USING STATISTICAL LEARNING METHODS FOR BETTER SPECTRUM CLASSIFICATION

By

Yaoyi Chen

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

In partial fulfillment of the requirements

For the degree of

MASTER OF SCIENCE

in

Biostatistics

August, 2014

Nashville, Tennessee

Approved:

Ming Li, Ph.D.

Christopher J. Fonnesbeck, Ph.D.

BIOSTATISTICS

Using Statistical Learning Methods for Better Spectrum Classification

Yaoyi Chen

Thesis under the direction of Professor Ming Li

Shotgun proteomics has become a widely used technology for identifying a large number of peptides and proteins in complex biological samples. However, any single score function from most search algorithms to evaluate the quality of peptide-spectrum matches (PSMs) is not adequate to discriminate between correct and incorrect spectrum identification. Here, we used and compared multiple logistic regression models with different flexibilities and support vector machines with various kernel functions and random forests to incorporate multiple scores from search engines. New features, such as retention time differences and a number of other modifications, were also incorporated to build a better binary classifier. We validated these methods through bootstrapping and compared their performance to each other. My study has shown that these methods, with their unique strengths, have improved performance - specifically with higher area under ROC curve and better discrimination indices - to classify correct from incorrect peptide spectrum matches.

Approved _____ Ming Li, Ph.D.

ACKNOWLEDGMENTS

I would like to specially thank my advisor, Dr. Ming Li, for her patience, invaluable support, supervision, and helpful suggestions through my research. It is not often that one finds an advisor and friend that always finds the time for listening to the little problems during performing research. Her technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on life and the workings of academic research in general.

I am also grateful to my other dissertation committee members, Dr. Christopher Fonnesbeck who was very supportive and provided valuable advice on my project. He was very patient to revise my thesis over and over and corrected almost every grammar mistakes I had in there.

I would like to thank Dr. Jeffrey Blume for his valuable advice to improve my thesis and support to finish my master's degree in Biostatistics. I would also like to thank Dr. Frank Harrell for his class regression modeling strategies and his suggestions for predictive modeling which improves my thesis a lot. My thanks also goes to Dean Roger Chalkley for his invaluable support through challenging circumstances to encourage me to continue my graduate studies at Vanderbilt University.

I am grateful to the Department of Biostatistics for their help and cooperation over the last two years. I would like to especially thank our program coordinator Linda Wilson for her support during my transferring from Biomedical Informatics Department and during my graduate study

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

TABLE OF CONTENTS

Page

Chapter

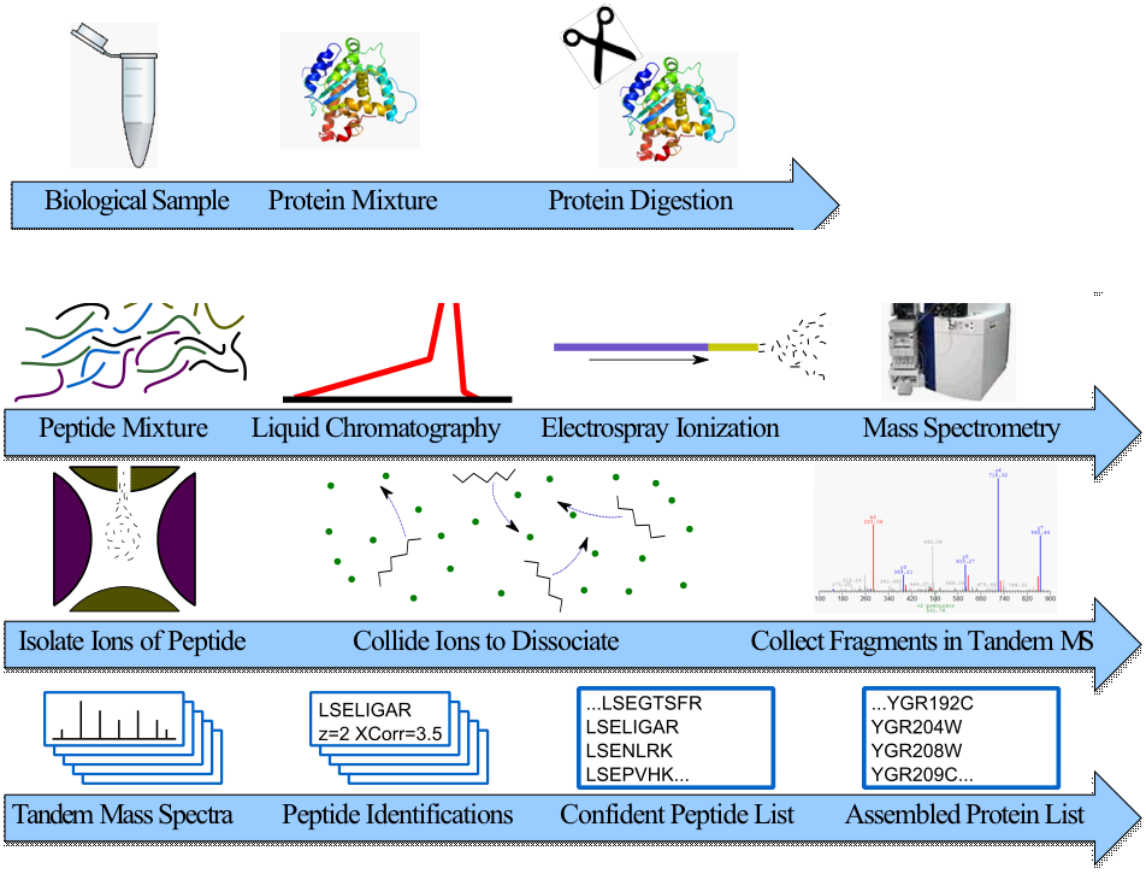
I. INTRODUCTION

1.1 Shotgun Proteomics

Proteomics is a discipline for identifying and quantifying the complete set of proteins in a sample. Mass spectrometry (MS)-based approaches are increasingly used to address diverse questions in proteomics research, enabling one to comprehensively analyze all proteins in complex samples. The application of MS-based proteomics approaches has proved to be successful in molecular and cellular biology research including protein-protein interaction and post-translational modification (PTM) identification. Proteomics has advanced greatly over the past few years with improvements in instrumentation and methodology, enabling many powerful applications such as global analysis of PTM [1-3], large-scale reconstruction of protein interaction networks, functional analysis of complex organisms [4-7], and introduction of proteomics in clinical and translational research [8].

