Comparison of Discrimination Methods for Peptide Classification in Tandem Mass Spectrometry

Anthony Bonner and Han Liu

***Abstract*— Proteomics—the direct analysis of the expressed protein components of a cell—is critical to our understanding of cellular biological processes. Key insights into the action and effects of a disease can be obtained by comparison of the expression of the expressed proteins in normal versus diseased tissue. Tandem mass spectrometry(MS/MS) of peptides is a central technology for Proteomics, enabling the identification of thousands of peptides from a complex mixture. With the increasing acquisition rate of tandem mass spectrometers, there is an increasing potential to solve important biological problems by applying data-mining and machine-learning techniques to MS/MS data. These problems include (i) estimating the levels of the thousands of proteins in a tissue sample, (ii) predicting the intensity of the peaks in a mass spectrum, and (iii) explaining why different peptides from the same protein have different peak intensities. In other works, we have focussed on the first two problems. In this paper, we focus on the last problem. In particular, we try to explain why some peptides produce peaks of great intensity, while others produce peaks of low intensity, and we treat this as a classification problem. That is, we experimentally evaluate and compare a variety of discrimination methods for classifying peptides into those that produce high-intensity peaks and those that produce low-intensity peaks. The methods considered include K-Nearest Neighbours (KNN), Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes, and Hidden Markov Models (HMMs). Experiments using these methods were conducted on three real-world datasets derived from tissue samples of Mouse. The methods were then evaluated using ROC curves and cross validation.**

***Index Terms*— Data mining, Machine learning, Proteomics, Bioinformatics, Tandem mass spectrometry, Peptide classification.**

I. Introduction

TANDEM Mass Spectrometry (MS/MS) of peptides is a central technology of Proteomics, enabling the identification of thousands of peptides and proteins from a complex mixture [16], [12], [1]. In a typical experiment, thousands of proteins from a tissue sample are fragmented into tens of thousands of peptides, which are then fractionated, ionized and passed through a tandem mass spectrometer. The result is a collection of spectra, one for each protein, where each peak in a spectrum represents a single peptide [11], [18]. With the increasing acquisition rate of tandem mass spectrometers, there is an increasing potential to solve important biological problems by applying data-mining and machine-learning techniques to MS/MS data [8], [6]. In this paper, we focus on one such problem: explaining why different peptides have different peak intensities. One important factor is clearly the amount of protein input to the mass spectrometer (since more input implies more output). However, other factors are important as well, such as the efficiency with which various peptides are produced, fractionated and ionized. Our goal is to determine how all these other factors are influenced by a peptide's amino-acid sequence. Once these influences are understood, it may be possible to predict the exact MS/MS spectrum of a protein, including the intensities of all its peaks. More importantly, it may also be possible to solve the inverse problem: given the MS/MS spectrum of a protein, estimate the amount of protein that was input to the mass spectrometer. This is a fundamental problem whose solution would enable biologists to determine the levels of the thousands of proteins in a tissue sample [2], [3].

As a first step in explaining why different peptides produce peaks of different intensity, we reduce the problem to its simplest terms: explaining why some peptides produce peaks of great intensity, while others produce peaks of low intensity. Moreover, we treat this as a classification problem. That is, given the amino-acid sequence of a peptide, we attempt to classify the peptide as either high-intensity or low-intensity. If we can do this reliably, then we have effectively discovered what it is about the amino-acid sequence of a peptide that determines whether it produces high- or low-intensity peaks. This is especially true if the parameters of the classifier can be interpreted biologically. To tackle this classification problem, this paper evaluates and compares a variety of classification methods on MS/MS data. Conducting the evaluation involves three main steps: producing a set of labeled data, training a number of different classifiers on a portion of the data (the "training data"), and evaluating the effectiveness of the classifiers on the remaining data (the "testing data").

To produce a labeled set of data, we first obtained a set of several thousand MS/MS spectra.[1] From these spectra, we produced two classes of peptides, those with high-intensity peaks, and those with low-intensity peaks. One complication is that in the high-throughput MS/MS experiments from which the data was derived, the amount of protein input to the mass spectrometer is unknown. Thus, it is in general unclear whether a high-intensity peak is due to the properties of the peptide (which we are interested in) or simply due to a large amount of protein at the input (which we are not interested in). We resolved this problem by focusing on the spectra one protein at a time. In the spectrum of a single protein, all the

Both authors are with the Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4. {bonner,hanliu}@cs.toronto.edu

[1] Courtesy of the Emili Laboratory at the Banting and Best Department of Medical Research at the University of Toronto

peaks are derived from the same (unknown) amount of protein, so differences in peak intensity are not due to differences in protein input. In this way, by picking the highest and lowest peaks in each spectrum, we produced two classes of peptides, one class labeled “high intensity” and the other labeled “low intensity”.

Since this is an initial study, we chose a number of basic classification methods which are extensively used in the data-mining literature: K-Nearest Neighbours (KNN), Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes, and Hidden Markov Models (HMMs). Most of these methods have also been used in the context of microarray data to distinguish various cancer types [5]. To evaluate the methods for our application, we first trained them on two thirds of the labeled data, and then evaluated them on the remaining one third. To compare the methods, we produced both ROC curves and a table of classification error rates. In general, classification error rates depend on the size of the labeled classes, and can be deceptively low when the classes have very different sizes (as ours do, since most peptide peaks are low-intensity, not high-intensity). To compensate for this, we used the ROC data to estimate classification error rates for classes of equal size. In addition, the ROC curves themselves are insensitive to differences in class size. In this way, we can compare the performance of the classifiers on various data sets without having to worry about variations in class size. The ROC curves also allow us conveniently to compare classifiers for many different values of a discriminant threshold parameter. To assess variability in performance, we generated a number of ROC curves for each method using 10-fold cross validation.

