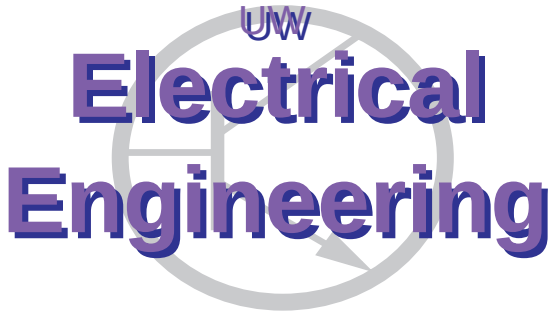
Filtering tandem mass spectra for quality

Sergey Feldman, Barbara Frewen, Michael J. MacCoss, and Maya R. Gupta
Departments of Electrical Engineering and Genome Sciences
University of Washington
Seattle, WA 98195



UWEE Technical Report
Number UWEETR-2012-0001
January 2012

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Abstract

Accurate protein and peptide identifications by database search depend on the quality of the mass spectrometer spectra. Excessive quantities of low quality spectra consume valuable computing resources and can decrease overall accuracy of peptide and protein identifications. We present a fast spectrum quality filter called French Press that can remove low quality spectra without database searching. The filter's speed is the result of a tuned random forest classifier and a greedily optimized classification feature subset, culled from features appearing in prior research on spectrum filtering and modeling. Results on diverse data sets of mass spectrometer runs show that the filter can remove roughly 50% of low quality spectra while retaining 99% of identifiable spectra.

Background

Tandem mass spectrometry (MS/MS) has become the primary tool for the identification of peptides within a complex protein mixture. Most proteomics laboratories first digest a complex protein mixture into peptides, then analyze these samples by microcapillary chromatography-tandem mass spectrometry (μLC-MS/MS) [29]. In these analyses, peptides are separated by reverse phase chromatography and are electrosprayed into the mass spectrometer where the instrument measures the peptide m/z (mass-to-charge ratio) and makes decisions on which peptides to select for MS/MS using data-dependent acquisition [28]. Computational approaches are then used to assign the resulting MS/MS spectra to peptide sequences and assemble the peptides back into a parsimonious list of proteins. This process is commonly referred to as "shotgun proteomics" [16, 18, 30, 34].

Modern advances in mass spectrometry [17, 23, 27] have facilitated the acquisition of MS/MS spectra at an astonishing rate, with improvements likely to continue in the future. Acquisition rates exceeding 10 Hz are now possible and result in tens of thousands of MS/MS spectra per hour that need to be processed to yield peptide and protein identifications. Data collection at such rapid rates is promising because it increases the number of peptides sampled, but it also places an immense computational burden on the laboratory.

While the development of faster scanning mass spectrometers improves the number of peptide identifications, an overwhelming majority of the MS/MS spectra still remain unassigned to a peptide sequence. Reasons for failing to identify a peptide from an MS/MS spectrum include (1) the peptide belongs to a sequence that is not accounted for by the "reference" sequence, (2) the peptide contains an unanticipated post-translational modification, (3) the spectrum is a chimera from multiple peptide sequences, (4) the spectrum is of poor quality (noisy or corrupted), or (5) the spectrum is of a non-peptide molecular species. We and others have shown that even for well-annotated model organisms a surprisingly large number of peptides can be identified from regions of the genome that are not annotated as protein coding [19]. Because of the importance of identifying spectra that belong to peptides containing: post-translational modifications, polymorphisms, or sequences from unannotated regions of the genome, we would like to be able to *efficiently filter out spectra of poor quality while retaining a high percentage of spectra that contain biologically interesting information.*

In pursuit of this goal, we have developed a computationally efficient classification filter termed French Press that discriminates between high and low quality MS/MS spectra prior to database searching. (Research-grade code is available at: http://students.washington.edu/sergeyf/FrenchPress.zip.) By removing the low quality MS/MS spectra, one can greatly reduce the total computational overhead of the database search, despite the additional overhead of the classifier. Also, by removing spectra that are unlikely to result in confident peptide identifications one can improve the discrimination between true and false peptide spectrum matches because of the reduction in the number of hypothesis tests that are performed without reducing the number of correct peptide spectrum matches. French Press can be used to discover MS/MS spectra that are of high quality but remain uninterpretable, after which more expensive computational methods can be devoted to this remaining subset to identify potential unanticipated modifications.

A number of other researchers have investigated the problem of pre-filtering [1, 7, 15, 35]. Also related is research into post-filtering to discover high-quality spectra previously unmatched by a protein database search (with SEQUEST [6], for instance) [15, 22, 32]. Both pre-filtering and post-filtering research has varied in what is used as training data, normalization schemes, the choice of features, the type of classifier, how

Table 1: Summary of Prior Work

Paper	Feature Type(s)	Classifier(s)	Spectrum Classification Problem
Bern et al. [1]	HC, PPD-Hist	QDA, SVM	score above threshold vs. below (SQ)
Ding et al. [5]	HC	WKM	score clustering (SQ)
Flikka et al. [7]	HC, PPD-Hist	ensemble	score above threshold vs. below (MT, MV)
Klammer et al. [14]	HC, PL-Hist	SVM	charge state determination
Koenig et al. [15]	HC, PPD-Hist	RF, SVM	score regression (MT)
Moore et al. [20]	HC	linear	score above threshold vs. below (SQ)
Na and Paek [21]	HC	SVM	score above threshold vs. below (SQ, MV)
Nesvizhskii et al. [22]	HC, PPD-Hist, ST	linear	score above threshold vs. below (PP, SQ)
Purvine et al. [24]	HC	threshold	high spectra quality vs. low (hand-specified)
Salmi et al. [26]	HC	DT, RF	score above threshold vs. below (SQ, ProlCAT, MV)
Wong et al. [31]	HC	logistic regression	score above threshold (PP, ProteinProphet, SQ) or p-value below threshold (pepNovo, SPIDER) or p-value below threshold (InsPecT) vs. not
Wu et al. [32]	HC	LDA	score above threshold vs. below (PP, SQ)
Xu et al. [33]	HC	quadratic	score above threshold vs. below (MT)
Zou et al. [35]	HC	SVM	score above threshold vs. below (PP, SQ)

Table 2: Legend for Table 1

DT	decision tree	PP	PeptideProphet
LDA	Linear Discriminant Analysis	PPD-Hist	pairwise peak differences histogram
HC	hand-crafted	RF	Random Forest [2]
MT	MASCOT	SQ	SEQUEST [6]
MV	manually verified	ST	sequence tag features
PL-Hist	peak locations histogram	WKM	weighted K-Means

much training data was used to form the filter, how much test data was used to evaluate performance, and how well the filter was evaluated on as-yet unseen and diverse data. Details of prior work are summarized in Tables 1 and 2, and will be discussed further where relevant. This paper builds on prior work's exploration of possible features and normalization schemes.

Methods

We first describe the data, then the class labels, and then the features.

Data

To represent a range of experimental conditions and the variations in spectra and spectra quality they produce, we selected runs from a number of source organisms analyzed under varying conditions. All experiments share the following features: proteins were extracted using standard methods and digested to peptides with trypsin; peptides were separated using reverse-phase chromatography and introduced into the instrument by electrospray; and spectra were acquired on the LTQ-FT Ultra Hybrid Mass Spectrometer (Thermo Fisher Scientific) using data-dependent acquisition. We used five of the MS/MS runs (run IDs 21 through 25 in Table 3, 51564 spectra) for the initial model selection and to run *independent and identically*

Table 3: Run IDs and Corresponding Data

Run ID	Data type
1	C. elegans daf-2/daf-16 mutant
2	C. elegans daf-2 mutant
3	S. cerevisiae
4	S. cerevisiae
5	Mouse Heart Mitochondria
6	Mouse Heart Mitochondria
7	Mouse Heart Mitochondria
8	Michrom 6 Protein Bovine Mixture
9	Media from Leishmania culture
10	Mouse Heart Mitochondria
11	Mouse Heart Mitochondria
12	Human P7 Nuclear Preparation
13	Human BJ Tert Nuclear Preparation
14	Human CACO Nuclear Preparation
15	Human Hep G2 Nuclear Preparation
16	Human K562 Nuclear Preparation
17	Human SKNSH Nuclear Preparation
18	Human GM12878 Nuclear Preparation
19	Human Hela Nuclear Preparation
20	Human HL60 Nuclear Preparation
21	C. elegans glp-4 mutant
22	C. elegans wild-type strain N2
23	S. cerevisiae
24	S. cerevisiae
25	S. cerevisiae strain S288c

distributed (IID) experiments. We used the additional 20 MS/MS runs (run IDs 1 through 20 in Table 3, 147346 spectra) to further validate the selected classifier model with non-IID experiments. The 25 runs are detailed in Table 3.