Shotgun proteomics has become the most widely used tool for global characterization of proteins within complex mixtures (Figure 1). The first step is to reduce the complexity of a biological sample by one of several separation techniques such as one- or two-dimensional gel electrophoresis. Large proteins are then digested to peptides using site-specific proteases. Next, peptide mixtures are separated by liquid chromatography and ionized in a mass spectrometer. Precursor ions with particular mass-to-charge (m/z) values are selected and collided with nonreactive gas to generate fragment ions. The corresponding m/z values and peak intensities of fragment ions are recorded in tandem mass spectra, which are interpreted as peptides by computational tools. Finally, the identified peptides are assembled into a list of proteins that are most likely present in the sample.

Figure 1. Shotgun proteomics workflow.

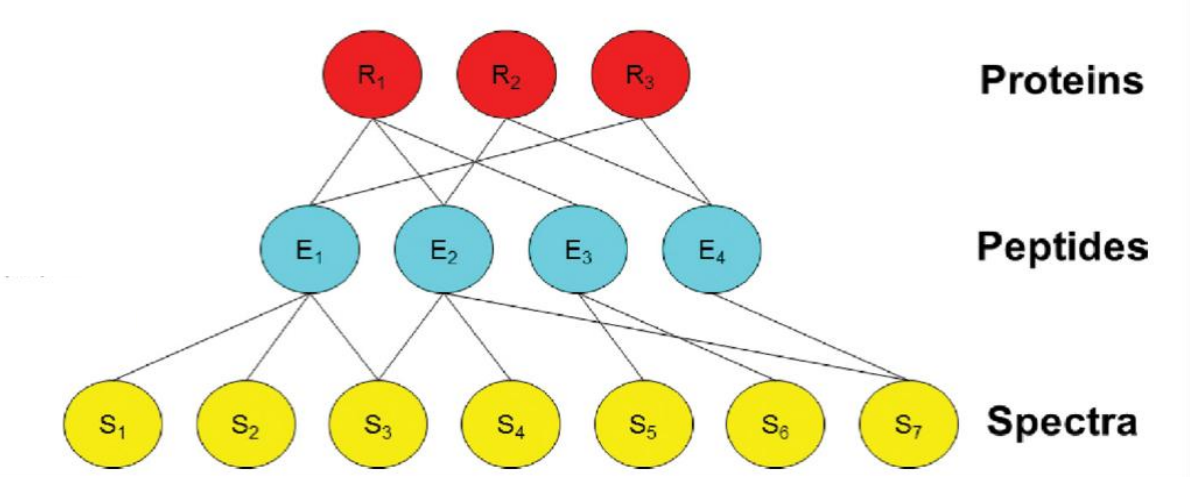


A typical shotgun proteomics experiment generates tens of thousands of tandem mass spectra. Database search techniques such as SEQUEST [9], Myrimatch [10] and MASCOT [11] are usually used for matching MS/MS spectra to a protein database, resulting in thousands of peptide identifications. Evaluating the quality of the match between an observed spectrum and a candidate peptide with the scoring function is critical to any database search technique. The scoring functions such as the MVH score in Myrimatch and Xcorr score in SEQUEST reflect the similarity between the observed and theoretical spectra. The scoring function ranks candidate peptides relative to a single spectrum, producing a best peptide-spectrum match (PSM) for each spectrum. The scores of different PSMs are compared to each other so that correct PSMs with high scores are distinguished from incorrect PSMs.

Although database search algorithms work well, the current scoring methods cannot distinguish correct and incorrect peptide identifications effectively [12-16]. The purpose of my thesis study is to reduce false positive identifications by developing a method to integrate search scores and features to filter out incorrect peptide identifications.

The protein identification problem can be represented as a tripartite graph with layers corresponding to spectra, peptides and proteins [17-19] (Figure 2). An edge between a spectrum and a peptide indicates that the spectrum is assigned to a peptide with a high score. It is also possible that more than one peptide are matched to the same spectra. One peptide can have multiple spectra and abundant peptides can have hundreds of spectra in a LC-MS experiment. An edge from a peptide to a protein indicates the peptide occurs in the protein. This relationship is many to many because one protein may contain multiple peptides while one peptide may be shared among multiple proteins.

Figure 2. Tripartite graph of relationship between spectra, peptides and proteins [20].



1.2 Previous studies using statistical learning methods

Database search engines generally report multiple score metrics to assess the quality of a PSM. Automating accurate spectral identification is an ongoing effort in the proteomics community. The easiest way for automating the analysis is to define specific score cut-offs (e.g. accepting SEQUEST scores with *xcorr*> 2 and *deltaCn* value of at least 0.1. However, a previous study has shown that combining multiple score criteria rather than any single score is possible to reach higher discriminations [21].

There are several approaches for re-ranking the PSMs and setting a threshold automatically in the re-ranked list. IDPicker, a protein assembly tool, uses either a user defined (static) or Monte Carlo simulation method (dynamic) to combine score metrics [22, 23]. In the dynamic method, IDPicker tests linear combination of scores with randomized weights to determine which maximizes the total number of confident identifications. However, the number of weight combinations that can be assigned to the scores is limited. Moreover, this method does not take into account the features of PSMs such as precursor mass error, number of peaks, peptide N/C terminus enzymatic specificity, peptide length, charge segregation, missed cleavages, and so forth. These features are important confounding factors that could improve the ability to discriminate correct and incorrect PSMs.