Finally, it is worth noting that of the six classification methods we chose to evaluate, three are designed for classifying vectors (LDA, QDA and Logistic Regression), two are most-easily applied to vectors (KNN and Naive Bayes), while only the last (HMM) is specifically designed for classifying sequences. HMMs can therefore be applied directly to the amino-acid sequences of peptides. To apply the other five methods, we first converted the peptide sequences to feature vectors. Since there is a certain loss of information in this conversion, one might expect HMMs to perform the best, especially since they have achieved considerable success in other areas of sequence classification, such as speech recognition [17] and DNA sequence analysis [15]. Surprisingly, we found that for our peptide-classification problem, HMMs performed the worst. We discuss reasons for this in Section V.

The paper is organized as follows: Section II introduces Peptide Tandem Mass Spectrometry; Section III reviews the six classification methods we evaluate; Section IV describes our data and our experimental design; Section V presents and discusses our experimental results; and Section VI presents conclusions.

II. Peptide Tandem Mass Spectrometry

In Tandem Mass Spectrometry, a mixture of tens of thousands of unknown proteins are taken from a tissue sample. By adding a specific amount of the enzyme trypsin, the proteins are digested and fragmented into peptides. The resulting mixture is then fractionated, ionized and sent at high speed through an electric field, where the paths of the different peptides are bent by different amounts before hitting a plate. This produces a so-called “mass spectrum” consisting of various peaks. Each peak occurs at a particular location on the plate, as determined by the peptide’s mass-to-charge ratio, and the intensity of the peak represents the number of peptide molecules to hit the plate at that location [16], [12], [1].

The intensity of a peak is influenced by the amount of protein in the input mixture. However, the exact mechanism determining the intensity of a spectral peak is poorly understood [8]. In an ideal experiment, there would be no loss during digestion, fractionation and ionization. So, in the MS/MS spectrum for a given protein, one would expect that each peak would contain one peptide molecule for each protein molecule in the input. Consequently, an ideal spectrum for a given protein would consist of peaks of equal intensity. However, this is not observed experimentally. Figure 1 illustrates the differences between an ideal MS/MS spectrum and an experimental one.

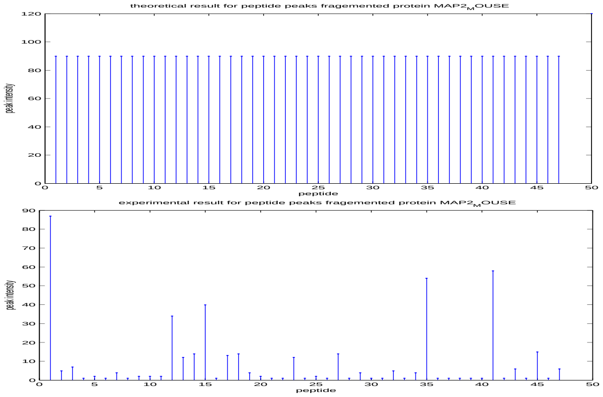

Fig. 1. The upper panel is the theoretical MS/MS spectrum for the 47 peptides fragmented from protein MAP2-MOUSE digested with trypsin. The lower panel is an experimentally derived spectrum of the same protein. There is good agreement between the locations of peaks in the two spectra, but virtually no agreement between peak intensities.

Numerous reports in the literature address the question of what factors affect the quality of a MS/MS experiment [11], [18]. An obvious factor influencing peak intensity is the concentration of the peptides in the sample. However, it is not the sole factor. Other factors include sample preparation methods, the pH and composition of the solution containing the protein mixture, the characteristics of the MS/MS apparatus, and the characteristics of the tissue sample being analyzed [8]. These factors and others all affect the intensities of peaks in a MS/MS spectrum. Unfortunately, no model exists for accurately predicting these peak intensities. Explaining why some peptides produce high-intensity peaks and some produce low-intensity peaks is a first step towards developing such a model, and is the goal of this paper.

To this end, the Emili Laboratory at the Banting and Best Department of Medical Research at the University of Toronto has provided us with several thousand MS/MS spectra. Table I shows a tiny sample of this data. (Details on how this data

TABLE I

A FRAGMENT OF THE ORIGINAL DATA FILE

Protein ID	Peptide	Count	Charge
Q91VA7	$TRHNNLVIIR$	4	2
Q91VA7	$KLDLFAVHVK$	3	2
⋮	⋮	⋮	⋮

was generated can be found in [21].) The first column in the table is the Swissprot accession number identifying a protein. The second column is the amino-acid sequence of a peptide fragmented from the protein. The third column correlates well with the intensity of the spectral peak produced by the peptide. (Its values are integers because it represents a count of peptide molecules.) The last column represents the charge of the peptide ion, which is typically 1 or 2. Notice that there may be many entries for the same protein, since a single protein can produce many peptides.

In fragmenting the proteins to produce peptides, the proteins were digested by the enzyme trypsin, which cuts c-terminal to lysine (K) or arginine (R). Trypsin may occasionally cut at other locations, and other enzymes may nick the protein; hence, we sometimes see partial tryptic peptides (a K or R at the c-terminus, or flanking the peptide at the N-terminus).

III. DISCRIMINATION METHODS

This section outlines the classification methods used in our study. Each of the methods can be used to assign objects to one of several classes, though we use them in a strictly binary mode, to assign peptides to one of two classes, which we refer to simply as Class 1 and Class 2, respectively. In our application, Class 1 is the set of peptides that produce high-intensity peaks, and Class 2 is the set of peptides that produce low-intensity peaks. How the peptides are represented depends on the classification method used. For one of the methods (Hidden Markov Models), each peptide is represented by its amino-acid sequence. For the remaining methods, each peptide is represented by a vector, **x**, of features extracted from the sequence. How these features are extracted is described in Section IV.

