Mining mass-spectra for diagnosis and biomarker discovery of cerebral accidents.

Julien Prados[1], Alexandros Kalousis[1],
Jean-Charles Sanchez[2], Laure Allard[2], Odile Carrette[2], and Melanie Hilario[1]

[1] University of Geneva, Department of Computer Science, Rue General Dufour, 24, 1211, Geneva, Switzerland {prados,kalousis,hilario}@cui.unige.ch
[2] Biomedical Proteomics Research Group, Central Clinical Chemistry Laboratory {Jean-Charles.Sanchez, Laure.Allard, Odile.Carrete}@sim.hcuge.ch

Abstract. In this paper we to try to identify potential biomarkers for early stroke diagnosis using SELDI mass-spectrometry coupled with analysis tools from machine learning and data-mining. Data consist of 42 specimen samples, ie, mass-spectra divided to two big categories, stroke and control specimens. Among the stroke specimens two further categories exist that correspond to ischemic and hemorrhagic stroke; in this paper we limit our data analysis to discriminating between control and stroke specimens. We performed two suites of experiments. In the first one we simply applied a number of different machine learning algorithms; in the second one we have chosen the best performing algorithm as it was determined from the first phase and coupled it with a number of different feature selection methods. The reason for that was two-fold, first to establish whether feature selection can indeed improve performance, which in our case it did not seem to confirm, but more importantly to acquire a small list of potentially interesting biomarkers. Of the different methods explored the most promising one was Support Vector Machines which gave us high levels of sensitivity and specificity. Finally analyzing the models constructed by Support Vector Machines we produced a small set of 13 features that could be used as potential biomarkers, and which exhibited very good performance both in terms of sensitivity and specificity.

Key words:stroke, biomarker discovery, data mining, support vector machines, feature selection, model stability.

1 Introduction

This paper is concerned with the first stage of protein biomarker discovery-validation, namely exploratory data driven discovery of protein profiles that appear to distinguish stroke from control specimens. Blood samples of the individuals participating in the study were submitted to mass spectrometry. The resulted spectra underwent a systematic preprocessing phase in order to acquire the appropriate data for analysis. Special care was given in the identification of peaks and mass clustering and a systematic procedure for performing that task is presented. Once the preprocessed data were available we undertook a systematic study of well known data mining algorithms on the given problem. The diagnostic power of the models build was high, however due to the high dimensionality of the input space the models were not easy to translate. In order to do so we examined a number of feature selection algorithms with the aim to reduce the dimensionality of the input space while preserving the good predictive performance.

One of the problems that we faced is that since we used resampling procedures to estimate the predictive performance of the algorithms we had a number of different models produced during evaluation. In order to come up with a final set of suggested biomarkers we had to fuse these models. However before even trying to do so we had to show that the produced models were not sensitive to perturbations of the training set, i.e., they were relatively stable with respect to different training data, so that their fusion would make sense. We devised a procedure to measure the stability of these models, and after showing that it was quite high we combined the suggestions of the individual models to come up with the final set of biomarkers.

2 Study population and sample handling

Forty-two consecutive patients admitted to the Geneva University Hospital emergency unit were enrolled in this study. The local institutional ethical committee board approved this study. Each patient or patient's relatives gave informed consent prior to enrollment. For each patient, a blood sample was collected at the time of admission in dry heparin-containing tubes. Of the 42 patients enrolled, 21 were diagnosed with orthopedic disorders (without any known peripheral or central nervous system condition) and classified as control samples (including 12 men and 9 women, aged 69.5 years, range 34-94 years) and 21 were diagnosed with stroke (11 men, 9 women and 1 unknown, aged 61.95 years, ranging from 27 to 87 years.) including 11 ischemic and 10 hemorrhagic. After centrifugation at 1500g for 15 minutes at 4C, plasma samples were aliquoted and stored at -70C until analysis. For the patients of the stroke group, the time interval between the neurological event and the first blood draw was 185 minutes (ranging from 40 minutes to 3 days). The diagnosis of stroke was established by a trained neurologist and was based on the sudden appearance of a focal neurological deficit and the subsequent delineation of a lesion consistent with the symptoms on brain CT or MRI images, with the exception of transient ischemic attacks (TIAs) where a visible lesion was not required for the diagnosis. The stroke group was separated according to the type of stroke (ischemia or haemorrhage), the location of the lesion (brainstem or hemisphere) and the clinical evolution over time (TIA when complete recovery occurred within 24 hours, or established stroke when the neurological deficit was still present after 24 hours).

3 Preparation of SELDI ProteinChips

The Strong Anion Exchange arrays (SAX2 ProteinChip©,Ciphergen Biosystems, Fremont, CA, USA) were used as a first fractionation step of the plasma samples. SAX2 spots were first outlined with a hydrophobic pap-pen and air-dried. Chips were then equilibrated 3 times during 5 min with $10\mu L$ binding buffer (20 mM Tris, 5 mM NaCl, pH9.0) in a humidity chamber at room temperature. Two μL of binding buffer was applied on each spot and $1\mu L$ of crude (stroke or control) plasma samples was added to each spot and incubated 30 min in a humidity chamber at room temperature. Plasma was removed and each spot was individually washed 5 times 5 min with $5\mu L$ of binding buffer followed by 2 quick washes of the chip with deionised water. Excess of H_2O was removed and while the surface was still moist, $0.5\mu L$ of sinapinic acid (SPA, Ciphergen) in 50% (v/v) acetonitrile and 0.5% (v/v) trifluoroacetic acid was added twice per spot and dried. The arrays were then read in a Protein-Chip reader system, PBS II sery (Ciphergen Biosystems, Fremont, CA, USA). The ionized molecules were detected and their molecular masses determined according to their time-of-flight (TOF). TOF mass spectra, collected in the positive ion mode were generated using an average of 65 laser shots throughout the spot at a laser power set slightly above threshold (10-15% higher than the threshold). Spectra were collected and analysed using the Ciphergen Proteinchip software (version 3.0) [1, 2]. External calibration of the reader was performed using the "all-in-1" peptide molecular weight standards (Ciphergen Biosystems, Inc.) diluted in the SPA matrix (1:1, vol/vol) and directly applied onto a well of a normal phase chip.