For our training data we took advantage of the high-resolution MS1 scans to measure a specific aspect of spectrum quality: the software program Hardklor [10] analyzed each region of the MS1 scans that was isolated for fragmention and determined if there was a peptide-like isotope distribution present. All MS/MS spectra were searched with SEQUEST [6] and q-values for each peptide spectrum match were computed by Percolator [11]. (A q-value is the minimum false discovery rate at which a particular spectrum-protein pair will appear in the list of matches that results after database searching for a given dataset [13].)

Class Assignment and Data Partitioning

A key question is how to define 'good' and 'bad' quality spectra (also referred to as positives and negatives) for the purpose of training a classification filter. Prior work relating to this question is detailed in the right-most column of Table 1. The standard approaches rely on database search scores and/or manual validation (in Table 1, 'Score above threshold vs. below' indicates that the classes were defined using the output scores of the relevant protein matching database such as SEQUEST). In contrast to prior work and motivated by the fact that a spectrum can receive a low score from a database search for reasons other than the quality of the spectrum (e.g. post-translational modifications, sequence polymorphism), our definition of 'bad' spectra is unrelated to the search score: we define 'bad' spectra to be those that are determined (by Hardklor [10]) to have no peptide isotope distribution in the MS1 scan. 'Good' spectra are those that have exactly one peptide isotope distribution as well as a q-value less than or equal to 0.01. Note that many spectra have an

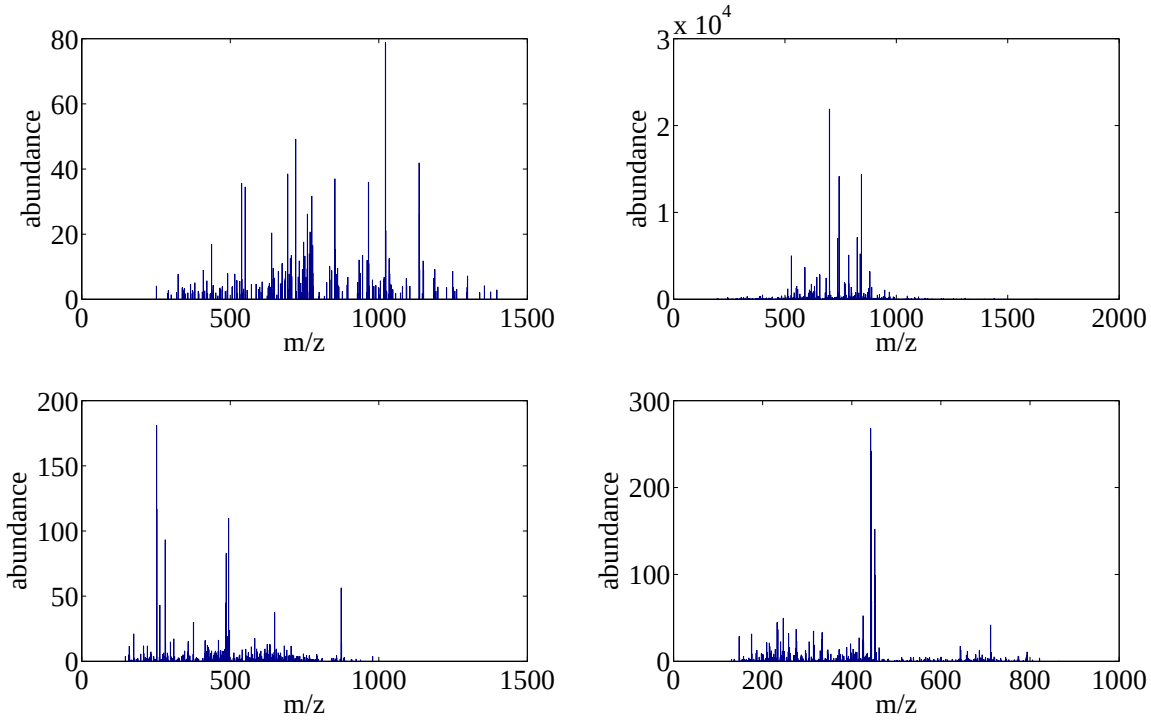


Figure 1: Examples of 'good' (above) and 'bad' (below) spectra.

MS1 peptide isotope distribution, but a q-value larger than 0.01; we refer to these as 'intermediate' spectra. Some randomly chosen example 'good' and 'bad' spectra are shown in Figure 1. The top two spectra are examples of our definition of 'good' spectra, with charges of +2 and +5, respectively. The bottom two spectra are examples of our definition of 'bad' spectra.

For development, validation, and training purposes, we worked with only five runs of data (IDs 21 through 25). First, we split these 51,564 spectra randomly in half, and all training was done on one half, with the other half reserved for evaluating the model once it was fully trained. To train (selecting features and parameters), we used randomized cross-validation (CV): we randomly split the training data into 50% CV training sets and 50% CV test sets ten times.

Features

We surveyed and categorized the set of features that have been proposed for spectrum filtering as described in Table 1, and implemented representative features from within each category for a total of 109 features considered. Only representative features were implemented because many of the features in literatures are nearly identical, except for minor and insignificant differences. Some researchers have taken an exhaustive approach by attempting to explicitly encode as much knowledge as possible, creating features like *pairwise peak difference histograms* and *peak locations histograms*. Other features were *hand-crafted*; either inspired by the expert visual intuition of the referenced authors and their associates (e.g. peak density), or guided by the physics of mass spectrometry (e.g. Bern's Good-Diff Fraction [1]). Prior work deals almost exclusively with ions of charges +1, +2, and +3. However, there is a need to have an algorithm flexible enough to handle data containing ions of higher charges, and, thus, some of our features are natural extensions of those found in literature to charges +4 and +5.

Let F_i be the ith feature. The first 26 features (F_{1-26}) we investigated are detailed in Table 5, and assume a given spectrum triplet (x, y, m), where $x \in \mathbb{R}^N$ is a N-length sorted vector of mass-to-charge peak

Table 4: Notation

N_k	number of peaks in kth spectrum
$x_k \in \mathbb{R}^{N_k}$	sorted peak intensity locations of kth spectrum (m/z)
$y_k \in \mathbb{R}^{N_k}$	peak heights of kth spectrum
$y_k[i]$	ith peak intensity of the kth spectrum
$\hat{y}_k$	normalized peak heights (see the section on normalization for more details)
m_k	precursor mass-to-charge of kth spectrum (m/z)
$TP(r)$	true positive rate at false negative rate r
I_Q	indicator function; equal to 1 if Q is true, 0 otherwise
$\Delta x_k \in \mathbb{R}^{N_k-1}$	discrete derivative of x_k, i.e. $\Delta x_k[i] = x_k[i+1] - x_k[i]$ for $i \in \{1, \ldots, N_k - 1\}$
$D_k \in \mathbb{R}^{N_k \times N_k}$	matrix of pairwise absolute peak location distances, i.e. $D_k[i,j] = \lvert x_k[i] - x_k[j] \rvert$
$\vec{D}_k \in \mathbb{R}^{N_k(N_k-1)/2}$	vectorized version of D_k, retaining only the upper triangular
$\mathbf{prct}(y,p)$	p% percentile of vector y; e.g. $\mathbf{prct}(y,50) = \mathbf{median}(y)$
$\text{rank}(y_k[i])$	rank; e.g. $\text{rank}(y_k[i]) = n$ if $y_k[i]$ is the nth largest peak intensity

locations, $y \in \mathbb{R}^N$ is a vector of corresponding abundances, and $m \in \mathbb{R}$ is the mass-to-charge ratio of the precursor ion. All summations are from 1 to N (unless otherwise noted). For notation, see Table 4.