All the classification methods used in this study produce classifiers that are discriminative [13], [14], [20]. That is, given an object, the classifier produces a score or likelihood, p_1, that the object belongs to Class 1, as well as a score, p_2, that it belongs to Class 2. An object is assigned to Class 1 if and only if $p_1/p_2 > t$, where t is a decision threshold.

a) Naive Bayes: Most of the classification methods used in this paper are based on Bayes Rule. They differ primarily in their assumptions about the prior probabilities, $f_k(\mathbf{x})$. In the Naive Bayes classifier, it is assumed (simplistically) that the features of the vector **x** can be treated as independent random variables. Thus, if $\mathbf{x} = (x_1, ..., x_n)$, then

$$f_k(\mathbf{x}) = \prod_i f_{ki}(x_i)$$

where $f_{ki}(x_i)$ is the (marginal) probability of x_i for class k. In our application, the components of the feature vector, **x**, are discrete (as we shall see), and the most common values are 0 and 1, so we use binary distributions to estimate f_{ki}. For each class, Bayes Rule is then used to compute the posterior probability that **x** belongs to the class.

b) Linear Discriminant Analysis: In Linear Discriminant Analysis (LDA), each class is modeled by a multivariate Gaussian distribution, where each class is assumed to have the same covariance matrix. In fitting the Gaussian distributions to the data, LDA produces maximum likelihood estimates for several parameters: π_k, the prior probability of class k; μ_k, the mean of class k; and Σ, the common covariance matrix. Because the class distributions are Gaussian and the covariance matrices are equal, the likelihood ratio test reduces to a particularly simple form: $\mathbf{x} \bullet \mathbf{w} > t$, where t is a decision threshold, **w** is a parameter vector computed from Σ, μ_1 and μ_2, and $\bullet$ denotes the inner product of two vectors [9]. The decision boundary between the two classes is therefore linear, and more specifically, a hyperplane normal to **w**.

c) Quadratic Discriminant Analysis: Quadratic Discriminant Analysis (QDA) is a generalization of LDA that does not assume the classes have the same covariance matrix. In this case, the decision boundary between the two classes is not linear but quadratic, and can in general be a hypersphere, a hyperellipsoid, a hyperparaboloid, a hyperhyperboloid, or any combination thereof [4].

d) Logistic Regression: Like QDA, Logistic Regression can be viewed as a generalization of LDA. However, whereas QDA has more parameters than LDA, Logistic Regression has fewer. Like LDA, Logistic Regression provides a linear decision boundary between classes. The main difference is that Logistic Regression is in a sense more direct. Instead of first fitting multivariate Gaussians to each class (which requires estimating $O(n^2)$ parameters), Logistic Regression fits a linear decision boundary directly to the data (which requires estimating only n parameters, where n is the dimension of the feature space).

e) K-Nearest Neighbors: K-Nearest Neighbors (KNN) is different from the other methods considered in this paper in that it is *non-parametric*, so there are no parameters to estimate. To classify an object, **x**, the method finds the K objects (or neighbors) in the training data that are closest to it. The object is then assigned to Class 1 if and only if $K_1/K_2 > t$, for some decision threshold, t. Here, K_1 is the number of K-nearest neighbors in Class 1, and K_2 is the number in Class 2 (so $K_1 + K_2 = K$). To use KNN, one must choose a measure of "distance" (or similarity) between two objects. This paper uses Euclidean distance.

f) Hidden Markov Models: A (first-order) Hidden Markov Model (HMM) is a finite-state automaton in which each state transition has a probability. In addition, each state has a set of outputs symbols, each with an associated probability. When an HMM is in a given state, it chooses an output symbol probabilistically, and prints it. Efficient algorithms have been developed for HMMs. For instance, given a sequence, **s**, and a HMM, inference algorithms compute the probability, $\pi(\mathbf{s})$, that **s** will be generated by the HMM. Learning algorithms also exist for HMMs. Given a set of sequences, these algorithms will find the HMM with a given number of states that is most likely to generate the set.

IV. Datasets and Study Design

A. Real-world Datasets

The experimental results in this paper are based on a number of tables of real-world data similar to Table I. They consist of three datasets derived from tissue samples taken from Mouse and were provided by the Emili Lab in the Banting and Best Department of Medical Research at the University of Toronto. We refer to these data sets as **Mouse Brain Data**, **Mouse Heart Data**, and **Mouse Kidney Data**. The Brain data set contains 10,786 peptides, with peak intensities ranging from 1 to 2,500; the Heart data set contains 9,623 peptides, with peak intensities from 1 to 1,996; and the Kidney data set contains 8,791 peptides, with peak intensities from 1 to 1,491.

B. Study Design

As described in Section I, we divide the peptides in the training data into two classes, those that produce high-intensity peaks, and those that produce low-intensity peaks. One complication is that in the high-throughput MS/MS experiments from which the data was derived, the amount of protein input to the mass spectrometer is unknown. Thus, it is in general unclear whether a high-intensity peak is due to the properties of the peptide (which we are interested in) or simply due to a large amount of protein at the input (which we are not interested in). We resolved this problem by focusing on the spectra one protein at a time. In the spectrum of a single protein, all the peaks are derived from the same (unknown) amount of protein, so differences in peak intensity are not due to differences in protein input. In this way, by picking the highest and lowest peaks in each spectrum, we can factor out the effect of protein input levels.

To do this, we first identified those proteins that have at least two peptides with observable peaks in their spectra. We then built an index for these proteins, and a separate index for their peptides. For the Brain dataset, the indexes contain 8,527 peptides and 1,664 proteins, respectively. For the Heart dataset, the indexes contain 7,660 peptides and 1,281 proteins, respectively. For the Kidney dataset, the indexes contain 7,074 peptides and 1,291 proteins, respectively.