PeptideProphet [21] uses four statistics computed by SEQUEST search as input to a linear discriminant analysis classifier. It implements a probabilistic approach to assess the validity of peptide assignments generated by database search algorithms. Their approach contains elements of both semi-supervised and supervised learning, achieving higher accuracy than the score threshold method. This system is trained from labeled correct and incorrect PSMs derived from a purified sample of known proteins and retrained in each dataset to which it applied. Problematically, the PeptideProphet algorithm is

challenging to improve with additional information, making this algorithm insufficiently flexible to adapt to the fast development of mass spectrometry.

Percolator built a support vector machine model with a linear kernel to classify PSMs [24, 25]. This method has the benefit of freely exploiting a variety of specific features of the data without overfitting to a particular type of spectrum. However, this method uses a linear kernel which is not capable of modeling non-linear separations and it adds additional parameters that must be tuned, usually numerically. This method freely exploits a variety of specific features of the data without overfitting to a particular type of spectrum; however, the optimization depends on the number of iterations during the course of training [25].

Anderson et.al [26] showed that support vector machines could perform well on ion trap spectra searched with SEQUEST database search algorithm. Ulintz et.al [27] demonstrated that tree-based ensemble methods such as boosting and random forests are suitable for peptide classification problem and provide improved classification accuracy. However, no study uses logistic regression, a popular binary classifier in statistics, to improve peptide classification. Moreover, there is no study comparing the performances of the three statistical learning methods: support vector machines, random forests and logistic regression within the same dataset and using the same validating and evaluating metrics.

I used three methods - logistic regression, SVM and random forests - to distinguish correct peptide-spectrum matches from incorrect ones in a real biological dataset with a gold standard. I also developed a valid comparison scheme to learn the properties of these three methods and suggested the best method for peptide classification problems in proteomics. For logistic regression, I used non-linear transformation of predictor and interactions between predictors to increase the flexibility. By comparing models of different complexity, I am able to explore the mechanism underlying the relationship between experimental measures and the correctness of spectra. For support vector machines, I used linear and Gaussian kernels. The former has been used in Percolator but they only used

3 fold cross validation to select the parameters, which might not be well tuned to the datasets. The latter has not been used in previous studies, but has proved very powerful and is maybe the most widely used kernel to transform the input domain into a nonlinear feature domain. Random forests were also used with the number of features per split tuned by 5-fold cross-validation. The results were validated by bootstrapping.

1.3 Logistic Regression

1.3.1 Logistic Function

Logistic regression is a type of probabilistic statistical classification model used for predicting the outcome of a categorical dependent variable (i.e., a class label) based on one or more predictor variables. The binary logistic regression model was first developed by Cox [28] and Walker and Duncan [29]. In logistic regression, we use the logistic function

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

The logistic function always produces an S-shaped curve. After a bit of manipulation of the formula, we have

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$$

Where $p(X)/[1-p(X)]$ is called the odds and can take any value between 0 and ∞

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

In a logistic regression, increasing X by one unit changes the log odds by β_1 or equivalently, it multiplies the odds by e^{β_1}.

1.3.2 Maximum Likelihood Estimation

Maximum likelihood can be used to estimate the coefficients. The likelihood function is

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

Values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize the likelihood function.

More generally, denoting the response and probability of response of the i th subject by Y_i and P_i, respectively, the model states that

$$P_i = \text{Prob}\{Y_i = 1|X_i\} = [1 + \exp(-X_i\beta)]^{-1}.$$

The likelihood of an observed response Y_i given predictors X_i and the unknown parameters β is

$$P_i^{Y_i}[1 - P_i]^{1-Y_i}.$$

The joint likelihood of all responses Y_1, Y_2.... Y_n is the sum of the log-likelihood for $i=1 \ldots n$:

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{N} \Big\{ y_i \log p(x_i;\beta) + (1 - y_i) \log(1 - p(x_i;\beta)) \Big\} \\ &= \sum_{i=1}^{N} \Big\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \Big\}. \end{aligned}$$

To maximize the log likelihood, the derivative is set to zero. The score equation $U(B)$ is

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta)) = 0$$

The maximum likelihood estimator of β usually cannot be calculated explicitly. The Newton-Raphson method, based on approximating $U(B)$ by a linear function of B in a small region, is usually used to

solve maximum likelihood estimator. The value of β- b^0 is initialized arbitrarily and the linear approximation

$$U(b) = U(b^0) - I(b^0)(b - b^0)$$

is equated to 0 and solve by b yielding

$$b = b^0 + I^{-1}(b^0)U(b^0).$$

At each iteration, the next estimate is obtained by the previous estimate using the formula

$$b^{i+1} = b^i + I^{-1}(b^i)U(b^i).$$

Iteration continues until the -2 log likelihood changes by some pre-specified small amount- Δ over the previous iteration. The reasoning behind this stopping rule is that estimates of B that change the -2 log likelihood by less than Δ do not affect statistical inference since -2 log likelihood is on the chi-squared scale.

The regression parameters can also be written in terms of odds ratios. Logistic regression is mostly used as an inference and data analysis tool to understand the roles of input variables in explaining the outcomes. Compared to other classification models (e.g. SVM, random forests), logistic regression is the most interpretable. One can study the effect and association of every predictor from the models.

The Akaike information criterion (AIC) is a measure of the relative quality of a statistical model for a given set of data. It provides a way for model selection [40]. For any statistical model, the AIC value is

$$AIC = 2k - 2\ln(L)$$

Where k is the number of parameters in the model, L is the maximized value of the likelihood function for the model.