4 Data Preparation

Each spectrum consists of 28351 data points of the form (mass/charge (m/z), intensity), with the m/z ratio ranging from 8 to 68600 Daltons (within the text the terms m/z, mass/charge and mass, will be used in indistinguishable manner). Analysis is further constrained to m/z values bigger than 1 kilodalton resulting in 24901 data points. Intensity values lower than this threshold where not considered due to the distortion caused by the matrix molecules. Biologists performed baseline removal and spectrum normalization with the aid of the Ciphergen ProteinChip Software. Spectra normalization was done with total ion current using a number of manually detected peaks. Then the Ciphergen ProteinChip Software was used to first identify peaks on each one of the available spectra (section 4.1), which should be clustered in order to determine the distinct ones (section 4.2).

4.1 Peak Detection

Peak detection is an effort to further reduce the dimensionality of the problem. It is a critical step on the outcome of which depends heavily the quality of the final results. The detected peaks will provide the basis for the construction of the final variables that will describe the spectra. Obviously variables of poor quality will produce poor results.

Peak detection was done within the Ciphergen ProteinChip Software. We used the software in order to determine a list of peaks for each spectrum. This was done for a single spectrum each time without taking into account the remaining spectra; the final outcome was a list of peaks for each spectrum.

The peak detection process accepts two parameters: *valley depth* and *height*, both used to control different aspects of the signal to noise ratio. The first one indicates how many times higher than the noise level should be the depth of the valley between two consecutive peaks, while the second indicates how many times higher than the noise level should be the height of a peak. Appropriate adjustment of these parameters gives rise to a different number of detected peaks per spectrum. We experimented with a number of different values of these parameters, setting them manually through the Ciphergen Protein Chip software, and produced different descriptions–datasets–for the problem. The names of the datasets follow the format vdX_hY, where X, and Y, are the values set for the valley depth and height parameters. The detailed results on the total number of detected peaks and the average number of peaks per spectrum for each parameter setting are given in table 1. The *default* entry denotes the setting of the parameters given as default by the software. *Manual* indicates a description of the problem that was the result of the intervention of the domain experts in order to define an initial set of peaks; the domain experts visually inspected all the 42 samples and identified manually points in the spectra that they considered to be peaks on a case by case basis. The reason behind this extensive experimentation with different values of the parameters is to acquire an initial understanding of the behavior of the used methods to different signal to noise ratios. If we allow for low values of that ratio we would detect more peaks some of them being possibly part of the noise. Allowing only for high values of the signal to noise ratio will produce fewer peaks but might result in loss of valuable information.

In a next step we will identify which peaks among the different spectra correspond to the same mass based on their mass distance.

4.2 Mass Clustering

Each detected peak corresponds either to a unique protein with the given m/z ratio or possibly to several proteins that share the same m/z ratio. The idea is to find which of the detected peaks among the different spectra correspond to the same m/z ratio. The problem is complicated by the measurement error, m_{err}, of the apparatus which is around $\pm 0.05\% - 0.03\%$ of the measured m/z ratio ($m/z \pm m/z \times m_{err}$). Using as a starting point the lists of peaks produced by the peak detection process we have to produce a list of unique features, each one corresponding to a m/z ratio, that will be used to describe all the spectra in a uniform manner. The idea is to group together into a single variable all those peaks that correspond to the same m/z value, i.e., all those peaks whose m/z ratios have a distance which is smaller than twice the mass measurement error,

Table 1. Results of the different peak detection settings for the *valley depth (vd),* and *height (h),* parameters.

datasets	Number of detected peaks	Number of distinct masses
manual	1001	33
vd10_h10	486	52
vd7_h7	675	75
vd6_h6(default)	788	86
vd4_h4	1126	123
vd3_h3	1482	154
vd2_h2	2441	256
vd1_h1	8950	681

i.e. $2 \times m_{err}$, of the apparatus. Under that scenario two masses $m/z_a, m/z_b$, will be considered as the same masses if the corresponding intervals $[m/z_a \pm m/z_a \times m_{err}], [m/z_b \pm m/z_b \times m_{err}]$ have an overlap. The variables constructed from that procedure will provide the description of each spectrum; wrong decisions on what is different and what is the same can have a great impact on the final results both in terms of diagnostic performance and the discovered biomarkers.

To determine which peaks correspond to the same mass and which are distinct we applied a hierarchical clustering procedure, [4], based only on the m/z values of the detected peaks. Furthermore due to the special nature of the problem some additional constraints should be imposed. Before proceeding to further details of the algorithm we will explain how we measure the distance between two individual masses (clustering algorithms are usually based on some notion of distance of the instances that should be clustered). The idea is to express mass distances relatively to the mass scale so that they can be directly comparable with the $2 \times m_{err}$. We decided to use the following distance measure between two masses m_1, m_2:

$$d(m_1, m_2) = \frac{|m_1 - m_2|}{\mu},\ \mu = (m_1 + m_2)/2.$$

where the distance of two masses is expressed relatively to their mean, a measure which is on the same scale as m_{err}.

For a hierarchical clustering algorithm to be completely defined one has to provide a measure of the distance between sets of instances in our cases sets of masses. We decided to represent a cluster of masses simply by the average of the masses it includes and the distance between two clusters of masses, C_1, C_2, simply as the $d(\mu_{C_1}, \mu_{C_2})$ distance of the corresponding averages. In essence we are performing *centroid linkage* based hierarchical clustering. The complete clustering procedure together with the appropriate constraints are given in algorithm 1.

The definition of the clustering procedure is not yet completed, we have to give the additional constraints imposed by the nature of the specific problem. First, clusters can be merged only if their distance is less than m_{err} (the first condition of the while loop). Since each final cluster contains small perturbations of a given mass we should not group together masses that in reality correspond to different masses. Second, if two clusters have been identified as possible candidates for merging, i.e., $d(C_1, C_2) \leq 2 \times m_{err}$, merging will only take place if the two farthest elements of the clusters have a distance which is smaller than $2 \times m_{err}$ (the second condition of the while loop). This constraint also covers the case of merging together masses that come from the same spectrum; since the distance between any two masses from the same spectrum would be more than $2 \times m_{err}$ this type of merging will not be allowed either. However there have been cases of spectra, when low signal to noise ratios was used in peak detection, where this condition was not true, i.e., the software detected peaks among the same spectrum with a mass distance smaller than $2 \times m_{err}$. We have chosen to keep all the cases like that and not allow their merging either, this is why sometimes some feature sets might contain masses that have a distance which is smaller than $2 \times m_{err}$.

Among the possible candidate pairs for merging that satisfy the constraints the algorithm chooses the one that has the minimum distance (third condition of the while loop). In figure 1 we present a schematic example of the situations that may appear, for the specific configuration of masses the algorithm would terminate at the sixth step since there are no more masses to merge.