For each protein, we then picked the peptides with the highest and lowest spectral peaks. Moreover, we did this only for proteins with a wide range of peak intensities, so that we could be sure of obtaining peptides with genuinely low- and high-intensity peaks. We did this in two different ways, producing two pairs of peptide classes, which we call MAX-MIN and STD-DEV, respectively. The two pairs differ only in the proteins selected. For MAX-MIN, a protein is selected if (i) the most intense peak in its spectrum has a count greater than 12, and (ii) the count of this highest peak is at least 12 times greater the count of the lowest peak. For the STD-DEV, a protein is selected if the standard deviation of all its peak intensities is greater than 4. Both MAX-MIN and STD-DEV contain two classes of peptides, one with high-intensity peaks, and one with low-intensity peaks. In MAX-MIN, the two classes of peptides are guaranteed to have very different peak intensities, since the high-intensity peaks are guaranteed to be at least 15 times greater than the low-intensity peaks. In STD-DEV, the peptides in the two classes do not always have such a great difference in peak intensity, but they contain more peptides, $i.e.$, more data on which to train and test the classifiers. In this way, from the original, unlabeled Kidney dataset, we generated two labeled datasets, one using MAX-MIN and one using STD-DEV. Likewise for the Heart and Brain datasets, for a total of six labeled datasets.

The sizes of the datasets generated by the MAX-MIN method are as follows: the Brain dataset has 658 peptides in the “high intensity” class, and 1567 in the “low intensity ” class; the Heart dataset has 389 peptides in the “high intensity” class, and 1232 in the “low intensity ” class; the Kidney dataset has 470 peptides in the “high intensity” class, and 1,211 in the “low intensity ” class. The sizes of the datasets generated by the STD-DEV method are as follows: the Brain dataset has 818 peptides in the “high intensity” class, and 1,735 in the “low intensity ” class; the Heart dataset 463 peptides in the “high intensity” class, and 1,297 in the “low intensity ” class; the Kidney dataset has 550 peptides in the “high intensity” class, and 1,327 in the “low intensity ” class;

We estimated the generalization error of the methods in two ways. In the first approach, we divided each of the six labeled datasets randomly into training and testing data, in a 2:1 ratio. We then trained and tested each classification method on each of the six datasets. The results show the variation in performance of the methods over different datasets and different data-generation methods. In the second approach, we used only one of the six datasets: the Kidney dataset generated by STD-DEV. On this dataset, we did 10-fold cross validation for each classification method. The results show the variation in performance of each method over different splits in the dataset.

Finally, for many of the classification methods used in this study, we must represent each peptide as a vector, **x**. We have two ways of doing this, using vectors with 21 and 421 components, respectively. The 21-component vectors represent the amino-acid composition of a peptide. Since there are twenty different amino acids, the vector has 20 components, $(x_1, ..., x_{20})$, where the value of x_i is the number of occurrences a particular amino acid in the peptide. In addition, the vector has a 21^{st} , x_0, whose value is always 1, to represent a bias term, as is common in machine learning models [9]. The 421-component vector includes the original 21 components plus 400 more representing the dimer composition of a peptide. A *dimer* is a sequence of two amino acids, and since there are 20 distinct amino acids, there are 400 distinct dimers. This, larger vector representation was used only with the Naive bayes classifier, because of its reputation for working well in high dimesnions. The other vector-based classifiers were used only with 21-component vectors, to prevent overfitting. For comparison purposes, Naive Bayes was sometinmes also used with this shorter vector representation.

V. Results and Analysis

Figures 2 through 5 show ROC curves evaluating the performance of each of the classification methods under various