1.3.3 Cubic Spline Function

The cubic spline function is a spline constructed of piecewise third-order polynomials which pass through a set of m control points [30]. A smooth cubic spline function with three knots (a, b, c) is represented as:

$$\begin{aligned} f(X) &= \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 \\ &\quad + \beta_4 (X-a)_+^3 + \beta_5 (X-b)_+^3 + \beta_6 (X-c)_+^3 \\ &= X\beta \end{aligned}$$

Where

$$(u)_+ = \begin{array}{l} u,\ u > 0, \\ 0,\ u \leq 0. \end{array}$$

If the cubic spline has k knots, the function will estimate k+3 coefficients [31]. The smooth spline function has the drawback that they are poorly behaved in the tails. The restricted cubic spline function has k-1 parameters. Restricted cubic spline function with k knots: ($t_1 \ldots t_k$) can be represented as:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{k-1} X_{k-1},$$

where $X_1 = X$ and for $j = 1, \ldots, k-2$,

$$\begin{aligned} X_{j+1} &= (X - t_j)_+^3 - (X - t_{k-1})_+^3 (t_k - t_j)/(t_k - t_{k-1}) \\ &\quad + (X - t_k)_+^3 (t_{k-1} - t_j)/(t_k - t_{k-1}). \end{aligned}$$

The restricted cubic spline function was implemented in the *rcs* function in the *Hmisc* package[32]. The fitting of restricted cubic splines depends on the number of knots. Placing knots at fixed quantiles of a predictor's marginal distribution is a good approach for most datasets. When the sample size is large with a continuous uncensored response variable, k = 5 is a good choice. The default knots are located at .05 .275 .5 .725 .95 for k=5 [31].

1.4 Support Vector Machines

The support vector machine is an approach for classification developed in the 1990's and has grown in popularity since then [33]. A support vector machine constructs a hyperplane or a set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class [34]. The dimension of the transformed space can be very large, even infinite in some cases. This seemingly prohibitive computation is achieved through a positive definite reproducing kernel, which gives the inner product in the transformed space.

1.4.1 Maximal Margin Hyperplane

In a p-dimensional space, a hyperplane is a subspace of dimension p – 1. In p-dimensional setting, a hyperplane is defined as:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

If a point $X = (X_1, X_2, \ldots, X_p)^T$ satisfies the formula, then X lies on the hyperplane.

If $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0.$, then X lies to one side of the hyperplane,

If $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$, then X lies to the other side of the hyperplane.

For binary responses, the two classes are usually labeled either by $y_i=1$ and $y_i=0$ or by $y_i=1$ and $y_i=-1$ respectively. These two labels are equivalent in separating two categories of observations. $y_i=1$ and $y_i=0$ was used in my study. To demonstrate the theory, we label the observations as $y_i=1$ and $y_i=0$.

A separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1.$$

Equivalently

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$$

A classifier that is based on a separating hyperplane leads to a linear decision boundary.

The maximal margin hyperplane is the separating hyperplane for which the margin is largest. In other words, it is the hyperplane that has the farthest minimum distance to the training observations.

Maximal margin is used to classify observations by which side of the maximal margin hyperplane it lies.

Maximizing the margin is good because points near the separating hyperplane represent very uncertain classification decisions. A hyperplane with a large margin makes less low certainty classification decisions: a slight error or variation is less likely to cause a misclassification. A good separation can be achieved by this hyperplane since usually the larger the margin the lower the generalization error of the classifier [54].

To build maximal margin classifier based on n training observations: $x_1, \ldots, x_n \in \mathbb{R}^p$ and associated outcomes $y_1, \ldots, y_n \in \{-1, 1\}$. The maximum hyperplane is the solution to the optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

M represents the width of margin of the hyperplane, C is non-negative tuning parameter, $\epsilon_1, \ldots, \epsilon_n$ are slack variables that allow individual observations on the wrong side of the hyperplane. $\epsilon_i = 0$ indicates the *i*th observation is on the right side. We classify the test observation based on the sign of

$$f(x^*) = \beta_0 + \beta_1 x_1^* + \ldots + \beta_p x_p^*$$

As C increases, we are more tolerant of violations to the margins, so the margins will widen. Therefore C controls the bias-variance trade-off of the support vector classifier. Support vectors are the subset of the training data that lie on the margin.

The optimization problem is solved by a Lagrange multiplier [35].

$$L_P = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - 1]$$

Setting the derivatives to zero, we have

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i,$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i,$$

Substituting these two into L_p, we have the Wolfe dual

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0.$$

1.4.2 Kernels

The solution to the SVM problem involves only the inner product of the observations. The inner product of two observations X_i, $X_{i'}$ is

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}.$$

The linear support vector classifier is

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

Where there are n parameters, one per training observation. α_i is non-zero only for support vectors, therefore, we have $f(x)$:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

We replace the inner product with a generalized function, termed Kernel

$$K(x_i, x_{i'})$$

Kernel is a symmetric, semi-positive definite function to quantify the similarity of two observations. A linear kernel is

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

The support vector classifier with a linear kernel is linear in the features. The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter C. Kernel function can implicitly map the data into a feature space. One can choose among many types of kernels. One choice is the polynomial kernel with degree d

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d.$$

Using $d>1$ leads to a more flexible decision boundary than linear kernel.

In practice, a common choice is the Gaussian kernel (radial kernel):

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

They are flexible and can build a lot of possible relations quickly. γ controls the radial basis of the kernel. It is necessary to choose appropriate tuning parameters for the model. The conventional way is to use k-fold cross validation:

1. Divide the data into k equal subsets, $s=1...k$, start with $s=1$;

2. Pick a value for the tuning parameter;

3. Fit the model using $k-1$ subsets other than subset s; predict for subset s and measure the associated loss;

4. Repeat the iteration for every $s=1...k$[36] .

The best combination of *C* and γ is often selected by grid search from a list of C and γ that uniformly spanned a logarithmic space [37]. Usually, each combination of parameter choices is checked using cross validation, and the parameters with best cross-validation accuracy are picked. The final model, which is used for testing and for classifying new data, is then trained on the whole training set using the selected parameters. Currently, polynomial kernels are less widely used than the Gaussian kernel, which maps

data to an infinite dimensional space. Previous studies have shown for some data, the testing accuracy through polynomial kernel is slightly worse than Gaussian kernel under similar training and testing costs [38].