When there are no more clusters to merge the algorithm simply returns the list of remaining clusters, C. Each of them will correspond to a specific mass/charge ratio, the mean of the mass/charge ratios found in it. Every cluster, C_i, will now become a feature of the description of our spectra. In the next section we will see how we assign values to these features for each of the spectra.

To summarize the first two preprocessing steps: different signal to noise trade offs (table 1) result in peak sets of varying sizes. The mass/charge values in each of these peak sets are clustered according to the algorithm given above, resulting in different feature sets, again of different size (their exact cardinality is given in *Number of distinct masses* of table 1).

Algorithm 1 MassCluster(L)

{L : list of masses m_i from all the spectra}
$C_i \leftarrow m_i, m_i \in L$
$C \leftarrow \{C_i\}$
while Exist $C_l, C_k, \in C$ with $d(\mu_{C_l}, \mu_{C_k}) \leq 2 \times m_{err}$ AND $argmax_{m_l,m_k} d(m_l, m_k) \leq 2 \times m_{err}$, $m_l \in C_l$, $m_k \in C_k$ AND $d(\mu_{C_l}, \mu_{C_k}) = argmin_{C_i,C_j}(d(\mu_{C_i}, \mu_{C_j}))$ **do**
  merge(C_l, C_k)
end while
return C

4.3 Intensity Values

One final issue that had to be addressed is how the values of the C_i features determined by the clustering algorithms were going to be calculated for a given spectrum (these values will correspond to the intensity of the corresponding peaks). The answer is obvious for all the clusters that are associated with one of the detected peaks in the spectrum, but less so for these clusters that do not have an associated peak, i.e., there was no peak detected in that m/z range for the given spectrum. The problem is that each spectrum is potentially described by a different subset of features of C. There is a number of possible options like assigning an indicator of non-applicability or an indicator of missing values. The first is more appropriate but it is not straightforward since most of the standard learning algorithms do not offer that possibility. The option of missing value is less appropriate because it does not really match the semantics of the problem. Another alternative would be to actually fill the values of the absent features C_i, either with the value of zero which could mean that the intensity of the corresponding peak is zero, a logical assumption since absence can be interpreted as an indication of zero intensity, or with a value taken directly from the spectrum within the *close neighborhood* of C_i, i.e. within the interval $C_i \pm C_i \times m_{err}$. We have opted for the second option because we think it is more robust, and the value that we assign is the maximum intensity over the close neighborhood. Having the actual intensity values means that the learning algorithm will less prone to errors introduced by the peak detection procedure. This is not possible when we assign a value of zero on the intensities of the non-detected masses since the intensity information is lost.

In an alternative direction one can consider a completely different family of learning algorithms that assume a different representation paradigm, namely relational learning algorithms. This type of algorithms allow for representations of learning paradigms that are of variable length. However they do not fall within the scope of the current paper.

4.4 Intensity Normalization

The above procedures gave rise to a fixed-length attribute-value representation of our spectra. Each spectrum is described by the set C of C_i features where each feature corresponds to a specific mass/charge ratio and the value of the feature is the intensity of the spectrum in the close neighborhood of the given ratio. Since different ratios have different domains of values (minimum and maximum values of intensities differ radically between low and high ratios of mass/charge) we had to scale them to the same interval. We applied a simple scaling where the values of each feature C_i where normalized by its corresponding maximum, i.e. $C'_i = C_i/max(C_i)$, the new values will be in the $[0, 1]$ interval. These algorithms based on distance measures or dot products, like the nearest neighbor algorithm, support vector machines and multilayer perceptrons, will not be affected by the different scales of the variables.

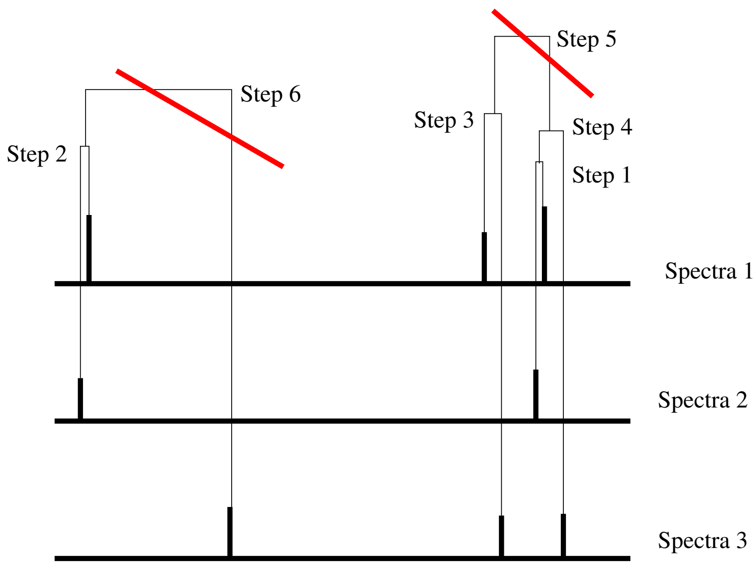

Fig. 1. Example of mass clustering. The numbered steps indicate the sequence of the merging steps. Steps 5 and 6 are not allowed because they violate domain constrains. Step 5 because it would put into the same cluster two masses that have a distance that is bigger than $2 \times m_{err}$; moreover it would have placed together two distinct masses from the same spectrum (spectrum 1), and step 6 because the distance of the two clusters exceeds the $2 \times m_{err}$ threshold.

5 Learning with mass-spectra

The learning experiments can be distinguished to two suites of experiments. In the first one we applied a series of algorithms on the eight available datasets (section 5.1). In the second suite we explored feature selection in order to see whether we can improve our classification performance and on the same time acquire models based on smaller feature sets which are easier to explore (section 5.2).

All the evaluations of performance were done using ten-fold cross validation. Control of the statistical significance of the differences between the learning algorithms was done using McNemar's test with the p value set to 0.05. Furthermore since we were comparing different learning algorithms we had to establish a ranking schema based on their relative performance as this was determined by the results of the significance tests. The procedure we followed was: in a given dataset every time an algorithm, a, was significantly better than another algorithm b then a was credited with one point and b with zero points, if there was no significance difference between the two algorithms then both were credited with half point. If one algorithm is significantly better than all the others then it will get $n-1$ points, where n is the total number of algorithms being compared, while if there is no significance difference between the algorithms then each one will get $(n-1)/2$ points.

We have to note here that with such a small sample it is very difficult to get significant differences between the algorithms; in some cases the test did not signal a significant difference even though one of the algorithms had more than double the error of the other. In some sense the test of significance we used was quite conservative in detecting significant differences.