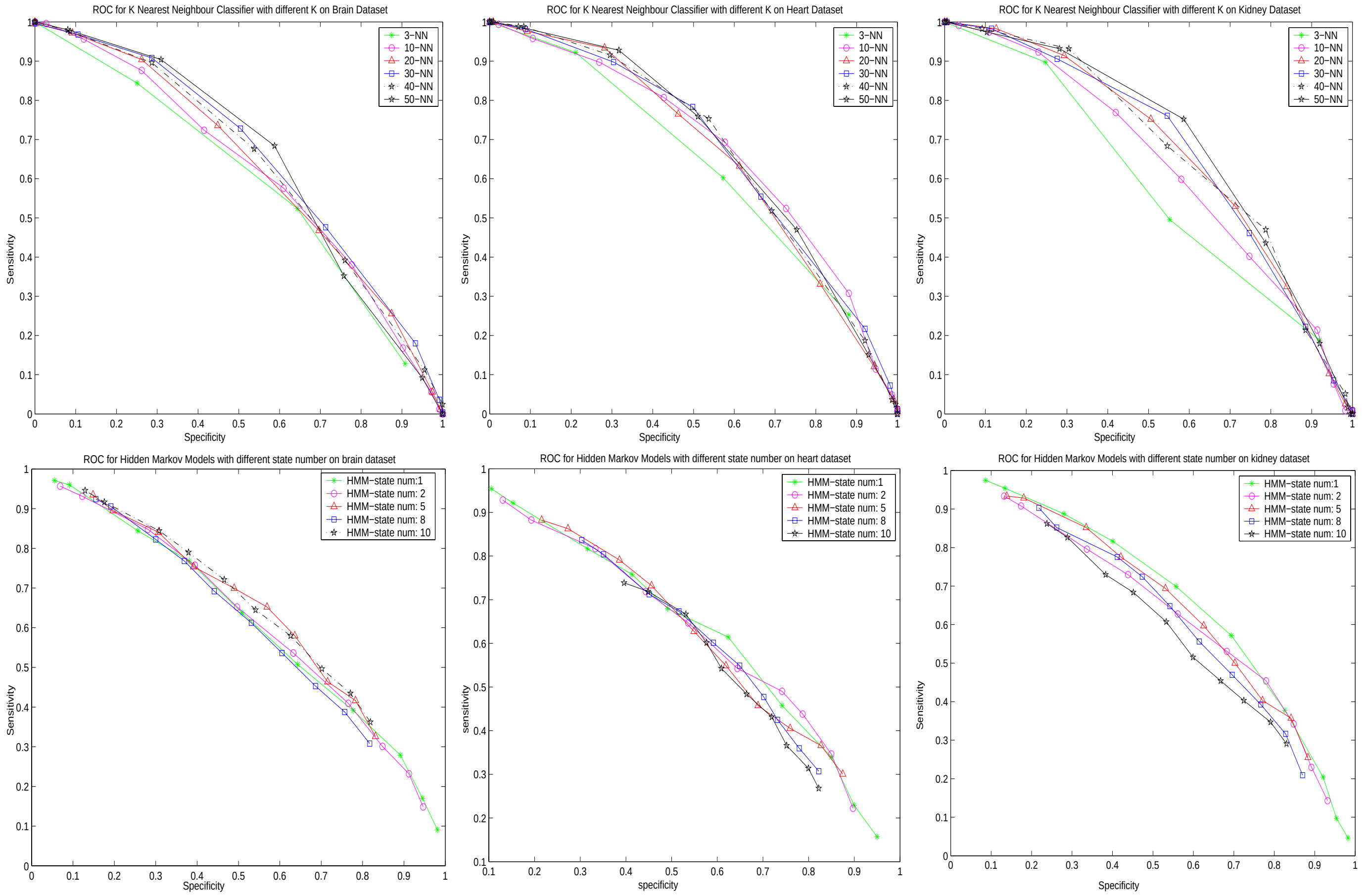

Fig. 2. ROC curves illustrating the performance on three different datasets of the KNN and HMM classifiers. The top panels show KNN with six different values of K. The bottom panels show HMM with five different numbers of states. The left figures are for the Mouse brain dataset, the middle figures are for the Mouse heart dataset, and the right figures are for the Mouse kidney dataset. All three datasets were generated by the STD-DEV method.

conditions. Following the convention given in [9], the horizontal axis of our ROC curves is specificity, and the vertical axis is sensitivity. In our case, specificity is the probability that a peptide with high-intensity peaks is predicted to have high-intensity peaks. Likewise, sensitivity is the probability that a peptide with low-intensity peaks is predicted to have low-intensity peaks. Each ROC curve corresponds to a single classification method, and each point in an ROC curve corresponds to the method being used with a different decision threshold.

Figure 2 shows ROC curves for K-Nearest Neighbors and Hidden Markov Models, with different values for K and number of states, respectively. The six subfigures represent experimental results for each of the two methods on each of the three datasets generated by the STD-DEV method. Note that in these figures, K-NN performs best when K=50, except on the heart dataset, where 10-NN performs better. Likewise, HMM performs best when it has only 1 state, except on the brain dataset, where it performs best with 5 states. Figure 3 shows ROC curves for all six classification methods on all six labeled datasets. In the top half of the figure, the datasets were generated by the MAX-MIN method. In the bottom half, they were generated by the STD-DEV method. Except for Naive Bayes, which used the 421-component vectors, all the vector-based classifiers in this figure used 21-component vectors.

Figure 4 shows the results of cross validation on each of the six methods. Here, KNN is used with K=50, and HMM is used with a single state, the values for which they performed best in the earlier experiments. Each of the six subfigures contains ten ROC curves, one for each run of 10-fold cross validation. The one exception is the subfigure for the Naive Bayes classifier, which contains twenty ROC curves, ten for Naive Bayes used with 21-component vectors, and ten for 421-component vectors. The other vector-based classifiers were all used with 21-component vectors. The curves in Figure 4 illustrate the variability in our estimates of generalization error, variability due to the splitting of the data into training and test sets. Figure 5 shows average ROC curves for each classification method. Each point in an average curve is the mean of ten corresponding points in ten curves in Figure 4. These average curves provide an estimate of the average generalization error of each classification method.

A. ROC Curves and Evaluation

From the ROC curves, several conclusions are immediately apparent: (i) LDA performs the best, (ii) Hidden Markov Models perform the worst, (iii) the variability in the other methods is large compared to the differences in their average