1.4.3 Advantages and Disadvantages of SVM

By introducing the kernel, SVM gains flexibility in classification. Since the kernel implicitly contains a non-linear transformation, no assumptions about the functional form of the transformation, which makes data linearly separable, is necessary. SVMs provide a good out-of-sample generalization, e.g. if the parameters *C* and γ in the case of a Gaussian kernel are appropriately chosen. This means that, by choosing an appropriate generalization grade, SVMs can be robust, even when the training sample has some bias. SVMs deliver a unique solution, since the optimality problem is convex. This is an advantage compared to neural networks, which have multiple solutions associated with local minima and for this reason may not be robust over different samples. The disadvantages of SVM include the lack of transparency of the results [27] and that it is memory intensive. Non-linear kernel suffers from high time and space complexity associated with the need of operating kernel matrix.

1.5 Random Forests

Random forests is an ensemble learning method for classification (and regression) that operates by constructing a set of decision trees that returns the class that is the mode of the classes output by individual trees[17][34]. The method randomly selects, with replacement, n samples from the original training data. A small group of input variables on which to split are randomly selected. Growing a tree means partitioning the data based on some attribute of them at each node. Each tree is grown to the largest extent possible. To classify a new sample from an input, one runs the input down each of the trees in the forest. Each tree gives a classification (vote) and the forest chooses the classification having the most votes over all the trees in the forest.

1.5.1 Decision Trees

Tree-based methods are useful in that they are easy to interpret. Decision trees can be applied to both regression and classification problems. For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. There are different measures of node impurity: classification error, the Gini index, and cross-entropy. These measures are used as the criterion to make the splits.

The classification error is the fraction of the training observations in that region that do not belong to the most common class [33].

$$E = 1 - \max_k(\hat{p}_{mk})$$

where $\hat{p}_{mk}$ represents the proportion of training observations in the *m* region that are from the *k*th class.

In addition to the classification errors, another measure, the Gini index, is preferred for tree-growing. The Gini index is defined as a measure of total variance across the *k* classes

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}).$$

The Gini index is small if all of the $\hat{p}_{mk}$ are close to zero or one. The Gini index is a measure of node purity - a small value indicates a node contains predominately observations from a single class.

Cross-entropy is defined as:

$$H= \quad -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

All three measures are similar; however, cross-entropy and the Gini index are more differentiable and more amenable to numerical optimization [39]. Therefore, when building a classification tree, the Gini index and cross-entropy are usually used to evaluate the quality of a particular split. The expected

information gain is the change in information entropy H from a prior state to a state that takes additional information. The information gain for a given attribute a is defined as follows

$$IG(T, a) = H(T) - H(T|a)$$

A tree model is able to capture multi-way interactions between the splitting predictors. It also naturally handles categorical and continuous variables, missing values, non-linearity, different scales between variables, etc. [30, 55]. However, tree models have problems in selecting a non-redundant feature set [55]. A decision tree select a feature at each split based on information-theoretic criterion, without considering if the feature is redundant to the features selected in previous nodes. Deng *et al.* proposed a tree regularization framework which penalizes selecting a new feature for splitting when its gain is similar to the feature used in previous splits. This regularization method was applied to random forests (regularized random forests (RRF)) and was shown to be able to select high-quality feature subsets [55]. Pruning is a technique in machine learning that reduces the size of decision trees by removing nodes that provide little power for classification. The goal is to reduce the complexity of the classifier and increase prediction accuracy by removing the section of the classifier that might be based on noise. There are several ways of pruning. The main two categories of pruning are top-down (traverse and trim sub-trees from the root of the tree) or bottom-up methods (traverse and trim sub-trees from the leaves of the tree). There are two main methods: reduce error pruning and cost complexity pruning. Reducing error pruning is to replace each node with its most popular class. If the prediction accuracy does not decrease, this change is kept. Cost complexity pruning generate a series of trees: $T_0 \ldots T_m$

T_i is created by removing a sub-tree from T_{i-1} and replacing it with a leaf node. T_0 is the initial tree and T_m is the root alone. The sub-tree that is removed is chosen to minimize the following measure:

$$\frac{err(prune(T,t),S) - err(T,S)}{|leaves(T)| - |leaves(prune(T,t))|}$$

where $err(T, S)$ is the error rate of tree T over data set S [35]. $|leaves(T)|$ is the number of leaves in tree T. The function $prune(T, t)$ defines the tree returned by pruning the sub-trees t from the tree T. Once the series of trees has been created, the best tree is chosen by generalized accuracy as measured by a training set or cross-validation [37].

1.5.2 Bagging

Trees are easier to interpret than linear regression. For example, trees can easily handle qualitative predictors without the need to create dummy variables. However, trees have lower predictive accuracy relative to other classification approaches such as SVM and logistic regression. They also suffer from high variance. However, by aggregating many decision trees using bagging or random forests, the predictive accuracy can be greatly improved and the variance reduced. For regression trees, a set of predictions $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ can be calculated from B separate training sets. The predictions from all individual regression trees can be averaged by:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x')$$

Bagging trains the method on the b^{th} bootstrap training set in order to get $\hat{f}^{*b}(x)$, on a total number of B bootstraps, we can get the average of all predictions:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

For classification trees, we take a majority vote of B trees: the overall prediction is the most commonly occurring class among B predictions. Each tree makes use of a subset of the observations. The remaining observations not used to fit a given bagged tree are out-of-bag (OOB) observations [34]. The classification error of these OOB observations is called OOB error.

1.5.3 Random Forests

The difference between random forests and bagging is that random forests choose one of a random sample of m candidate predictors from a full set of p predictors. A fresh sampling of m predictors is taken at each split, where m is usually approximately the square root of the total number of predictors. The predictions from the bagged trees will be highly correlated. Averaging these predictions will not reduce variance as much as averaging uncorrelated predictions. Random forests overcome this problem by only considering a subset of predictors in each split. On average, $(p\text{-}m)/p$ of the splits will not examine the strong predictor, making the resulting trees more reliable. No pruning is needed for random forests [30]. Overfitting is less likely to happen in random forests since the samples used to train individual trees are random bootstrap samples, and in each split random features are used.