The next 50 features (F_{27-87}) are a generalization of a subset of features from Klammer et al. [14] to determine charge states of peptides. These features rely on the intuition that, in good quality spectra, precursor ions tend to fragment into pairs of ions, of which the individual masses and individual charges sum to the total mass and the total charge of the precursor ion, respectively. For example: if the precursor ion had mass 1035 and charge +5, then one of the ways it could fragment would be into two ions, with masses 460 and 575, and charges +2 and +3, respectively. Klammer et al. only consider charges of +1, +2, and +3. Our data, however, also contains charges of +4 and +5, and so we expanded the features to cover these cases. We implemented features from equations numbered (1)-(7) in Klammer's paper, as well as the final set of features described after (but not including) equation (11). For full descriptions of these features we refer the reader to Klammer's paper. As an example, consider

$$F_{27} = \sum_{i=1}^{km} S[i]S[2m - i],$$

where S is a surrogate spectrum such that $S[i]$ is the sum of all spectrum peaks within 0.5 m/z of i, for $i \in \{1 \ldots, km\}$ (the value of m is rounded to the nearest integer), and k, an integer, is a user-defined parameter. We recommend setting k to the maximum ion charge that is expected to appear in the training and test data. The reason for this is that the maximum theoretic mass of any fragment should be no larger than km. We set k to be 5 because that was the maximum charge of the precursors in our training data. Feature F_{27} should theoretically be large for +2 and +4 ions as they tend to be symmetric around the precursor mass-to-charge ratio m. The rest of the features are analogously extended to larger ion charges, and also extended in an attempt to encode various neutral losses.

Features F_{63-87} are designed to capture the distribution of the signal intensity in the spectrum across the m/z range. To do so, we bin the peak intensities into regions, the sizes of which are relative to the precursor m/z m. This feature set makes no distribution assumptions. For these features, we also round m to the nearest integer. The features are:

$$F_{62+j} = \frac{\sum_{i=m\frac{j-1}{l}+1}^{m\frac{j}{l}} S[i]}{\sum_{i=1}^{km} S[i]}, \quad j \in \{1, \ldots, kl\},$$

Table 5: Hand-crafted Features

$F_1 = N$	number of raw peaks
$F_2 = \sum y[i]$	TIC (total ion current)
$F_3 = \frac{F_2}{F_1}$	mean raw peak intensity
$F_4 = \mathbf{mean}(\hat{y})$	mean normalized peak intensity
$F_5 = \sigma(\hat{y})$	stand. dev. of the peak intensity
$F_{6-10} = \mathbf{prct}(\hat{y}, \beta)$	βth percentiles of the normalized peak intensity $\beta = \{5, 25, 50, 75, 95\}$
$F_{11-13} = \frac{1}{N} \sum I_{\hat{y}[i] > \alpha \sum \hat{y}[i]}$	fraction of peak intensities above α times normalized TIC
$F_{14-16} = \frac{1}{N} \sum I_{\hat{y}[i] > \alpha \max_i(\hat{y}[i])}$	fraction of peak intensities above α times the max peak $\alpha = \{0.01, 0.10, 0.50\}$
$F_{17} = \frac{1}{N} \sum I_{x[i] \geq m}$	fraction of peak locations above precursor
$F_{18} = \max_i(x[i]) - \min_i(x[i])$	peak location range
$F_{19} = N / (\max_i(x[i]) - \min_i(x[i]))$	peak location density
$F_{20} = \sum \hat{y}[i] / (\max_i(x[i]) - \min_i(x[i]))$	TIC density
$F_{21} = \mathbf{mean}(\Delta x)$	mean of successive peak location differences
$F_{22} = \sigma(\Delta x)$	stand. dev. of successive peak location differences
$F_{23} = m$	precursor m/z
$F_{24} = \mathbf{mean}(\vec{D})$	mean distance between peaks
$F_{25} = \sigma(\vec{D})$	stand. dev. of the distances between peaks
$F_{26} = \mathbf{median}(\vec{D})$	median distance between peaks

where l captures the granularity of the histogram, and is another user-defined parameter; we set l to be 5, as suggested by Klammer et al. [14]. After calculating features F_{63-87}, they are normalized to sum to 1. This is done to turn the features into a probability distribution and make them independent of the number of peaks.

Features F_{88-98} and F_{99-109} are unweighted and weighted normalized histograms, respectively, of pairwise peak location differences $\vec{D}$. In our data, peak location differences are never larger than 2048 m/z, and, we divide the entire range from 1 to 2048 m/z into bins whose sizes increase as powers of 2: the first bin is from 1 to 2 m/z, the second bin is from 3 to 4 m/z, the third bin is from 5 to 8 m/z, the fourth from 9 to 16 and so on until the last bin which is from 1025 to 2048 m/z. The non-uniform bin sizes were inspired by the fact that the majority of meaningful peak pair differences are expected to be below $200m/z$, as that is the range within which amino acid weights lie. Thus, we wanted finer grain histograms in the lower end of the m/z range, while keeping the total number of bins relatively small. The histograms are calculated as follows:

$$F_{87+i} = \frac{\sum_{h,j}\{1 \mid x[h] - x[j] \in [2^{i-1}+1, 2^i]\}}{\sum_{j=1}^{11} F_{87+j}}, \quad i \in \{1, \ldots, 11\},$$

and

$$F_{98+i} = \frac{\sum_{h,j}\{\min(y[h], y[j]) \mid x[h] - x[j] \in [2^{i-1}+1, 2^i]\}}{\sum_{j=1}^{11} F_{98+j}}, \quad i \in \{1, \ldots, 11\}.$$

The min (instead of a fixed constant of 1) in the second expression is inspired by the results in [1]. Feature sets F_{88-98} and F_{99-109} are, as shown above, normalized to sum to 1.

Results and Discussion

In this section we discuss the algorithm and then both IID and non-IID tests.

Algorithm

We first describe the classifier, then the performance metric, then discuss pre-processing normalization of the spectra, followed by our feature selection process and parameter selection.

Classifier

We chose to use Breiman and Cutler's *random forest* (RF) classifier [2,9], due to its speed of evaluation at test (and relative speed of training), its state-of-the-art classification performance, its robustness to parameter choices, and the ease of implementation into an efficient end-user package. The RF composed of a total of B independent decision trees, where B is user-set parameter. To grow the bth tree, the following steps are taken:

1. Draw a bootstrap sample from the training data (that is, take a subsample of the data with replacement).
2. Grow the bth decision tree by recursively repeating the following steps, until the minimum node size is reached):
   (a) Select n variables at random from the d-dimensional feature space.
   (b) Pick the best split among them by minimizing node impurity (measured by Gini index [9]).
   (c) Split the node into two child nodes.

A test sample's class is then chosen to be the one that received the most votes amongst the B trees. We use the default bootstrap subsample parameter, so the RF parameters to select are the number of trees to grow B, and the number of feature dimensions n to be randomly selected at each split. Initially, we set

Table 6: Normalization Schemes

$\hat{y}[i] = y[i]$	no normalization (none)
$\hat{y}[i] = \sqrt{y[i]}$	square root of peaks (sqrt)
$\hat{y}[i] = \log(y[i])$	log of peaks (log)
$\hat{y}[i] = \frac{1}{\text{rank}(y[i])}$	inverse peak rank (inv rank)
$\hat{y}[i] = \frac{y[i]}{\sum_j y[j]}$	peaks divided by total ion current (raw/TIC)
$\hat{y}[i] = \frac{y[i]}{\max_j(y[j])}$	peaks divided by base peak intensity (raw/max)
$\hat{y}[i] = \frac{\sum_j y[j] I_{\text{rank}(y[j]) \geq i)}}{\sum_j y[j]}$	cumulative intensity normalization from Na and Paek [21] (cumul)

$B = 100$ features and $n = 30$ trees, which were the smallest values required to obtain good performance on some preliminary experiments. With these fixed parameter values, we then (a) chose a method with which to normalize the raw spectra based on cross-validated accuracy, and (b) performed optimal feature subset selection using a greedy exhaustive search (see subsequent sections for details). As detailed below, we then froze the normalization scheme and feature subset, and lastly cross-validated the RF parameters B and n for the final classifier.