5.1 Learning Algorithms and Parameters

We experimented with a number of different classification algorithms trying to cover a variety of different learning approaches. Moreover for each one of them we did not rely on the default parameters settings but explored a number of them. We used one decision tree algorithm J48, [5, 6], with three different values for the M parameter, M=2,5,7, a parameter that controls the minimum number of examples allowed in each leaf node of the decision tree, in one sense it controls the complexity of the model, higher values mean simple and more general models; a nearest neighbor algorithm, IBL, [4], with the number of nearest neighbors, k, varying $k = 1, 3, 5$, low values of k correspond to complex and highly variant models similarly to low values of the M parameter; a support vector machine algorithm, SVM, with a simple linear kernel and the value of the C parameter being C=0.5, 1, 2, [7], and a multilayer perceptron, MLP, of a single layer of ten hidden units [8]. The implementations of the algorithms were the ones of the WEKA machine learning environment, [5].

Table 2. Algorithms and their explored settings

algorithm	Parameter	Value
SVM	C	0.5,1,2
J48	M	2,5,7
IBL	K	1,3,5
MLP	-	-

Base Learning Results Each of the learning algorithms was applied to each one of the eight datasets given in table 1, for each one of its parameter settings given in table 2. Overall the number of base experiments was 80.

We do not present the complete results of each parameter setting but only the results of the best setting for each algorithm over the eight datasets, table 3. To get a better picture of the relative performance of the algorithms we also give graphically the error evaluation results, figure 2. What is immediately evident is the bad performance of the decision trees algorithm. In almost all the different datasets it is the worst classification algorithm. In terms of its ranking it is never significantly better than any other algorithm and it is once significantly worse than two (*vd1_h1, SVM, MLP*) and twice significantly worst than one (*vd2_h2-MLP, vd10_h10-SVM*). There are more datasets in which J48 has more than double the error of other algorithms but the test did not signal a significant difference probably due to the small number of available instances as mentioned earlier.

One of the surprising results is that the *manual* dataset (the dataset in which the domain experts defined the set of peaks based on the visual examination of the spectra) is the one that shows the lowest performance among the eight examined datasets.

We will now take a closer look at the sensitivity and specificity performance of SVM since it was the algorithm that achieved not only the best average performance among all the different versions of the data sets, but it was also the one whose performance exhibited the smallest variance. Sensitivity in this application problem will be the number of detected strokes over the total number of strokes, while specificity the number of detected controls over the total number of controls (complete results in table 4). We get excellent results for sensitivity, actually in five of the eight datasets it is 100% and in the remaining three it is between 95% and 90%, resulting in an average of 97%. However the specificity is lower and the number of control samples misclassified as stroke ranges from three up to nine resulting in values of specificity between 57% and 86%, with an average of 71%. The fact that the performance of SVM is good and stable over the different versions of the datasets is an indication that we are dealing with a problem in which one has to closely examine more than one variable in the same time in order to make a classification of a sample.

We have to note here that the results reported are based on cross validation which means that the created models are always tested on samples which have not been used in the construction of the classification model. Error in the training set is much smaller some times even zero but it should never be used as an indicator of performance since it is always overly optimistic.

Table 3. Estimated errors in base experiments, the numbers in parentheses are the scores that the algorithms achieve for a given dataset, see details in text.

dataset	IBL-5	J48-5	SVM-0.5	MLP	Average
manual	30.95(1.5)	28.57(1.5)	21.42(1.5)	28.57(1.5)	27.38
vd10_h10	21.42(1.5)	30.95(1.0)	14.28(2.0)	19.04(1.5)	21.42
vd7_h7	21.42(1.5)	33.33(1.5)	16.66 (1.5)	21.42(1.5)	21.42
vd6_h6(default)	16.66(1.5)	28.57(1.5)	16.66 (1.5)	21.42(1.5)	22.61
vd4_h4	21.42(1.5)	33.33(1.5)	14.28(1.5)	19.04(1.5)	23.21
vd3_h3	26.19(1.5)	33.33(1.5)	19.04(1.5)	14.28(1.5)	22.02
vd2_h2	30.95(1.0)	33.33(1.0)	14.28(1.5)	11.90(2.5)	20.83
vd1_h1	26.19(1.0)	33.33(1.0)	11.90(2.0)	14.28(2.0)	32.14
Average	24.40	31.84	16.07	18.75	

Table 4. Specificity and Sensitivity results of SVM on the base experiments. Cells of the tables contain counts.

		manual		vd10_h10		vd7_h7		vd6_h6	
		Ctrl	Strk	Ctrl	Strk	Ctrl	Strk	Ctrl	Strk
True	Ctrl	12	9	15	6	14	7	14	7
	Strk	0	21	0	21	0	21	0	21
		Predicted							

		vd4_h4		vd3_h3		vd2_h2		vd1_h1	
		Ctrl	Strk	Ctrl	Strk	Ctrl	Strk	Ctrl	Strk
True	Ctrl	15	6	14	7	17	4	18	3
	Strk	0	21	1	20	2	19	2	19
		Predicted							

We will take a closer look on the behavior of the SVM among the different datasets in what concerns its specificity, i.e. the number of control instances that are wrongly classified as stroke. In table 5 we give the control instances that are wrongly misclassified as stroke among the different datasets. There are two instances, 34, 30, which are systematically misclassified among all the datasets; two which are misclassified in seven out of the eight datasets, and the remaining range from six misclassifications down to one. In order to have a more precise idea of why these instances are misclassified we will take a look to a specific dataset, *vd6_h6*, and see the values of the linear function produced by

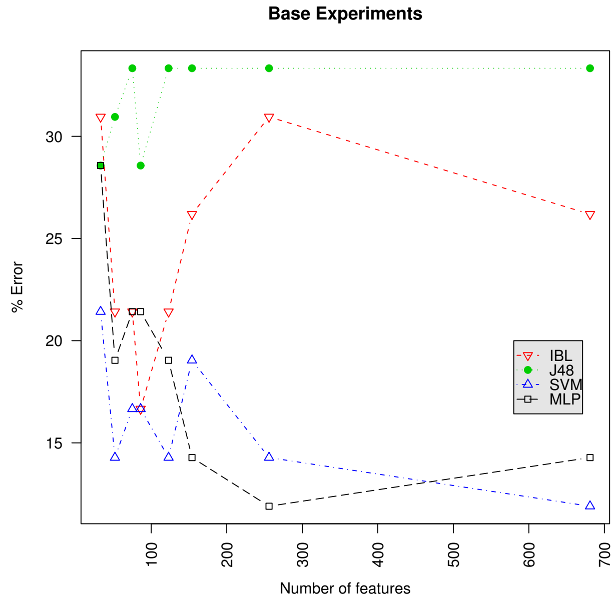

Fig. 2. Errors of the learning algorithms on the eight different datasets traced with respect to the number of features of the initial datasets.