1.5.4 Importance of Predictors

Bagging a number of trees makes it impossible to interpret the resulting model. Random forests use the OOB samples to construct an importance value to measure the prediction strength for every predictor. When the tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is computed. Then we randomly permute values of the predictor of interest- m in the OOB samples and examine the resultant changes in accuracy; decreased accuracy is recorded as the raw importance score of m in the dataset.

If the values of this score from tree to tree are independent, then the standard error can be computed by a standard computation. The correlations of these scores between trees have been computed for a number of datasets and proved to be very low; therefore, standard error can be computed by dividing the raw score by its standard error to get a z-score, and assigning a significance level to the z-score assuming normality [34]. If the number of variables is large, random forests can be run once with all the variables then run again using the most important variables from the first run [56].

This permutation method of determining variable importance has some drawbacks. For example, for data including categorical variables with different number of levels, random forests are biased in favor of the attributes with more levels. Methods such as partial permutations can be used to solve the problem [57].

Every time a split of a node is made on variable m, the Gini index for the two descendent nodes is less than that of the parent node. Adding up the Gini index increase for each individual variable over all trees in the random forests gives a fast variable importance called the Gini importance. The Gini importance is often very consistent with the permutation importance measure [56].

Another useful tool from random forest is the proximity which can be used to cluster data. The proximities originally formed an N*N matrix (N is the number of cases in the data). After a tree is grown, put all of the data, both training and testing down the tree. If case k_1 and k_2 are in the same terminal node, increase their proximity by one. At the end, normalize the proximities by dividing by the number of trees.

1.5.5 Advantages and Disadvantages of Random Forests

Random forests have several nice features: they give estimates of which variables are important for classification and are robust with respect to input variable noise. When the number of variables is large but the number of relevant variables is small (there are a lot of noise variables), in each split the probability that the relevant variables are selected is small. Random forests may perform poorly with small m (number of variables selected at each split). When the number of relevant variables increases, random forests are robust when the number of noise variables increases. The robustness is largely due to the fact that misclassification cost is relatively insensitive to the variance and bias of the probability estimates in each tree. Random forests also account for interactions among predictors and do not need to tune a lot of parameters, unlike SVM. Generally, random forests are robust to overfitting; however, they may overfit for some datasets with noisy classification or regression tasks. For data that includes

categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for these data[38].

1.6 Model Validation

Model validation determines whether predicted values from the model are likely to accurately predict responses on future subjects or subjects not used to develop our model. There are two major modes of model validation, external and internal. The most stringent form of external validation is to validate in a completely different source of data, e.g. testing a final model developed in one country on subjects in another similar country at another time. In other words, external validation of a prediction tool uses data that were not used to fit the model. The least stringent form of external validation involves using the first m of n observations for model training and using the remaining n - m observations as a test sample. Even though external validation is frequently favored by non-statisticians, it is often difficult to obtain separate validation data. Internal validation involves fitting and validating the model by carefully using one series of subjects [40]. One uses the combined dataset in this way to estimate the likely performance of the final model on new subjects, which is often of most interest. Bootstrap is a general-purpose technique for obtaining estimates of the properties of statistical estimators without making assumptions about the distribution giving rise to the data. The basic idea is to repeatedly simulate a sample of size n from the dataset, computing the statistic of interest, and assessing how the statistic behaves over $\boldsymbol{B}$ repetitions [41].

In this study, I performed classification for shotgun proteomics using logistic regression and multiple statistical learning methods to combine multiple scores and PSM features, and compared the fitted or predicted outcomes to the gold standard . I also demonstrated the advantages and disadvantages of

these algorithms in working with mass spectrometry data and recommended the best methods to use in this field.

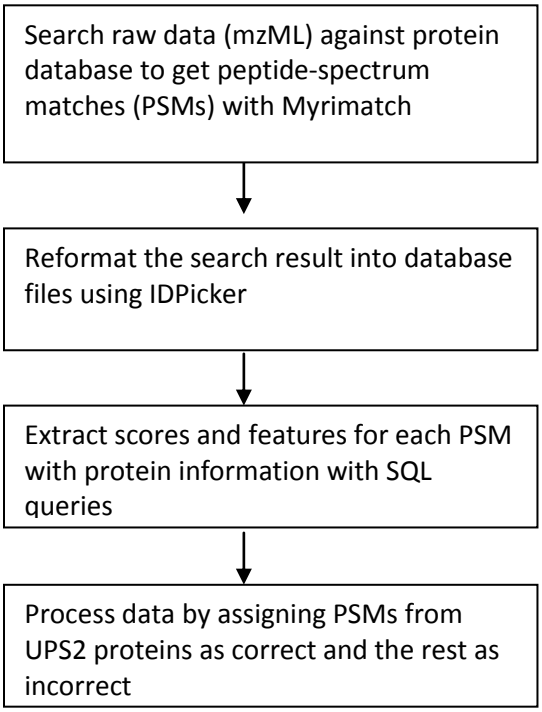
II. MATERIALS AND METHODS

2.1 Mass Spectrometry Data Collection

In most statistics studies, we use simulations to model random events. As some real-world datasets may be difficult, expensive or time-consuming to analyze, using simulation can help researchers gain insights on real world situations. Mass spectrometry datasets in particular are easily obtained but challenging to interpret due to their large size and high noise levels. The mechanism of these noises is not fully understood, making it hard to mimic the datasets with simulations. Simulations are useful only if it closely mimics real-world situations. Therefore, using a real dataset instead of a simulated one is a more feasible way to train and test the classifiers. A shotgun proteomics dataset, "UPS2 standard dataset", created by Ivanov et al [42, 43] was selected as a standard scheme to compare three different statistical learning methods. This dataset has become the "gold standard" of training datasets due to its prior use in several studies for validating new bioinformatics and biostatistics algorithms [42, 43].