Performance Metric

To train a normalization scheme, feature subset, and parameters B and n, we needed a single performance metric. Since the goal is to filter out as many true negatives (bad spectra) as possible at high rates of true positives (good spectra), the leftmost portion of the receiver operating characteristic (ROC) curve is not of interest. Thus to train the classifier we used a modified area under the ROC ($AUROC$) curve score, what we refer to as $AUROC95$. The standard $AUROC$ is defined as

$$AUROC = \int_{r=0}^{1} TP(r)dr.$$

Using the indicator function I_A, our $AUROC95$ is defined as

$$AUROC95 = \int_{r=0}^{1} TP(r)I_{TP(r)>0.95}dr.$$

See Figure 2 for an illustrative example: the area of the shaded region is the AUROC95 score, and takes values from 0 to 1. Intuitively, it is the fraction of 'bad' spectra filtered out while retaining 0.95 of the 'good' spectra.

Normalization

Nearly every algorithm listed in Table 1 has used some normalization scheme. Normalization schemes that are designed to address uninformative variability (noise) in spectrum peak heights are numerous and ad hoc. Because many of our features (detailed in the next section) are based on raw peaks, normalization has a large effect on the features. We summarize the normalization methods we found and tested in Table 6, for a spectrum triplet (x, y, m). To compare normalization schemes, we implemented each of the approaches detailed in Table 6, extracted the entire set of 109 features (see next section), and compared the cross-validation $AUROC95$ score, averaged across 10 splits. RF parameters were fixed, as noted above, at $B = 100$ trees and $n = 30$ randomly chosen dimensions per tree. The results averaged over the 10 CV splits are shown in Table 7, for both $AUROC95$ and standard $AUROC$ scores. Surprisingly, not normalizing yields better performance than any individual normalization scheme according to both metrics. Because of these findings, all of the results reported for the rest of this study are for features extracted from unnormalized spectra.

Figure 2: An example of an $AUROC95$ metric calculation.

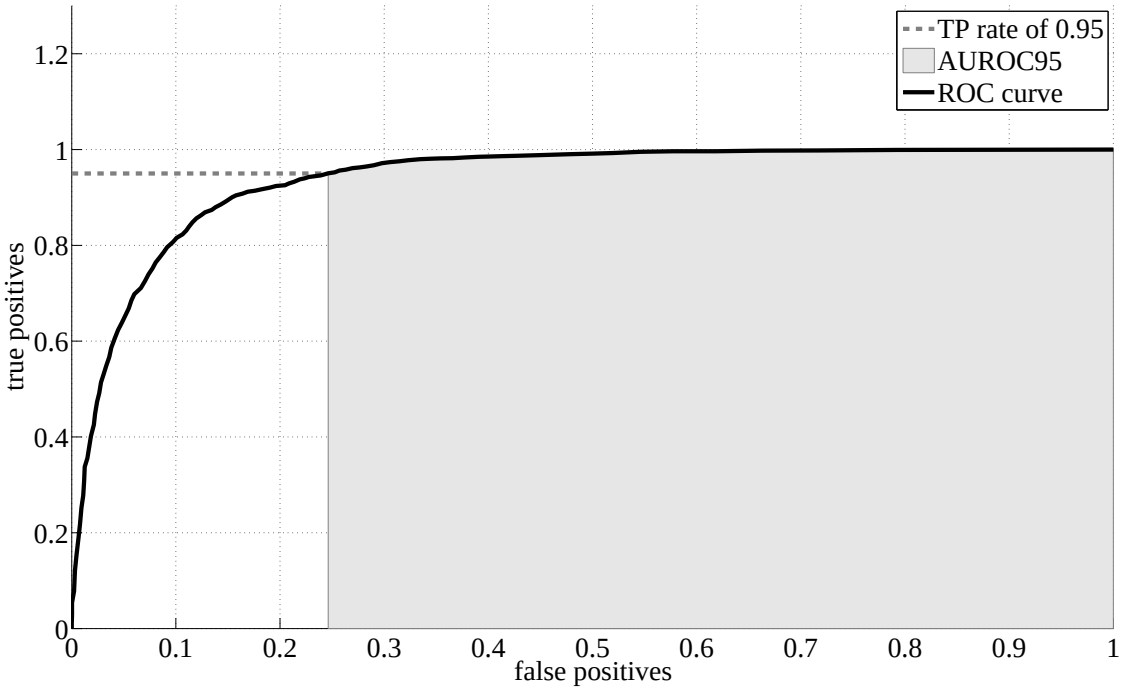
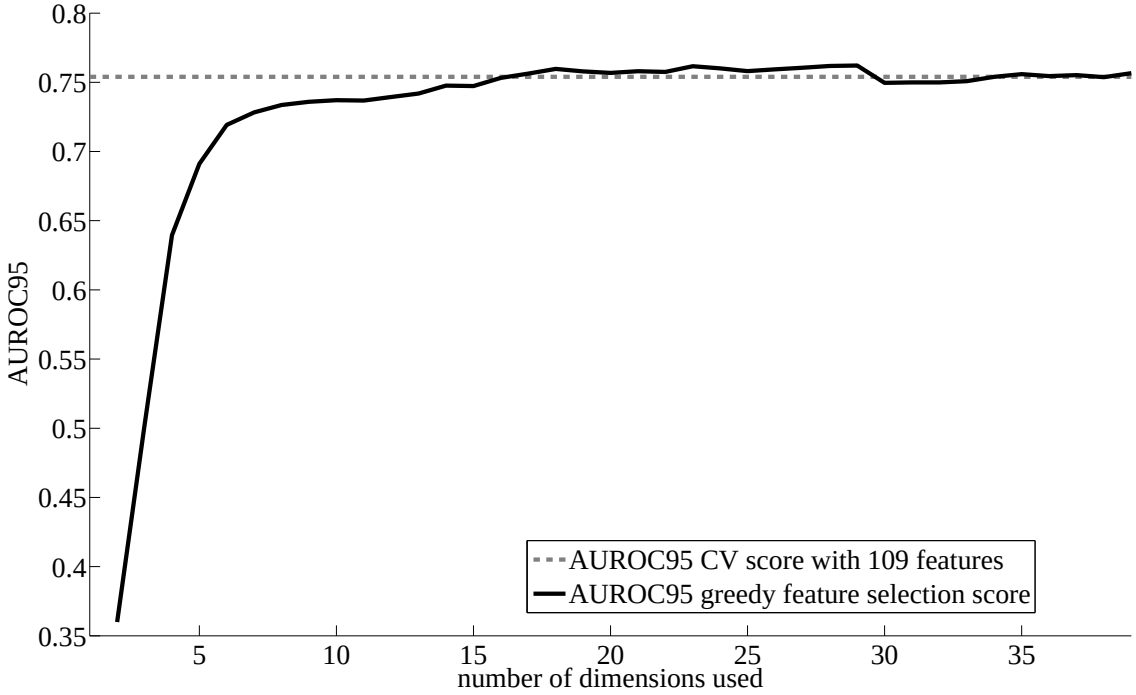


Table 7: Normalization Classification Results: Mean (and standard deviation) AUROC95 scores for various normalization schemes.

Normalization type	mean (std) $AUROC$	Mean (std) $AUROC95$
none	.941 (.003)	.754 (.015)
sqrt	.935 (.003)	.732 (.014)
log	.931 (.003)	.719 (.012)
inv rank	.924 (.004)	.711 (.019)
raw/TIC	.935 (.003)	.736 (.013)
raw/max	.933 (.003)	.725 (.014)
cumul	.932 (.003)	.724 (.014)

Figure 3: Greedy Feature Selection Results



Feature Selection

Classification with RF is fast, so a large portion of the running time of the French Press algorithm is devoted to feature extraction. Thus, to achieve the goal of computationally efficient spectrum filtering, we reduced the 109 considered features to a small, speedy subset using a greedy exhaustive search, similar to forward-stepwise selection [9]. Feature reduction can also increase the accuracy of machine learning methods [9].

We began with feature F_1, the number of raw peaks. Then, at the Nth iteration of our greedy feature search, there were N features in the feature set, and we compared the mean AUROC95 CV score that would result if one of the remaining $109 - N$ features was added. The remaining feature that yielded the lowest mean AUROC95 CV score was then added to the feature set, and this process was repeated.