SVM for each instance when that instance was a part of a fold test. Remember here that since we are using ten fold cross-validation to perform error estimation we have ten different learned models one for each separation to train and test sets. Graph 3 give us for each fold I of the cross validation the values of the linear function, learned on the train set of the Ith fold, when applied to each one of the instances of the corresponding test set. When the value of the linear function on a given instance is higher than zero then that instance is classified as stroke, otherwise it is classified as control. From the graph we can see that the most problematic instances are 34, 30 and 38 which had output values that were much further than the decision surface. The remaining four instances were very close to the decision boundary and they can be considered as near misses. It remains to be seen what are the particularities of these three control samples that place them so far and on the wrong side of the decision surface.

5.2 Feature Selection Experiments

In order to examine whether it is possible to further improve the predictive performance of the SVMs we also examined a number of feature selection algorithms. Even if we do not manage to improve performance but keep it at the same levels, having smaller feature sets would give us a better understanding of what factors are important in determining stroke or no-stroke. Error evaluation was done with feature selection as a part of the cross-validation loop. That is, for each fold we first applied feature selection and then the learning algorithm on the selected features. Alternatively feature selection could be done only once in a preprocessing step but this would optimistically bias the results of the error

Table 5. Control instances that were wrongly classified as stroke by SVM among the different datasets.

	Instances Ids													
dataset	34	30	23	38	40	29	33	5	28	3	41	36	27	8
manual	x	x	x	x	x	x					x	x	x	
v10_d10	x	x	x	x	x	x								
vd7_d7	x	x	x	x	x	x	x							
vd6_d6	x	x	x		x	x	x		x					
vd4_d4	x	x	x	x	x		x							
vd3_h3	x	x	x	x	x	x	x							x
vd2_d2	x	x	x	x				x						
vd1_d1	x	x		x				x		x				

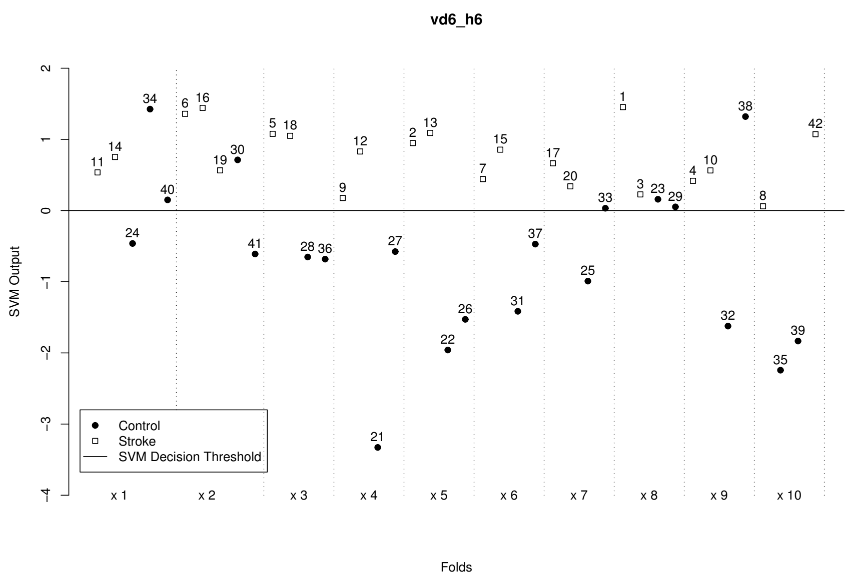

Fig. 3. Each one of the xI partitions of the graph corresponds to the Ith fold of the cross validation. Within it we see the test instances that were associated with the test set of that fold and their output values as they were determined by the linear model produced by the SMV algorithm when trained on the train set of the Ith fold.

evaluation, since the whole data would have been used to provide a part of the model, in this case the selected features.

We experimented with three different feature selection algorithms, Information Gain based feature selection (IG) [4], Relief-F (RF) [9], SVM based feature selection (SVMfs). They follow completely different paradigms of feature selection. In information gain features are selected on the basis of their mutual information with respect to the target variable. It is a univariate feature selection method and not able to capture feature interactions; moreover it can result in feature sets that contain many correlated features, ie, redundant features, which happen to have high score with the class. The Relief-F algorithm is able to better capture feature interactions and is based on the notion of nearest neighbor classification; features which help to predict the class correctly get a high score while features that do not discriminate or lead to false predictions get a low score. In SVM based feature selection a simple linear kernel is used to construct a classification model; based on that model features that get high coefficients are considered of high importance (this is true when all features are scaled to the same interval).

All the methods can be used either with a threshold, ie, select all features that get a score higher than the threshold, or to select a given number N of features. We have opted for the second choice since apriori we did not have any idea of what could be a good value for a threshold. We have chosen to set the number of selected features to $N = 15$, a number of features which was considered acceptable from the domain experts. IG though had a problem since the features for which it was assigning a score more than zero were always less than 15 so for this algorithm we used instead a threshold set to zero.

Feature Selection Results Overall the results of feature selection are rather disheartening. The complete results are given in table 6 and figure 4. All feature selection methods apart from SVMfs significantly harmed the predictive performance. In the case of SVMfs the performance in average overall the datasets was also damaged. However there were two datasets in which the performance of feature selection was comparable with the performance on the complete set, namely *vd10_h10* where there was a small deterioration of the predictive error, and *vd7_h7* where there was a small improvement, the corresponding estimated errors are 16.66% and 14.28%. The fact that only SVMfs had an acceptable performance is a further indication of the importance of accounting for interactions between features.

It seems that 15 features could provide a sufficient basis for discriminating between the two populations, since we can get similar performance with the complete datasets, at least for two of the eight datasets. However further experiments should be performed in order to determine the optimal set of features. Here we simply restrict ourselves to feature sets of size 15. It might be the case that fewer are required to discriminate; more experiments are needed to address that issue. An interesting issue in the same direction is the possibility of finding different feature subsets of equally good classification performance, this could provide a basis for further exploration of the features interactions.