The UPS2 dataset contains 48 human proteins with a dynamic range spanning 0.5-50,000 *fmol*. I used a subset consisted of proteins with UPS2 concentration over 100 *fmol*. There were 18 LC MS/MS runs, and each run contained one specified amounts of the UPS2 standard spanning over 2 orders of magnitude. Data were searched against the human protein database.

Figure 3. Mass spectrometry data collection workflow



2.2 Mass Spectra Data Processing

MS/MS scans were converted to mzML by the msconvert tool that is part of the ProteoWizard[44] software package. All protein databases contained both forward and reverse sequences for estimating protein and peptide identification errors. Peptides were identified with MyriMatch (version 1.6.79) [10]. MyriMatch applied a precursor tolerance of 10 *ppm* for Orbitrap data and was configured to use a static mass shift of 57.0215 *Da* for alkylated cysteines and allowed the variable modification of oxidation of methionine (+15.9949 *Da*) and formation of N-terminal pyroglutamate (–17.0265 *Da*) . The search results were converted by IDPicker (version 3.0.515) from pepXML format to IDPDB format [22, 23, 43, 45]. There were 17095 PSMs in total, 11354 (66.42%) of them were correct and 5741 (33.58%) of them were incorrect.

Scores and features for each PSM were collected from the assembled summary database file from IDPicker by SQL queries (Figure 3). Because there were only the UPS2 proteins and contaminant proteins in the set, the spectrum that matched to peptides from the correct proteins were marked as correct, otherwise marked as incorrect. Spectra are stratified into four groups depending on their charges: 1, 2, 3 and >=4. Peptide hydrophobicity was calculated by SSRCalc[46]. Retention time was predicted by hydrophobicity and the differences between observed scan time and predicted retention time were calculated as variable : *Rtdiff*.

2.2.1 Features of PSMs

PSMs were presented using 3 scores and 12 features. Scores described the quality of match between the observed and theoretical spectra, and features described properties of the peptide or the spectrum (Table 1). Some of these scores and features were also used by Percolator: *xcorr*, *massError*, *absMassError*, *monoMass*, *missCleavages*, *enzN*, *enzC*, *pepLen*, *charge* 1-3. Others were new features that were considered important in this algorithm: *matchedPeaks*, *unmatchedPeaks*, *numMods*, *RTdiff*.

2.3 Descriptive Statistics

The distributions of univariates were described by empirical cumulative distribution with the *Ecdf* function and univariate summary plots from *Hmisc* package. The Spearman's rank correlations between pairs of predictor variables were calculated to measure their statistical dependence. Variables were clustered based on Spearman's rank correlation by *varclus* function from *Hmisc* package.

2.4 Models

2.4.1 Logistic Regression Models

Four logistic regression models were fitted to the dataset. To avoid scaling issues in models, all continuous predictor were standardized to mean=0 SD=1 in logistic regression. Only *missCleavages* (categorical, 0-6), *enzN* (binary, 0/1), *enzC*(binary 0/1), *charge.cat*(categorical, 1-4) were used in their original scale for interpretability.

1. Linear main effect model (*modelReg*)

The first model was a logistic regression with only main effects. The outcome was the correctness of the PSMs and the predictors were the 3 scores and 12 features [31]. Since the dataset was large, variable reduction was not necessary. The odds ratio of each predictor and its confidence interval were estimated to understand the effect of each predictor on the outcome.

The main effect model can be represented below:

$$\text{Prob}\{\text{correctness}\} = \frac{1}{1+\exp(-X\beta)}, \text{ where}$$

$$\begin{aligned} X\hat{\beta} = {} & MVH + mzFidelity + xcorr + unmatchedPeaks + massError + Charge\ group\ 2 \\ & + Charge\ group\ 3 + Charge\ Group\ 4 + enzN + enzC + missCleavages \\ & + monoMass + RTdiff + ModNumber \end{aligned}$$

2. Spline model (*modelReg.spline*)

I also fitted a logistic regression model with transformed continuous predictors and untransformed categorical predictors [47, 48]. Continuous variables were transformed with restricted cubic spline function (*rcs*) with 5 knots. This model was compared to the main effect model to show how non-linear transformation of these predictors affects accuracy of prediction.

The spline model can be represented below:

$$\text{Prob}\{\text{correctness}\} = \frac{1}{1+\exp(-X\beta)}, \text{ where}$$

$$\begin{aligned} X\hat{\beta} = {} & rcs(MVH) + rcs(mzFidelity) + rcs(xcorr) + rcs(unmatchedPeaks) \\ & + rcs(massError) + enzN + enzC + missCleavages + rcs(monoMass) \\ & + rcs(RTdiff) + ModNumber + Charge\ group\ 2 + Charge\ group\ 3 \\ & + Charge\ Group\ 4 \end{aligned}$$

where rcs is the restricted spline function with five knots

3. Linear model with interactions (*modelReg.inter*)

I fitted logistic regression with non-transformed predictors and 2-way interaction terms between charge and *MVH*, *massError*, *RTdiff*, *ModNumber*, *enzN* and *enzC*. According to previous studies, charge is an important confounding factor which may interact with other predictors in the model. This interaction model was used to show how interactions between charge and other predictors affected accuracy of prediction.