Despite being greedy, this method is quite computationally intensive. If the total number of features is d, and we are performing n-fold cross-validation, then the random forest has to be run $O(nd^2)$ times. On a standard 2.6 GHz PC, the feature selection process took approximately 2 weeks of continuous computation.

The results are shown in Figure 3. The mean AUROC95 CV score is practically maximized with only the first 18 features chosen by the greedy search, which were (in order),

$$1, 58, 8, 5, 23, 68, 106, 98, 10, 71, 99, 35, 70, 9, 73, 65, 101, 51.$$

We chose these 18 features for our final classifier.

The first chosen feature, F_{58}, was extrapolated from features in Klammer et al. [14]:

$$F_{58} = \frac{\sum_{i=1}^{m-1} S[i] - \sum_{i=m+1}^{4m} S[i]}{\sum_i S[i]},$$

and is one of a number of features designed to capture information about +4 ions (in this case, the feature value would be small if the ion broke into a +1 ion and a +3 ion).

Table 8: Random Forest Parameter Selection Scores: Mean (and standard deviation) AUROC95 Scores for various numbers of trees (columns) and numbers of features per tree (rows).

	100 trees	200 trees	300 trees	400 trees
3 features	.755 (.011)	.760 (.011)	.760 (.011)	.762 (.011)
5 features	.758 (.010)	.761 (.012)	.762 (.011)	.765 (.010)
10 features	.753 (.010)	.757 (.013)	.757 (.012)	.757 (.011)
15 features	.745 (.012)	.750 (.012)	.752 (.012)	.752 (.014)

Features $F_{8,9,10}$ are three out of the five peak height percentile features. $F_8 = \mathbf{prct}(y, 50)$ accounts for the median peak height, and $F_{10} = \mathbf{prct}(y, 95)$ is a robust max. The fact that these features have good discriminatory power seems to indicate that the peak height variability, which is usually considered a hindrance requiring normalization, is in fact an informative indicator of spectrum quality. Similarly with $F_{23} = m$; even though our training data contains a large number of different precursors, the precursor mass-to-charge ratio is by itself a good indicator of spectrum quality.

Features $F_{65,68,70,71,73}$ are part of a set of features that attempts to capture the distribution of peak locations relative to the precursor mass-to-charge ratio. For our choices of k and l, there were 25 total features in this set. These 25 features split the m/z range into bins of equal widths, the sizes of which depend on the precursor m/z. The chosen features are the 3rd, 6th 8th, 9th, and 11th bins.

While chosen features $F_{35,51,58}$ make assumptions about the distribution of the peak heights (dependent on the charge of the precursor ion as well as on the neutral loss of 18 Da), $F_{65,68,70,71,73}$ are free of such assumptions. To calculate these features one must actually compute all of the features in the range F_{63-87}, because this set of features is normalized to sum to one.

F_{98} is the last bin of the weighted peak location difference histogram, from 1024 to 2048 m/z. F_{98} captures information about the relative proportion of peaks that are significantly far apart. $F_{99,101,106}$ are bins from the unweighted peak location difference histogram, covering ranges of 1 to 2 m/z, 5 to 8 m/z, and 129 to 256 m/z, respectively. Similarly, they capture information about how often peaks are in close and mid-range proximity.

Parameter Selection

Having settled on a normalization type and feature subset, we proceed to parameter selection for the random forest. As in all of the above experiments, we used 10-fold cross-validation to find the best set of parameters. The number of trees B was chosen from the parameter set $\{100, 200, 300, 400\}$, and the number of randomly chosen features per tree n was chosen from the parameter set $\{3, 5, 10, 15\}$. Mean (and standard deviation) AUROC95 scores for a range of random forest parameters are shown in Table 8. (All results are reported for the reduced feature space with 18 features.)

Since we are concerned with algorithmic speed as well as accuracy, it is beneficial to set B and n to the smallest necessary values. Our results show that using more than $B = 100$ trees and more than $n = 5$ features per tree offers only incremental improvements Therefore, we set the number of trees to 100 and the number of randomly chosen features per tree to 5.

Testing

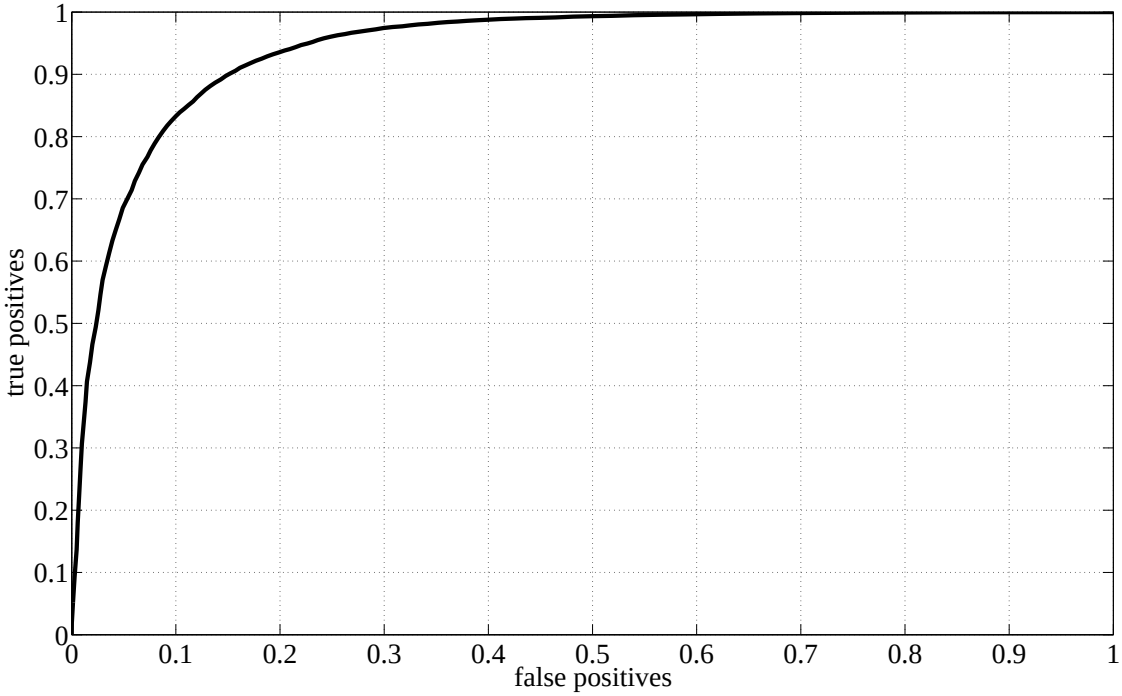
Having settled on a normalization scheme, subset of features, and classifier parameters, we retrained by CV the random forest using all of the training data (a random half of the spectra in five MS/MS runs), and then classified the as-yet-untouched test data (the remaining random half). On the test data we obtained an AUROC95 score of .770, which is very close to the CV score. We hypothesize that the slight improvement is due to the increased size of the training data compared to the CV experiments, where, for each of the ten splits, only half of the data was used to train the RF.

The magnitude of the AUROC95 scores does not carry a great deal of interpretable information. To gain some intuition about our results we first examine exactly how the AUROC95 score translates into more

Table 9: Relevant Points and Thresholds from the ROC Curve

Threshold	TP	TN	Fraction 'intermediate' filtered out
.51	.901	.854	.542
.37	.953	.776	.432
.28	.971	.712	.362
.15	.990	.587	.257

Figure 4: ROC Curve for the Test Data



familiar values: true positives (TPs) and true negatives (TNs). We are interested in high rates of TPs. Consider Table 9, which contains a few points of interest from the ROC curve in Figure 4, as well as their associated thresholds. The threshold values are votes that result from the RF algorithm, normalized to sum to 1; samples above the threshold are classified as 'good', and below or equal to the threshold are classified as 'bad'. For example, if the user wishes to retain approximately 97% of the 'good' spectra, she can expect to accurately filter out about 71% of the 'bad' spectra. The fourth column will be discussed in the section on 'intermediate' spectra, below.

Computational Efficiency

The running time to extract the 18 optimal features from 10,000 spectra on a standard 2.6 GHz PC using MATLAB 7.5 is approximately 240 seconds. The time to actually classify the extracted features is negligible: 10,000 spectra are classified in 0.23 seconds. There is additional time overhead importing the data into MATLAB. However, the overhead as well as the feature extraction time would both be significantly reduced in a more efficient programming language such as C++. In particular, a RF is an ensemble of decision trees, and as such is easy to parallelize.