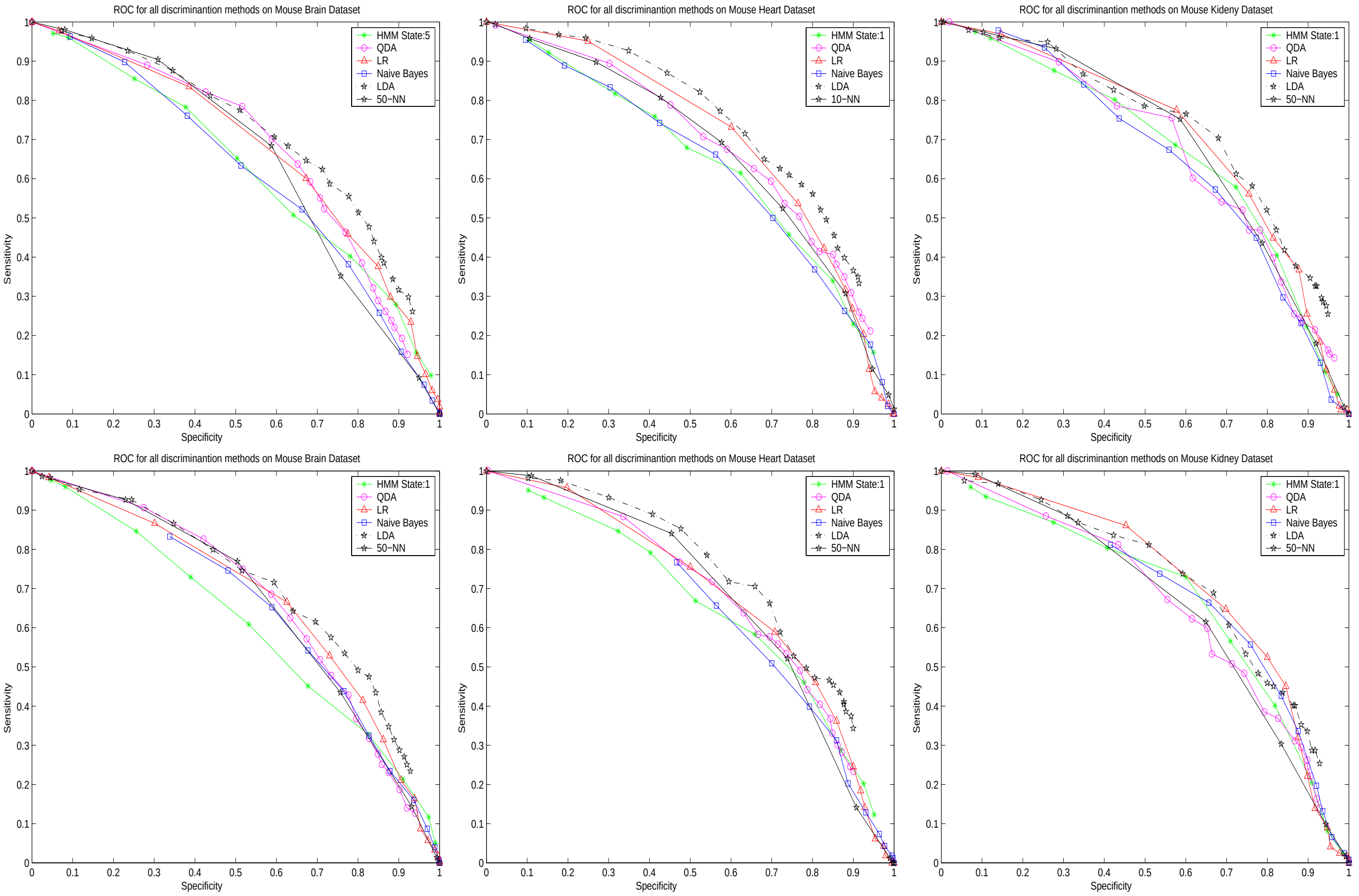

Fig. 3. Comparison of all the discrimination methods on three datasets. The top panels are for datasets generated by the MAX-MIN method, and the bottom panels are for datasets generated by the STD-DEV method.

performance, and (iv) except for HMM, all the methods performed better with a 9:1 split of training to testing data than with a 2:1 split. This last observation is probably due to the larger portion of training data available in the 9:1 split.

The simple LDA method produced impressively better performance than other, more-sophisticated methods, such as QDA and especially Hidden Markov Models. With the exception of the Mouse Kidney dataset generated by STD-DEV, on which Logistic Regression performed best, LDA performed better than all other methods on all datasets. Many of the other methods appear to be overfitting, which may be why their performance on the testing data is worse than LDA. For example, since QDA has almost twice as many parameters as LDA, it may be overfitting. Likewise, Logistic Regression is more general than LDA and can require about 30% more data to obtain the same fit [9].

Although it is hard to rank methods other than LDA and HMM, it should be noted that Naive Bayes performed better with 421-component vectors than with 21-component vectors. This is apparent both in Figure 5 and in Figure 4. Of course, more parameters can always lead to a better fit on the training data, and with only 1000 to 1500 training points, Naive Bayes is in danger of overfitting when used with 421-component vectors. So, it is surprising that it performs well on the testing data, living up to its reputation of performing well in high-dimensional spaces. It may be possible to further improve the performance of Naive Bayes by using more-complex prior distributions for the individual features. At present, we use binary distributions. Since each feature represents an amino-acid count, a binary distribution effectively means that each amino acid is modeled as being either present or absent in a peptide. This represents a loss of information, which more-complex prior distributions could eliminate.

To our surprise, Hidden Markov Models performed the worst of all the classification methods: sometimes even worse than randomly guessing. This is despite the fact that they work directly on the peptide sequences, and not on vectors with lower information content. One reason seems obvious from the ROC curves in Figure 2. These show that performance generally degrades as the number of states in the HMM increases. This is a sure sign of overfitting. In fact, a HMM with 10 states will have $10 + 10^2 + 10 \times 20 = 310$ parameters. The factor of 20 comes from the 20 amino acids that each of the 10 states must be able to output. Since the classes in our training data often have only about 300 samples, over fitting is surely taking place. Even with only 5 states, the number of parameters is $5 + 5^2 + 5 \times 20 = 130$, which is still too many. In addition to this, HMMs only capture the *proportion* of amino acids in a peptide sequence. In particular, with only a small number of states, they cannot capture the