Table 6. Results of feature selection with SVMs

dataset	IG	SVMfs	Relief
manual	33.33	26.19	28.57
vd10_h10	30.95	16.66	30.95
vd7_h7	33.33	14.28	38.09
vd6_h6(default)	45.23	21.42	35.71
vd4_h4	45.23	21.42	42.85
vd3_h3	38.09	26.19	33.33
vd2_h2	40.47	35.71	35.71
vd1_h1	16.66	21.42	33.33
Average	35.41	20.76	33.03

6 Identification of Potential Biomarkers

One of the main goals of this study, probably the most important, is to suggest a small set of features that could provide the basis for a potential set of biomarkers. For this we have to analyze the models produced by our learning algorithms in order to determine which features were most important. The task would have been relatively straightforward if the best performing algorithms had been those that produce readable models such as J48 decision trees. This was not the case. Since SVMs turned out to be the most effective algorithm both as a base learner and a feature selector, we will choose its models for further analysis.

6.1 Model Stability Control

Before proceeding to the actual analysis of the models we will undertake a small study of the stability of the models produced with respect to perturbations of the training set. Obviously models that change radically with different training sets would not be of much use.

In order to examine stability we relied on the different models constructed by cross validation. Since we used ten-fold cross validation as the evaluation strategy in essence we used ten different training sets, one for each fold of the cross-validation. Any two training folds have a difference of around 22% when one is using ten fold cross-validation.

To quantify the stability of the produced models we adopted the following strategy: for each fold we produced a ranking of the features based on the importance assigned to them by the coefficients of the linear discriminator produced by the SVM. To compare the rankings among two different folds a, b, we used

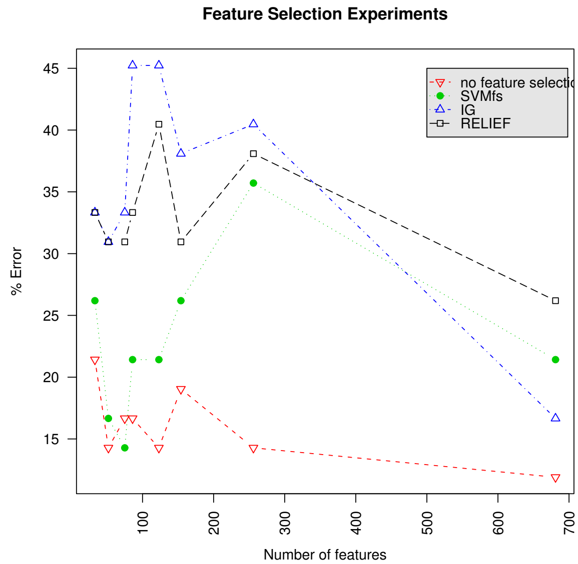

Fig. 4. Error of SVMs on the different data sets when coupled with different feature selection algorithms.

Spearman's rank correlation coefficient, [10]:

$$r_{ab} = 1 - 6 \sum_{f} \frac{(f_{r_a} - f_{r_b})^2}{N(N^2-1)},$$

where the sum is taken overall the features f and f_{r_a}, f_{r_b}, are the ranks of the f feature in the two different folds and N is the number of features. At the end we average the pairwise rank correlation coefficients overall the fold pairs. The results are given in table 7. As one can see there the average rank correlation coefficients are quite high which means that the relative order of the features among the different training folds is preserved to a great extend. We have to note that if we restrict attention only to the top ranked features the averages are even higher. This happens because there are a lot of differences in the way that the less important features are ranked from fold to fold[3], whereas in the top the changes are small. The rank correlation coefficients show that the produced models are quite stable.

We can take a closer look at the model stability by examining how the list of the top 15 features is determined for the $vd7\_h7$ dataset. This dataset produced

[3] Less important features get very small coefficients by the linear discriminator, in these cases a small change in the coefficient can change a lot its ranking at the last positions.

some of the best results both at the base experiments but also in feature selection with SVM with estimated errors being as low as 14.28%. In table 8 we give for the top 15 features, ie, mass/charge ratios, the rank that they get for each of the cross validation folds. As we can see especially for the top ranked features their rank is quite stable among the different folds. The order in which the features appear in the table is determined by their average rank among the folds of the cross-validation, so it reflects their importance.

Table 7. Averaged rank correlation coefficient of feature rankings.

manual	vd10_h10	vd7_h7	vd6_h6	vd4_h4	vd3_h3	vd2_h2	vd1_h1
0.8123	0.8226	0.7752	0.7825	0.7827	0.7388	0.7050	0.7459

Table 8. Top 15 feature for $vd7\_h7$ based on their average rank among the ten folds of the cross-validation for the different datasets.

mass/charge	10	1	2	3	4	5	6	7	8	9	avg	var
15142.21	1	2	3	2	7	1	4	1	3	1	2.5	1.9
6650.63	2	3	1	3	5	2	2	2	4	2	2.6	1.1
66454.06	6	1	4	5	1	3	3	7	5	4	3.9	1.9
4480.20	4	8	7	11	2	4	1	3	2	6	4.8	3.1
9114.72	8	5	8	4	8	6	6	5	6	8	6.4	1.5
7578.39	3	6	6	7	13	5	13	4	7	5	6.9	3.4
28130.95	5	21	2	1	9	7	5	19	8	3	8.0	6.8
66704.85	16	4	13	12	3	10	9	15	10	9	10.1	4.2
16001.45	7	9	20	9	18	8	12	6	9	7	10.5	4.7
33357.24	9	7	9	13	6	12	8	21	13	17	11.5	4.7
22290.18	14	25	12	8	11	19	11	14	14	18	14.6	4.9
9394.80	20	15	17	15	14	17	15	8	12	20	15.3	3.5
8611.98	11	16	5	10	17	15	14	18	11	41	15.8	9.6
8010.05	12	17	19	16	19	16	17	12	18	13	15.9	2.6
4077.23	13	42	11	6	38	9	7	10	15	15	16.6	12.7

6.2 Model Stability across Datasets

We will now examine whether the models produced by the SVM change over the different datasets that we used. The procedure is somehow similar to the one followed in the previous section. For each dataset we identified the 15 most discriminating features based on the results of the ten fold cross-validation, so we got seven more feature tables similar to table 8. From these tables we created a pool of 15×8 features and after accounting for the m_{err} of the mass/charge ratios we ended up with 44 different features [4]. The meaning of these final features is that each one of them was ranked among the top 15 features in at least one of the eight datasets.

