R., III, Wold, B. J., and Deshaies, R. J. (2004) Applicability of tandem affinity purification MudPIT to pathway proteomics in yeast. *Mol. Cell. Proteomics* **3,** 226–237
21. Hastie, T., Tibshirani, R., and Friedman, J. H. (2001) *Elements of Statistical Learning*, Springer, New York
22. Freund, Y., and Schapire, R. (Vitany, P., ed.) (1995) A decision theoretic generalization of on-line learning and an application to boosting, in *Proceedings of the 2nd European Conference on Computational Learning Theory, Barcelona, Spain (March 13–15, 1995)* pp. 23–37, Springer, New York
23. Breiman, L. (2001) Random forests. *Machine Learning* **45,** 5–32
24. Vapnik, V. N. (1999) *The Nature of Statistical Learning Theory*, pp. 138–167, Springer, New York
25. Jaakkola, T., Diekhans, M., and Haussler, D. (1999) Using the Fisher kernel method to detect remote protein homologies. *Proc. Seventh Int. Conf. Intell. Syst. Mol. Bio.* 149–158, AAAI Press, Menlo Park, CA
26. Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16,** 906–914

27. Brown M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., Jr., and Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. U. S. A.* **97,** 262–267
28. Strahler, J. R., Veine, D., Walker, A., Kachman, M., Ulintz, P., and Falkner, J. (2005) A publicly available dataset of MALDI-TOF/TOF mass spectra of known proteins, in *53rd American Society for Mass Spectrometry Conference on Mass Spectrometry and Allied Topics, San Antonio, Texas (June 5–9, 2005)*, Abstr. TP22-398, American Society for Mass Spectrometry, Santa Fe, NM
29. Keller, A., Purvine, S., Nesvizhskii, A. I., Stolyar, S., Goodlett, D. R., and Kolker, E. (2002) Experimental protein mixture for validating tandem mass spectral analysis. *OMICS* **6,** 207–212
30. Wysocki, V. H., Tsaprailis, G., Smith, L. L., and Breci, L.A. (2000) Mobile and localized protons: a framework for understanding peptide dissociation. *J. Mass Spectrom.* **35,** 1399–1406
31. Kapp, E. A., Schutz, F., Reid, G. E., Eddes, J. S., Moritz, R. L., O'Hair, R. A., Speed, T. P., and Simpson, R. J. (2003) Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. *Anal. Chem.* **75,** 6251–6264
32. Tabb D. L., Huang, Y., Wysocki, V. H., and Yates, J. R., III (2004) Influence of basic residue content on fragment ion peak intensities in low-energy collision-induced dissociation spectra of peptides. *Anal. Chem.* **76,** 1243–1248
33. Platt, J. C. (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in *Advances in Large Margin Classifiers* (Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., eds) pp. 61–74, MIT Press, Cambridge, MA
34. Caruana, R., Niculescu, S., Rao, B., and Simms, C. (2003) Evaluating the C-section rate of different physician practices: using machine learning to model standard practice, in *Proceedings of the Annual Conference of the American Medical Informatics Association, Washington, D. C. (November 8–12, 2003)*, American Medical Informatics Association, Bethesda, MD
35. Nesvizhskii, A. I., Keller, A., Kolker, E., and Aebersold, R. (2003) A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* **75,** 4646–4658
36. Link, A. J., Eng, J., Schieltz, D. M., Carmack, E., Mize, G. J., Morris, D. R., Garvik, B. M., and Yates, J. R., III (1999) Direct analysis of protein complexes using mass spectrometry. *Nat. Biotechnol.* **17,** 676–682