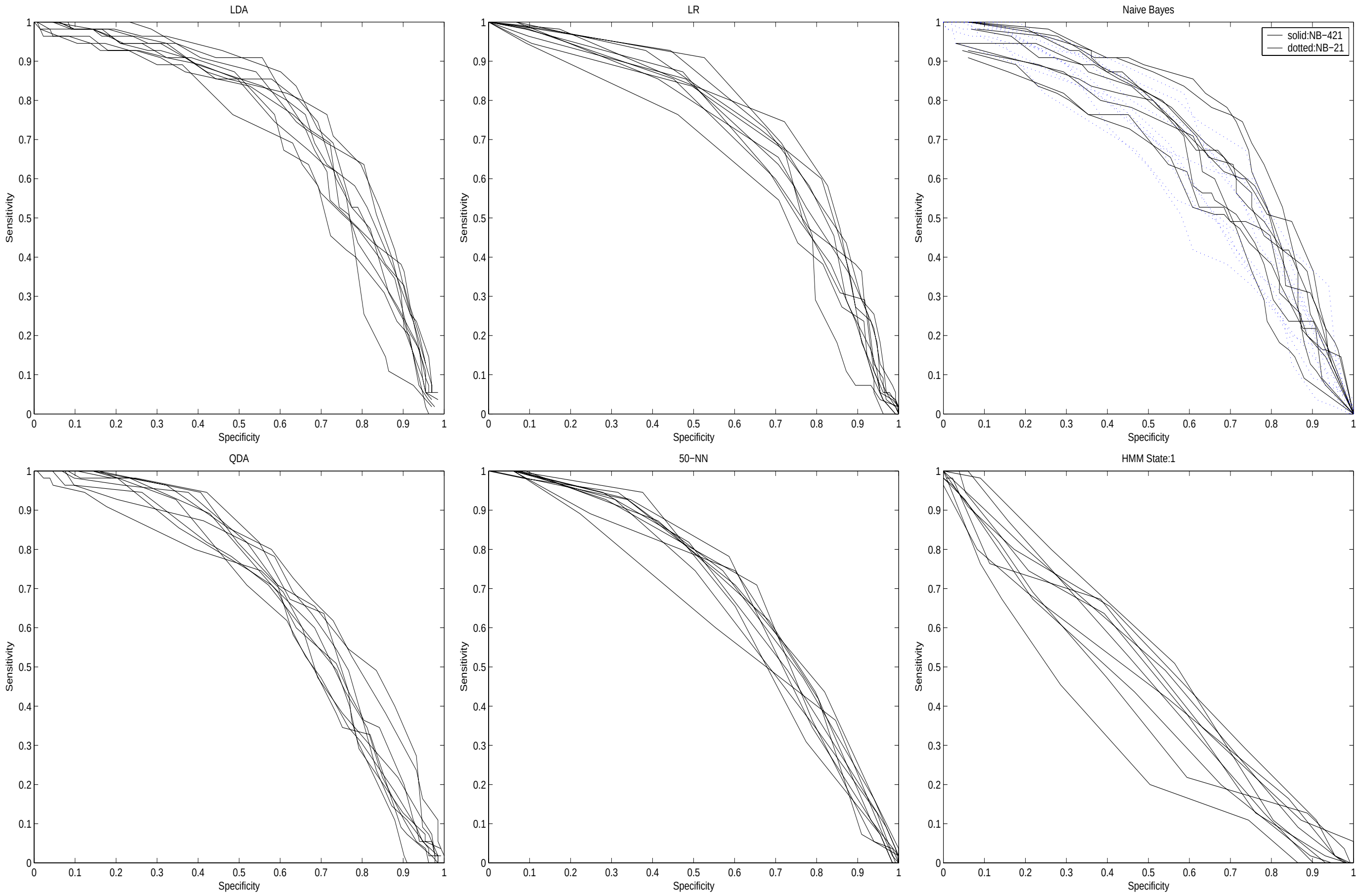

Fig. 4. Results of the cross-validation experiments on all the discrimination methods.

TABLE II

TEST-SET ERROR RATES. MISCLASSIFICATION RATES FOR SIX DISCRIMINATION METHODS APPLIED TO THREE DATASETS, EACH GENERATED IN TWO DIFFERENT WAYS.

	Mouse	Brain	Mouse	Heart	Mouse	Kidney
	MAX-MIN	STD-DEV	MAX-MIN	STD-DEV	MAX-MIN	STD-DEV
Linear Discriminant Analysis	0.3316	0.3446	0.3192	0.3178	0.3080	0.3218
Quadratic Discriminant Analysis	0.3498	0.3635	0.3542	0.3632	0.3398	0.3744
Logistic Regression	0.3632	0.3544	0.3336	0.3517	0.3237	0.3272
K-Nearest Neighbour Classifier	0.3641	0.3633	0.3651	0.3528	0.4056	0.3681
Naive Bayes Classifier	0.4078	0.3794	0.3880	0.3832	0.3775	0.3397
Hidden Markov Models	0.4065	0.4235	0.3808	0.3791	0.3492	0.3352

absolute number of amino-acid occurrences, as our vectors do. It seems reasonable to suppose that the actual presence of a particular number of certain acids may be an important factor in the ionization of peptides. In fact, in a separate experiment, we used vectors whose components represented amino-acid proportions, instead of absolute numbers, and this caused the performance of Logistic Regression to decrease greatly, to about the same level as HMMs.

B. Classification Error Rate

In general, classification error rates depend on the size of the labeled classes, and can be deceptively low when the classes have very different sizes (as ours do, since most peptide peaks are low-intensity, not high-intensity). Even random guessing can appear to produce very low error rates. To compensate for this, we used the ROC data to estimate classification error rates for classes of equal size. This is easily done with the following formula

$$\text{error rate} \;=\; 1 - \frac{\text{specificity} + \text{sensitivity}}{2} \qquad (1)$$

We applied this formula to every point on every ROC curve in Figure 3. For each curve, we chose the lowest resulting error rate, to indicate the best possible performance for the classifier on the dataset. Table II shows the resulting classification error rates for each of the six classifiers on each of the three different datasets, with both MAX-MIN and STD-DEV used to choose

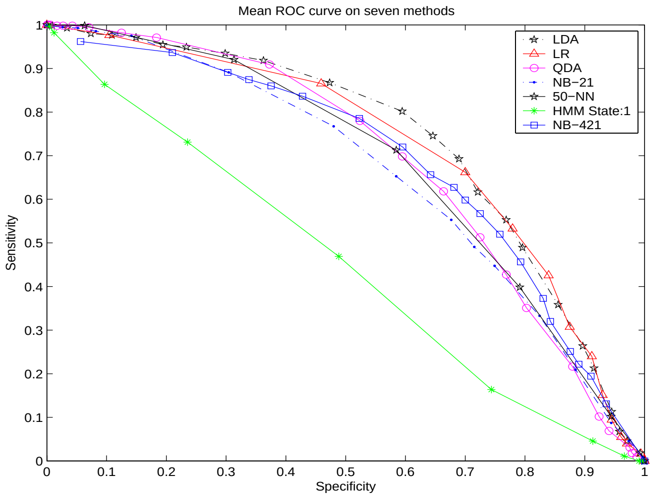

Fig. 5. Mean ROC curves for the seven methods

proteins. The table shows that the classification error rates range from 0.3 to 0.4. In addition, it corroborates some of the results discussed above. For instance, LDA performs the best across all datasets, while HMMs perform the worst.