We further characterized the quality of a given feature for a given dataset to a finer grain level by the percent of the folds in which it appears among the ten folds of the cross validation. So finally we had for each dataset a vector of 44 dimensions where each dimension was giving the frequency of selection of the corresponding mass/charge ratio in the folds of that dataset. Ordering these features by their quality over all the datasets gave figure 5. The darker the color of a cell is the higher the selection frequency of the corresponding feature is for the corresponding dataset. The quality of a feature is determined by an eight dimensional vector (each dimension is the frequency of selection of the feature in the top 15 features among the folds of a given dataset) and it is simply the average of the vector values. The features that appear on the top of the graph are the ones selected most often among the different datasets, the higher a ratio appears in the figure the most important it is considered by the SVM over all the datasets.

There are three different groups among the eight datasets on the basis of the features that they select. In the first we find all the datasets with a low number of features, ie, $group_a$={*vd3_h3, vd4_h4, vd6_h6, vd7_h7, vd10_h10*}, the second consists of the datasets with many features, $group_b$={*vd2_h2, vd1_h1*}. The *manual* version is closer to the first pattern but it has still some differences. Namely the differences in the mass/charge ratios with values 66454 and 66704 which are completely absent because in the manual version they were removed since they correspond to albumin. What is interesting is the completely different set of features found in the datasets of $group_a$ and $group_b$. The datasets of the second, especially $vd1\_d1$, contain a lot of peaks and many of them can be part of the noise.

The noise is an intrinsic characteristic of the samples and not of the sampling procedure since this was exactly the same for all samples used in this study. Considering the good predictive performance on the datasets of $group_b$ the question that arises is whether the noise, especially since it is intrinsic to the samples, can provide some discriminative information. A possible explanation of the good performance on the datasets of $group_b$ could be in the way the spectra were initially normalized. Normalization was done using total ion current, if the peaks chosen as the normalization basis were discriminative peaks then by scaling all spectra according to them would also scale the noise making it thus discriminative. For the moment though this remains an open issue.

6.3 Identifying potential biomarkers

We will now summarize the work done so far in view of suggesting a small set of potential biomarkers. Applying SVMs and MLP on the different complete

[4] Accounting for the m_{err} also resulted in the merging of two masses of the *vd1_h1* dataset, namely 3326.102 and 3335.321, so this is why for that dataset there will be only 14 top features.

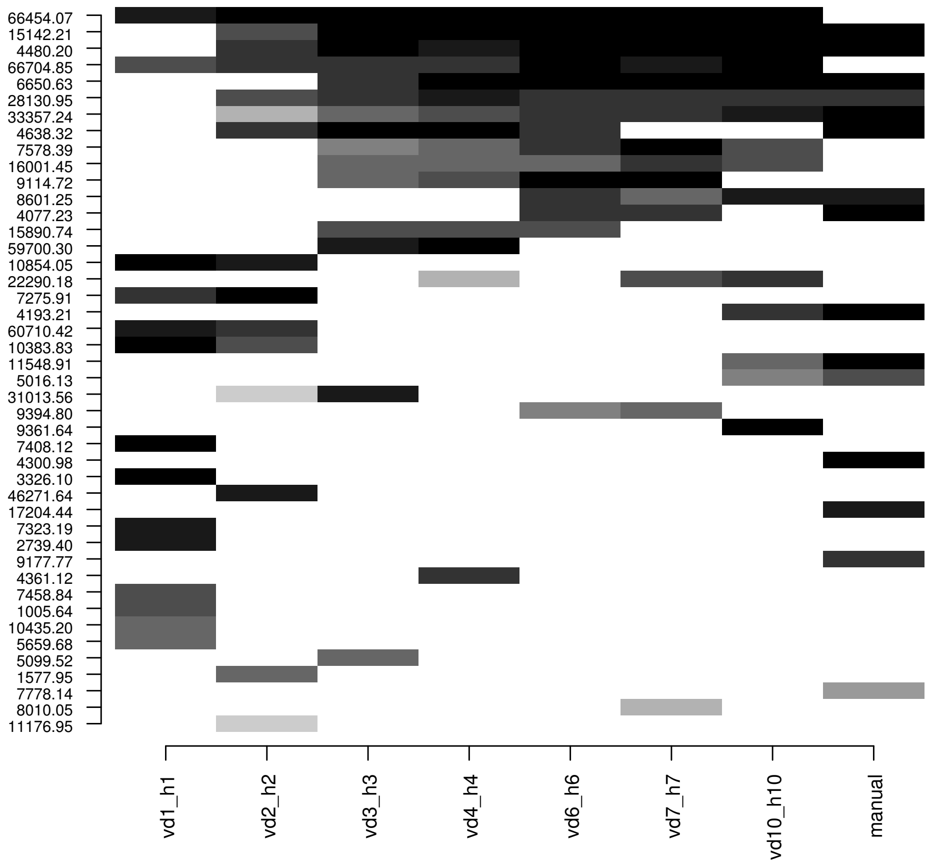

Fig. 5. Frequency of appearance of the top mass/charge ratios among the folds of cross-validation. Black denotes 100% frequency of appearance and white 0%.

datasets, ie, no feature selection, gave us good results (table 3). When we performed feature selection with SVMs and feature sets of size 15 we got similarly good results for a couple of datasets, namely *vd10_h10, vd7_h7* (table 6), so a set of 15 features could provide a good basis for discrimination. Examining the stability of the models produced by SVMs we have shown that this is high (tables 7,8) in other words the set of the top 15 features is quite stable. Based on this observation we retrieved the sets of the top 15 features for each dataset (always with SVMs). We distinguished three groups of datasets from which we think that the most interesting is $group_a$. We did not continue with $group_b$ because we are not sure on how to explain the good performance on these datasets. The *manual* dataset was left outside since it did not provide very good results.

Focusing on the features chosen in $group_a$ we see that there are a lot of commonalities in the top selected features among the different datasets (neighboring dark cells at the top of the graph in figure 5). We consider these to be the most interesting potential biomarkers. If we had to suggest a precise set of masses we would say that the ones which in the graph of figure 5 appear higher than the 59700, including 59700, are the most interesting ones. From these ones we can exclude 66454 and 66704 since they correspond to albumin resulting in the features given in table 9.

Just to provide an indication of the predictive performance with the masses of table 9 we can say that the error of a very simple algorithm like IBk evaluated with ten fold cross-validation on that subset of 13 features was 90.5%, 88.1%, 85.7% (respectively for k=1,3,5). A high performance that shows that all the features are highly relevant and should be considered in a parallel manner in order to perform the classification. The sensitivity and specificity results are given in table 10, specificity is stable at 85.7% while sensitivity takes the following values: 95%, 90% and 85.7%, for k=1,3,5.

Table 9. Potential biomarkers, from left to right and top to bottom line, in increasing order of importance.