Expanded Positives

To test the flexibility of our classifier, we repeated the above experiment with an expanded notion of 'good' spectra. In the previous experiment 'good' spectra were defined to be those that have a SEQUEST [6] q-value of at most 0.01. In the following experiment, we have expanded this definition to include other commonly-used database search algorithms: to qualify as 'good' a spectrum must have a SEQUEST q-value of at most 0.01 OR an OMSSA [8] q-value of at most 0.01 OR a X! Tandem [4] q-value of at most 0.01. Having reassigned classes, we had to resplit the training/test data. As before, we trained the RF on 50% of the data, and tested on the remaining 50%. The resulting AUROC95 score was 0.762, which is well within the standard deviation of the CV scores, which indicates that our classifier is robust to feature and parameter choice. The very similar performance is indicative of the fact that OMSSA and X! Tandem do not provide much discriminative information above and beyond that of SEQUEST.

Filtering 'Intermediate' Spectra

Approximately half of all of the data in our original data source is neither 'good' or 'bad'. These 'intermediate' spectra have at least one peptide isotope distribution in the MS1 scan but have excessively high q-values and would generally not be considered to have a reliable peptide match. Not all of these spectra, however, are beyond identification and may be worth further investigation. For example, the peptide may have a post-translational modification that was not considered in the search, or there may be two peptide species isolated in the same window that could be identified with a specialized search. In general, any time there is a variation in the peptide sequence that is not represented in the database, the correct identification cannot be made. Because our definition of 'bad' spectra is based on the isotope distribution in the MS1 scan (and not, for instance, a Xcorr score), we suspect that our classifier is agnostic to such peptide variations, and would filter out only 'intermediate' spectra without a good peptide signal.

To understand how our classifier behaves with the 'intermediate' spectra, we turn once again to Table 9. The fourth column contains the fraction of the 'intermediate' set classified as 'bad' for a number of thresholds. This gives a complete picture of how all of the resulting data in an individual round of MS/MS would be dealt with by our classifier. For instance, if the user wants to retain 99% of the 'good' spectra, her or she can expect to filter out 57.8% of the 'bad' spectra and 26% of the intermediate spectra.

Non-IID Classification

The experimental results reported above are somewhat optimistic because the training and test partitions were drawn randomly from a superset of the five MS/MS runs, so there are spectra from each of the 5 runs in both the training and test sets. Thus the test set and training set are statistically similar. Since each MS/MS run has different statistics from any other, and, thus, a new test MS/MS run will likely not be IID with respect to the training data. To mimic this practical issue and better evaluate the performance of French Press we performed another set of experiments using all 25 runs detailed in Table 3, where, in turn, each of the 25 runs was held out entirely as a test set and the classifier trained on the remaining 24 runs.

These test results averaged over the 25 runs are in shown in Figures 5 and 6. The plot in Figure 5 shows the range of TP and TN that we observed across the 25 runs when using various thresholds for filtering. The threshold defines the maximum number of the trees in the random forest that classify the spectra as 'good' (divided by 100, the total number of trees) allowed such that the RF classifier calls the spectrum 'bad'. Thus, we see that at in the extremely conservative case of setting the threshold to 0 (that is, only remove spectra that have exactly 0 'good' votes), one can still remove 20% of the true negatives on average. More realistic thresholds are between 0 and 0.2, where at least 95% of true positives are retained with up to 80% of the true negatives removed. These results are promising and satisfying, in that they provide statistical evidence for the performance of the classifier at various thresholds.

The plot in Figure 6 is similar, but the threshold is chosen such that a certain percentage of the total spectra are retained after filtering. This is an alternative way of choosing a threshold, and provides a different viewpoint on performance. For example, if one wishes to retain 80% of the spectra, one can expect about 50% of the true negatives to be filtered out, and 95% of the true positives to be retained.

Figure 5: TP and TN Rates vs Thresholds

The plot depicts the mean and standard deviation true positive (TP) and true negative (TN) rates for a range of potential user-selected RF threshold. The centers of the bars are the means of the 25 runs, and the width of the bars indicates standard deviation over the 25 runs.

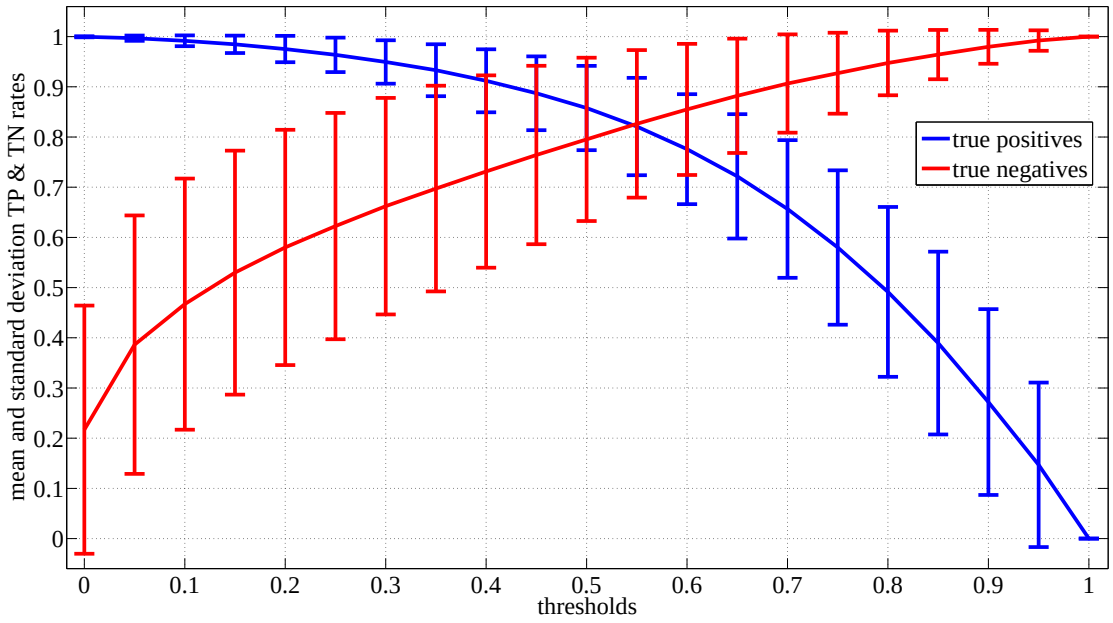


Figure 6: TP and TN Rates vs Fractions

The plot depicts the mean and standard deviation true positive (TP) and true negative (TN) rates for a range of user-selected fractions. The fractions on the x-axis are fractions of spectra retained after filtering; this is also user-selected parameter, much like the threshold in Figure 5. The centers of the bars are the means of the 25 runs, and the width of the bars indicates standard deviation over the 25 runs.

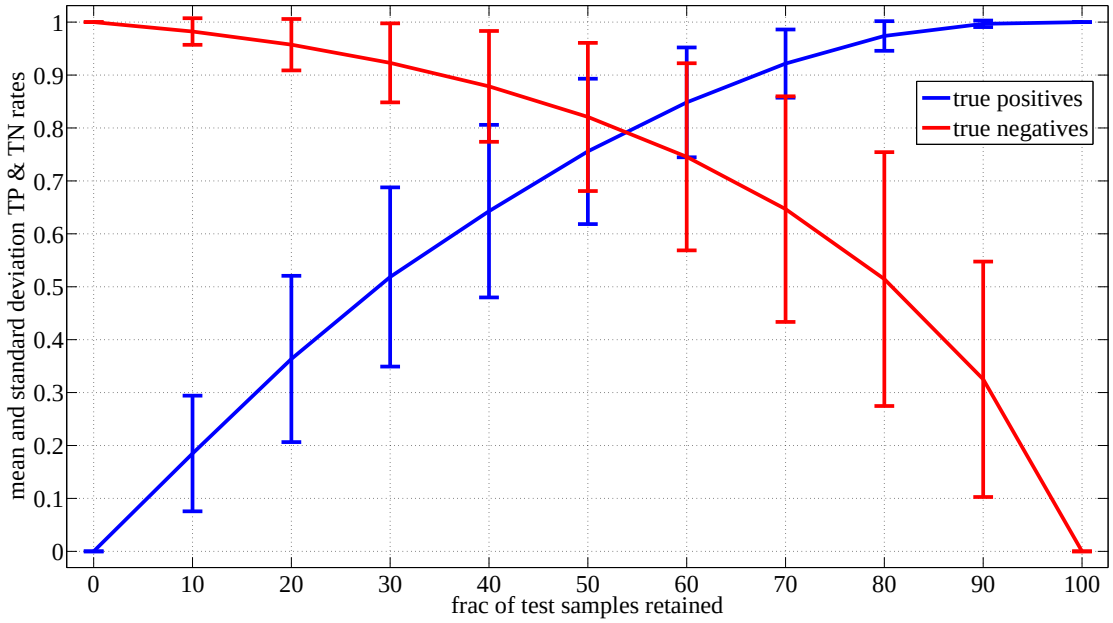


Table 10: Detailed Non-IID Results: Table shows the numbers of positives kept and negatives filtered out for three relevant thresholds. Each row shows the results for held-out run.