VI. Conclusions and Future Work

In this paper, we applied several well-known classification methods to Peptide Tandem Mass Spectrometry data. We evaluated and compared the methods based on their ability to predict the intensity of peaks in a mass spectrum based on the amino-acid sequence of peptides. Specifically, the methods were required to divide peptides into two classes: those that produce high-intensity peaks, and those that produce low-intensity peaks.

Overall, we found that for our datasets, simple classifiers such as LDA performed well compared with more sophisticated ones, such as QDA and Hidden Markov Models. In the main comparison based on ROC evaluation, LDA has the best generalization performance. In decreasing order of performance, the other classifiers were Logistic regression, Naive Bayes (with 421-component vectors), QDA, KNN (with K=50), Naive Bayes (with 21-component vectors), and finally HMMs (with any number of states), which sometimes performed worse than random guessing. The performance of all the methods was quite variable. Misclassification rates for classes of equal size were also estimated, and they corroborate our conclusions drawn from the ROC evaluation.

Several further steps suggest themselves. For instance, the performance of each of the methods might be improved by boosting. Decision trees (which often work well with boosting) could be tried. Clustering methods could also be used to search for structure within each of the labeled classes, in order to provide better priors for the classification methods. In addition, Bayesian inference, bootstrapping and bagging could be used to deal with the problems of overfitting. To prevent over fitting of Hidden Markov Models, we are considering more-structured versions of them, such as Profile Hidden Markov Models, which have significantly fewer parameters. Finally, we are considering methods for extracting more training data from the MS/MS spectra. In this paper, we considered two methods, MAX-MIN and STD-DEV, but many other methods are also possible.

Acknowledgment

The authors would like to thank Dr. Andrew Emili for his comments and discussions and for generous access to his data. We would also like to thank Clement Chung for generating and preprocessing the data. Mr Liu is supported by the University of Toronto.

References

[1] Aebersold, R., Mann, M. *Mass spectrometry-based proteomics*, Nature 422, pp. 198-207, 2003.

[2] Bonner, A., Liu, H. *Development and Evaluation of Methods for Predicting Protein Levels and Peptide Peak Intensities from Tandem Mass Spectrometry Data*. Submitted for publication. Available at `www.cs.toronto.edu/˜bonner/papers.html`.

[3] Bonner, A., Liu, H. *Canonical Correlation, an Approximation, and its Application to the Mining of Tandem Mass Spectrometry Data*. Submitted for publication. Available at `www.cs.toronto.edu/˜bonner/papers.html`.

[4] Duda, R., Hart, P., Stork, D. *Pattern Classification*, second edition. Wiley-Interscience, 2001.

[5] Dudoit, S., Fridlyand, J., and Speed, T.P. *Comparison of discrimination methods for the classification of tumors using gene expression data.* Journal of the American Statistical Association 97(457), 77-87.

[6] Elias, J.E., Gibbons, F.D., King, O.D., Roth, F., Gygi, S.P. *Intensity-based protein identification by machine learning from a library of tandem mass spectra*. Nature, biotechnology, Volume 22 Number 2, 2004.

[7] Fisher, R.A. *The use of multiple measurements in taxonomic problems*. Annals of Eugenics, 7, 179-188.

[8] Gay, S. ,Binz, P.A., Hochstrasser, D.F., Appel, R.D. *Peptide mass fingerprinting peak intensity prediction: extracting knowledge from spectra.* Proteomics 2, pp. 1374-1391, 2002.

[9] Hastie, H., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Series in Statistics, 2001.

[10] Jordan, M. *Why the Logistic Function? A tutorial discussion on probabilities and neural networks*. Computational Cognitive Science Technical Report 9503, Massachusetts Institute of Technology, 1995.

[11] Kussmann, M., Nordhoff,E., Rahbek-Nielsen, H., Haebel, S., et al. Journal of Mass Spectrom 32, pp. 593-601, 1997.

[12] Mann, M., Hendrickson, R.C., Pandey, A. *Analysis of proteomes by mass spectrometry*. Annu. Rew. Biochem. 70, pp. 437-473, 2001.

[13] Mardia, K.V., J.T.Kent, and J.M.Bibby *Multivariate Analysis*. Academic Press, Inc., San Diego, 1979.

[14] McLachlan, G.J. *Discriminant analysis and Statistical pattern recognition*. Wiley, New York. 1992.

[15] Pachter, L., Alexandersson, M., Cawley, S. Applications of Hidden Markov Models to Alignment and Gene Finding Problems. *Proceedings of the Fifth Annual Conference on Computational Biology*, pp. 241–248, 2001. ACM Press.

[16] Pandey, A., Mann, M. *Proteomics to study genes and genomes*. Nature 405, pp. 837-846, 2000.

[17] Poritz, A.B. Linear Predictive Hidden Markov Models and the Speech Signal. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1291–1294, 1982.

[18] Purves, R. W., Gabryelski, W., Li, L., Rapic. Commun. Mass Spectrom 2, pp. 695-700, 1998.

[19] Rice, J. *Mathematical Statistics and Data Analysis*, second edition. Duxbury Press, 1995.

[20] Ripley, B.D. *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, New York, 1996

[21] Thosma, K., Khaled, R., Dragan, Radulovic. *PRIM, a Generic Large Scale Proteomics investigation Strategy for Mammals*. Molecular & Cellular Proteomics 2.1, 2003.