59700.03	15890.74	4077.23	8601.25
9114.72	16001.45	7578.39	4638.32
33357.24	28130.95	6650.63	4480.20
15142.21			

7 Related Work

The analysis of mass spectrometry data using machine learning methods has attracted a lot of attention recently. It posses a number of significant challenges namely the high dimensionality of the input space and the data preparation and preprocessing issues. Just to shortly review the relevant literature we should

Table 10. Specificity and Sensitivity results of IBk on the list of potential biomarkers.

		IB1		IB3		IB5	
		Ctrl	Strk	Ctrl	Strk	Ctrl	Strk
True	Ctrl	18	3	18	3	18	3
	Strk	1	20	2	19	3	28
		Predicted					

mention the special issue on data mining methods for mass spectrometry in the Proteomics journal [11], devoted to the presentation of the results of a workshop whose goal was the analysis of mass-spectrometry data for lung cancer diagnosis and biomarker discovery using machine learning and data mining methods. The papers presented in that issue explore a number of different machine learning and data mining methods including decision trees, genetic algorithms, logistic regression, and neural networks.

Other relevant work includes [12] where the authors used decision trees and more precisely CART, [13], to distinguish between prostate cancer, benign prostate hyperplasia and healthy samples based on the mass-spectra of serum samples. [14] where they tried to discriminate between breast cancer and healthy samples on the basis of serum mass-spectra. In this work they used a special form of linear discriminant functions based on statistical learning called Unified Maximum Separability Analysis which was first applied in microarray analysis in [15]. [16] performed a study on prostate cancer. One of the interesting parts of that study was that they have chosen to represent the spectra using the coefficients of the wavelet decomposition of the initial spectra and apply on these coefficients a linear discriminant function. The problem though working with the wavelet coefficients is that the final model is not easily interpretable since it is given in a different space than m/z ratios. The same team applied boosted decision trees, [17], in [18] on the same prostate cancer problem. A very interesting work is that presented in [19] where the problem is again prostate cancer diagnosis and biomarker discovery. In this paper the authors follow an exhaustive procedure of data preparation and preprocessing that includes noise reduction, baseline elimination and peak identification not necessarily in independent stages and use boosting to perform the final classification.

8 Future Work

Although the results are quite good, there are still many things that could be improved. We see most of the work mainly on the preprocessing stage. More work should be done in the peak identification part and the handling of the noise. We should further examine whether there is some information in the noise patterns possibly by experimenting with the complete spectra and not only with their identified peaks. Normalization is also a crucial factor and different methods of spectra normalization should be explored.

Other possible directions include a more systematic experimentation with SVMs in order to further fine tune their parameters, here we limited ourselves only to a small set of values of a single parameter.

The search for a good subset of features was limited to sets of fixed length, this is an issue that should be further explored. Are there other, possibly smaller feature sets, with equally good discriminating power? Can we get different subsets with similar good performance? And if yes what can we conclude about the cross-set interactions? Some work has already been done in identifying feature interactions with promising results. Some of these can be used either to provide new insights about the protein interactions or as part of the preprocessing to reduce the initial set of features.

References

1. Scot Weinberger, Enrique Dalmasso, and Eric Fung. Current achievements using proteinchip array technology. *Current Opinion in Chemical Biology*, 6:86–91, 2001.
2. Eric Fung, Vanitha Thulasiraman, Scot Weinberger, and Enrique Dalmasso. Protein biochips for differential profiling. *Current Opinion in Chemical Biology*, 6:86–91, 2001.
3. L. Allard, P. Lescuyer, J. Burgess, K. Leung, W. Ward, M. Walter, P. Burkhard, G. Corthals, and J-C. Hochstrasser, D. Sanchez. ApoCI and CIII as potential plasmatic markers to decipher ischemic and hemorrhagic stroke. *Proteomics*, 2004. in press.
4. Richard Duda, Peter Hart, and D. Stork. *Pattern Classification and Scene Analysis*. John Willey and Sons, 2001.
5. Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.
6. J.R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufman Publishers, 1992.
7. Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2002.
8. B.D. Ripley. *Pattern Recognition and neural networks*. Cambridge University Press, 1996.
9. Marko Robnik-Sikonja and Igor Kononenko. Comprehensible interpretation of relief's estimates. In Carla Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 433–440. Morgan Kaufmann, 2001.
10. R.V. Hogg and A.T. Craig. *Introduction to Mathematical Statistics*, pages 338–400. Macmillan, New York, 1995.
11. Michael Campa, Michael Fitzgerald, and Edward Patz. Editorial, proteomics 9/2003. *Proteomics*, 2003.
12. Bao-Ling Adam, Yinsheng Qu, John W. Davis, Michael D. Ward, Mary Ann Clements, Lisa H. Cazares, O. John Semmes, Paul F. Schellhammer, Yutaka Yasui, Ziding Feng, and George L. Wright. Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer Research*, 62:3609–3614, 2002.
13. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth Inc, 1984.

14. Jinong Li, Zhen Zhang, Jason Rosenzweig, Young Y. Wang, and Chan Daniel W. Proteomics and bioinformatics approaches for identification of serum biomarkers to detect breast cancer. *Clinical Chemistry*, 48(8):1296–1304, 2003.
15. Z. Zhang, G. Page, and H. Zhang. Applying classification separability analysis to microarray data. In *Proceedings of the First International Conference for the Critical Assessment of Microarray Data Analysis*, 2000.
16. Yinsheng Qu, Bao-ling Adam, Mark Thornquist, John D. Potter, Mary Lou Thompson, Yutaka Yasui, John Davis, Paul F. Schellhammer, Lisa Cazares, MaryAnn Clements, George L. Wright, and Ziding Feng. Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensional data. *Biometrics*, 59:143–151, 2003.
17. R. Freund, Y. Schapire. A decision theoritical generalization of on-line learning and an application to boosting. *Journal of Computer Systems Science*, 55:119–139, 1997.
18. Yinsheng Qu, Bao-ling Adam, Yutaka Yasui, Micheal Ward, Lisa Cazares, Paul Schellhammer, Ziding Feng, John Semmes, and Wright George. Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. *Clinical Chemistry*, 48(10):1835–1843, 2003.
19. Yutaka Yasui, Margaret Pepe, Mary Lou Thompson, Adam Bao-ling, George Wright, Yinsheng Qu, John Potter, Marcy Winget, Mark Thornquist, and Ziding Feng. A data-analytic strategy for protein biomarker discovery: profiling of high-dimensional proteomic data for cancer detection. *Biostatistics*, 4(3):449–453, 2003.