Run ID	Total positives	Positives kept at thresh.: 0.1	0.2	0.3	Total negatives	Negatives filtered at thresh.: 0.1	0.2	0.3
1	7208	7182	7139	7058	1100	1085	1096	1100
2	3181	3159	3095	2992	1018	603	708	777
3	3412	3406	3392	3349	5297	984	1554	2218
4	583	558	523	490	1656	493	646	760
5	4540	4537	4502	4439	6028	1253	2027	2777
6	3363	3325	3269	3158	905	544	640	707
7	2279	2265	2230	2176	1057	602	729	816
8	7716	7702	7665	7609	14967	696	1150	1807
9	2077	2074	2059	2040	7267	2202	3167	3944
10	776	775	771	755	2035	520	883	1222
11	2032	2030	2024	1997	7381	2346	3347	4159
12	8615	8609	8600	8572	1400	1398	1400	1400
13	1865	1862	1845	1808	7404	2362	3236	3944
14	1805	1795	1773	1717	1159	594	775	897
15	2204	2178	2130	2053	1063	574	751	835
16	2617	2606	2582	2518	1065	212	274	328
17	3153	3105	3022	2930	978	544	661	747
18	8370	8350	8310	8245	6581	2084	3045	3835
19	7490	7471	7439	7378	6499	2182	3061	3796
20	2494	2460	2405	2315	1062	508	666	753
21	4596	4522	4313	4010	1481	724	961	1096
22	7743	7493	7219	6835	3767	1822	2307	2572
23	3588	2305	1863	1614	4117	1015	1398	1577
24	3149	3145	3092	2991	7249	3337	4192	4900
25	7385	7363	7341	7299	1967	1967	1967	1967

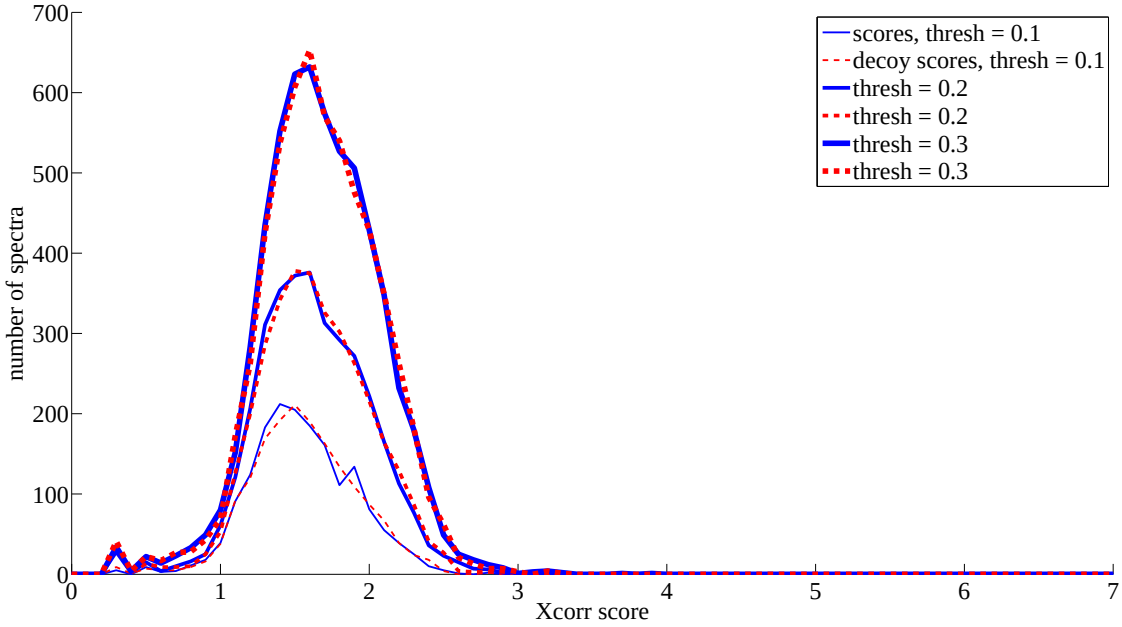
The reason we present the data in this form, and not as a collection of AUROC95 scores, has to do with how we anticipate the end-user might use the filter. It is difficult to choose the final threshold setting for the filtering of heretofore unseen data just by examining a set of 25 AUROC95 scores. Therefore, depending on the need, we offer two choices of setting a threshold: directly setting TP and TN rates based on information in the first plot, or choosing to retain a certain percentage of the total spectra based on information in the second plot.

Table 10 provides a closer look at the experiments in this section. We report the absolute numbers of positive spectra retained and negative spectra filtered out for thresholds of 0.1, 0.2, and 0.3. For almost every run (the notable exception being run #23) the overwhelming majority of positives are retained, while large percentages of negatives (as high as 100%) are filtered out.

Examining *Xcorr* Values

To gain more intuition about the quality of filtered-out spectra, we turn to the *Xcorr* score [6]. For filtering thresholds of 0.1, 0.2, and 0.3, we obtained the histograms of *Xcorr* values for the spectra that were filtered out, for both decoy and target databases. Two examples randomly chosen from the 25 runs are shown in Figure 7. The two panels in the figure show that the bulk of the filtered-out spectra have *Xcorr* values between 0.5 and 2.5, which are considered quite low. This matches the above results in that 'good' spectra are rarely filtered out. With respect to these figures, there is little variation amongst the 25 runs, and the two figures shown are representative.

Figure 7: Histograms of *Xcorr* scores of filtered-out spectra. Top and bottom figures are two randomly chosen runs (out of 25).



Conclusions

Mass spectrometry has become the primary technology for the characterization of proteins within complex mixtures. With each advance in instrumentation we obtain the capability to collect more and more MS/MS spectra per unit time. With the release of the new LTQ-Velos ion trap mass spectrometer we can collect MS/MS spectra at a rate that is more than two times faster than the previous generation LTQ. However, we do not obtain twice as many peptide spectrum matches; thus, this improvement in instrument speed also increases the fraction of MS/MS spectra that remain unidentifiable [27]. For French Press to successfully work with other ion trap instruments, it may be necessary to re-implement the feature selection and training with data from that instrument.

There are many reasons for MS/MS spectra remaining unidentified after a database search. Some spectra are of poor quality or belong to non-peptide molecular species. It is likely that sampling peptides of lower abundance will further exacerbate this problem. Thus, it is imperative to be able to differentiate spectra that are identifiable versus those that are not prior to performing the computationally expensive database search.

The key contribution of this work is in our definition of low quality spectra, motivated by the observation that a spectrum can have a low score from database searches for reasons unrelated to the quality of its spectrum. In contrast to prior work, our definition of 'bad' spectra is *unrelated* to the search score: we define 'bad' spectra as those that are determined to have no peptide isotope distribution in the MS1 scan. Indeed, this is the main advantage of French Press over other spectra quality filters: we have encoded the computationally expensive information that Hardklor is able to obtain from the MS1 scan into an efficient, quick-to-evaluate classifier. That is also why we did not compare to other classifiers in this work. Namely, none of the prior work detailed above uses the same definition of 'good' and 'bad' spectra as we do.

We have included all the relevant details of our classifier and experimental design, and have demonstrated its efficacy. Research-grade code is available that runs the F_{press} classifier, currently trained on all of the 25 runs discussed in this work. We can maintain $> 99\%$ of the peptide identifications while eliminating

roughly 50% of the unidentifiable MS/MS spectra. While many of these spectra may be identifiable if we perform extensive modification searches, these searches are computationally expensive and can be performed on only the MS/MS spectra that are classified as "good" and that are not identified as being derived from unmodified peptides [3].

The analysis of MS/MS spectra represents a challenging problem in obtaining statistically significant identifications because of the enormous number of multiple hypothesis tests that are made in the analysis of the mass spectrometry data [11, 12], as well as the complications inherent in very large proteomics datasets [25]. More peptides can be identified at a constant false discovery rate by either improving the discrimination between the correct and incorrect results or just eliminating incorrect identifications. By eliminating MS/MS spectra that are of insufficient quality to provide a correct peptide spectrum match, in selected cases the database search score threshold can be set lower to obtain more correct peptide identifications while maintaining the same false discovery rate.

Acknowledgements

This work was funded by two United States PECASE Awards.

Acknowledgements

This research was supported by two United States PECASE awards.

References

[1] M Bern, D Goldberg, W H Mcdonald, and J R Yates. Automatic quality assessment of peptide tandem mass spectra. *Bioinformatics*, 20(1):49–54, 2004.

[2] L Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[3] R Craig and R C Beavis. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid. Commun. Mass Spectrom.*, 17(20):2310–2316, 2003.

[4] R Craig and R C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, June 2004.

[5] J Ding, J Shi, and F-X Wu. Quality assessment of tandem mass spectra by using a weighted k-means. *Clinical Proteomics*, 5(1):15–22, 2009.

[6] J K Eng, McCormack A L, and J R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.*, 5(11):976–989, 1994.

[7] K Flikka, L Martens, J Vandekerckhove, K Gevaert, and I Eidhammer. Improving the reliability and throughput of mass spectrometry-based proteomics by spectrum quality filtering. *Proteomics*, 6(7):2086–94, 2006.

[8] L Y Geer, S P Markey, J A Kowalak, L Wagner, M Xu, D M Maynard, X Yang, W Shi, and S H Bryant. Open mass spectrometry search algorithm. *J. Proteome Res.*, 3(5):958–964, October 2004.

[9] T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[10] M R Hoopmann, G L Finney, and M J MacCoss. High-speed data reduction, feature detection, and ms/ms spectrum quality assessment of shotgun proteomics data sets using high-resolution mass spectrometry. *Anal. Chem.*, 79(15):5620–32, 2007.

[11] L Käll, J D Canterbury, J Weston, W S Noble, and M J Maccoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods*, 4(11):923–5, October 2007.

[12] L Käll, J D Storey, M J MacCoss, and W S Noble. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J. Proteome Res.*, 7(1):29–34, 2008.

[13] L Käll, J D Storey, M J MacCoss, and W S Noble. Posterior error probabilities and false discovery rates: two sides of the same coin. *J. Proteome Res.*, 7(1):40–44, 2008.

[14] A A Klammer, C C Wu, M J MacCoss, and W S Noble. Peptide charge state determination for low-resolution tandem mass spectra. In *CSB'05*, pages 175–185, Washington, DC, USA, 2005. IEEE Computer Society.

[15] T Koenig, B H Menze, M Kirchner, F Monigatti, K C Parker, Thomas Patterson, Judith Jebanathirajah Steen, Fred A. Hamprecht, and Hanno Steen. Robust prediction of the mascot score for an improved quality assessment in mass spectrometric proteomics. *J. Proteome Res.*, 7(9):3708–17, 2008.

[16] A J Link, J Eng, D M Schieltz, Edwin Carmack, G J Mize, D R Morris, B M Garvik, and J R Yates. Direct analysis of protein complexes using mass spectrometry. *Nat. Biotechnol.*, 17:676–682, 1999.

[17] G C McAlister, D Phanstiel, C D Wenger, M V Lee, and J J Coon. Analysis of tandem mass spectra by ftms for improved large-scale proteomics with superior protein quantification. *Anal. Chem.*, 2009.

[18] A L McCormack, D M Schieltz, B Goode, S Yang, G Barnes, D Drubin, and J R Yates. Article direct analysis and identification of proteins in mixtures by lc/ms/ms and database searching at the low-femtomole level. *Anal. Chem.*, 69:767–779, 1997.

[19] G E Merrihew, C Davis, B Ewing, G Williams, L Käll, B Frewen, W S Noble, P Green, J H Thomas, and M J MacCoss. Use of shotgun proteomics for the identification, confirmation, and correction of c. elegans gene annotations. *Genome Res.*, 18(10):1660–1669, 2008.

[20] R E Moore, M K Young, and T D Lee. Method for screening peptide fragment ion mass spectra prior to database searching. *J. Am. Soc. Mass Spectrom.*, 11(5):422–6, 2000.

[21] S Na and E Paek. Quality assessment of tandem mass spectra based on cumulative intensity normalization. *J. Proteome Res.*, 5(12):3241–8, 2006.

[22] A I Nesvizhskii, F F Roos, J Grossmann, M Vogelzang, J S Eddes, Wi Gruissem, S Baginsky, and R Aebersold. Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data: toward more efficient identification of post-translational modifications, sequence polymorphisms, and novel peptides. *Mol. Cell. Proteomics*, 5(4):652–670, 2006.

[23] J V Olsen, J C Schwartz, J Griep-Raming, M L Nielsen, E Damoc, E Denisov, O Lange, P Remes, D Taylor, M Splendore, E R Wouters, M Senko, A Makarov, M Mann, and S Horning. A dual pressure linear ion trap orbitrap instrument with very high sequencing speed. *Mol. Cell Proteomics.*, 8(12):2759–2769, 2009.

[24] S Purvine, N Kolker, and E Kolker. Spectral quality sssessment for high-throughput tandem mass spectrometry proteomics. *OMICS*, 8(3):255–265, 2004.

[25] L Reiter, M Claassen, S P Schrimpf, M Jovanovic, A Schmidt, J M Buhmann, M O Hengartner, and R Aebersold. Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry. *Mol. Cell Proteomics*, 8(11):2405–2417, 2009.

[26] J Salmi, R Moulder, J-J Filen, O S Nevalainen, T A Nyman, R Lahesmaa, and T Aittokallio. Quality classification of tandem mass spectrometry data. *Bioinformatics*, 22(4):400–406, 2006.

[27] T P Second, J D Blethrow, J C Schwartz, G E Merrihew, M J MacCoss, D L Swaney, J D Russell, J J Coon, and V Zabrouskov. Dual-pressure linear ion yrap mass spectrometer improving the analysis of complex protein mixtures. *Anal. Chem.*, 81(18):7757–7765, 2009.

[28] D C Stahl, K M Swiderek, M T Davis, and T D Lee. Data-controlled automation of liquid chromatography/tandem mass spectrometry analysis of peptide mixtures. *J. Am. Soc. Mass Spectrom.*, 7(6):532–540, 1996.

[29] H Steen and M Mann. The abc's (and xyz's) of peptide sequencing. *Nat. Rev. Mol. Cell Biol.*, 5(9):699–711, September 2004.

[30] M P Washburn, D Wolters, and J R Yates. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat. Biotechnol.*, 19(3):242–247, 2001.

[31] J W H Wong, M J Sullivan, H M Cartwright, and G Cagney. msmseval: Tandem mass spectral quality assignment for high-throughput proteomics. *Bioinformatics*, 8:51+, 2007.

[32] F-X Wu, P Gagne, A Droit, and G G Poirier. Quality assessment of peptide tandem mass spectra. In *IMSCCS'06*, pages 243–50, Washington, DC, USA, 2006. IEEE Computer Society.

[33] M Xu, L Y Geer, S H Bryant, J S Roth, J A Kowalak, D M Maynard, and S P Markey. Assessing data quality of peptide mass spectra obtained by quadrupole ion trap mass spectrometry. *J. Proteome Res.*, 4(2):300–5, 2005.

[34] J R Yates. Mass spectrometry and the age of the proteome. *J. Mass Spectrom.*, 33(1):1–19, 1998.

[35] A-M Zou, F-X Wu, J-R Ding, and G G Poirier. Quality assessment of tandem mass spectra using support vector machine (svm). *Bioinformatics*, 10(1):S49, 2009.