The linear model with interaction can be represented as:

$$\text{Prob}\{\text{correctness}\} = \frac{1}{1+\exp(-X\beta)}, \text{ where}$$

$$X\hat{\beta} = as.factor(charge.cat) * \{rcs(MVH) + rcs(massError) + rcs(RTdiff) + ModNumber + enzN + enzC\} + missCleavages + rcs(mzFidelity) + rcs(xcorr) + rcs(monoMass) + rcs(unmatchedPeaks)$$

4. Spline model with interactions (*modelReg.spline.inter*)

4.1 Two models were fitted with the data – the first model was based on the previous observations that charge may interact with other variables such as scores and retention time difference. 2-way interaction terms between charge and *MVH*, *massError*, *RTdiff*, *ModNumber*, *enzN* and *enzC* were included in the model. Predictors such as *missCleavages*, *mzFidelity*, *xcorr*, *monoMass*, *unmatchedPeaks* were included as additive predictors since including them in the interaction terms caused singularity in regression. All continuous predictors were transformed with restricted cubic spline function.

This model can be represented as:

$$\text{Prob}\{\text{correctness}\} = \frac{1}{1+\exp(-X\beta)}, \text{ where}$$

$$X\hat{\beta} = as.factor(charge.cat) * \{rcs(MVH) + rcs(massError) + rcs(RTdiff) + ModNumber + enzN + enzC\} + missCleavages + rcs(mzFidelity) + rcs(xcorr) + rcs(monoMass) + rcs(unmatchedPeaks)$$

4.2 The second model included interactions between *misscleavages* (the variable with the highest apparent importance) and other variables, and also included the interactions introduced in 4.1. It was a complex model which allowed a lot of flexibility.

This model can be represented as

$$\text{Prob}\{\text{correctness}\} = \frac{1}{1+\exp(-X\beta)}, \text{ where}$$

$$\begin{aligned} X\hat{\beta} = {} & missCleavages * \{rcs(MVH) + rcs(mzFidelity) + rcs(xcorr) \\ & + rcs(unmatchedPeaks) + rcs(massError) + enzN + enzC \\ & + rcs(monoMass) + rcs(RTdiff) + ModNumber\} \\ & + as.factor(charge.cat) * \{rcs(MVH) + rcs(massError) + rcs(RTdiff) \\ & + ModNumber + enzN + enzC\} \end{aligned}$$

AIC was used to evaluate the quality of these models. The corresponding AIC value of each model was obtained and compared to each other.

I did not fit more complex models due to collinearity in the predictors. For the spline model with interactions, validation with 32 out of 200 bootstrapped samples could not converge, indicating no maximum likelihood estimator exists. Therefore, I used these two models to show how the combination of non-linear transformation and variable interactions affected the prediction accuracy.

I did not penalize the models since there were only less than 20 predictors but over 5000 observations in each category of dataset. The sample size was sufficient for logistic regression analyses. To show this, I tried to use the function *pentrace*[49] from the *rms* package to choose the best penalty factor for the models. The best penalty factor is 0, indicating no penalty was needed for these analyses.

2.4.2 Support Vector Machine

1. SVM with a linear kernel

All predictors were standardized before analysis. Five cost parameters that uniformly spanned a logarithmic space were used: 0.01, 0.1, 1, 10, and 100. The best cost parameter was selected based on 5-fold cross validation.

2. SVM with a Gaussian kernel

I trained SVM models with a Gaussian kernel. I selected from five cost parameters (0.01, 0.1, 1, 10, 100) and five gamma parameters (0.005, 0.05, 0.1, 0.5, 1) using a grid search. The best parameter combination was selected based on 5-fold cross validation. The best parameters were 10 for *C* and 1 for γ. Gaussian kernels non-linearly map the data space into a higher dimensional space. Its use with appropriate regularization guarantees a globally optimal predictor which minimizes both the estimation and approximation errors of a classifier. In this thesis, I compared different SVM with different kernel allowing flexibility to classify spectra. I did not use polynomial kernel because previous studies have shown that polynomial kernel may not give higher accuracy than Gaussian kernel under similar training and testing costs [38].

2.4.3 Random Forests

I used *tuneRF* function in *RandomForest* package in R to tune parameter *m* (the size of the random subsets of features to consider when splitting a node). I started with the default number of *m* and search for the optimal value (with respect to out-of-bag error estimate) of *m* for *randomForest*. The validated best *m* was 4 which was approximately the same as the default square root of number of variables (15).

2.4.4 Model Validation

All models were validated using the same bootstrapping strategy. The logistic regression models were validated using the *validate* function in the *rms* package [49]. This function validates models with statistical indices to quantify discrimination ability (e.g., R^2, model χ2, Somers' D_{xy}, Spearman's ρ, area under ROC curve). Two important indices are D_{xy} and calibration slope. D_{xy} is Somers' rank correlation between predicted probability that $Y = 1$ vs. the binary Y values. This equals $2(C-0.5)$ where C is the

ROC Area or concordance probability. Calibration slope is slope of predicted log odds vs. true log odds [40].

For SVM and random forests, I drew a training set of N (N=17095) observations from the original data of size N with replacements. Then, I fitted models with the training data and evaluated the model based on the out-of-bag (oob) samples. These three steps were repeated 1000 times. For each iteration, sensitivity, specificity, precision, accuracy (at probability threshold 0.5), AUC and F-measure were estimated in the training sets and out-of-bag test sets.

The probabilities based on the output of the machine learning models were used as the classifier for the PSMs. PSMs with probability greater than 0.5 were considered positive by the test, otherwise negative. I used this criterion to assess the performance of the classifiers by classification matrix, accuracy, sensitivity, precision, specificity, F-measure etc. I understand this criterion- 0.5 might be arbitrary, and that choosing different cutoffs will lead to different results. Therefore, I mainly used discrimination indices and AUC to assess the models.

Table 1. Three scores and twelve features used to distinguish correct and incorrect PSMs

	Predictor	Description
Score 1	*MVH*	Score evaluating the probability that a random peptide would match to fragments as intense in the observed spectrum as a particular candidate sequence by multivariate hypergeometric (MVH) distribution[10]
Score 2	*mzFidelity*	Score evaluating a PSM occurred by random change by multinomial distribution[50-52]
Score 3	*Xcorr*	Cross correlation between calculated and observed spectra[53]
Feature 1	*matchedPeaks*	number of matched peaks

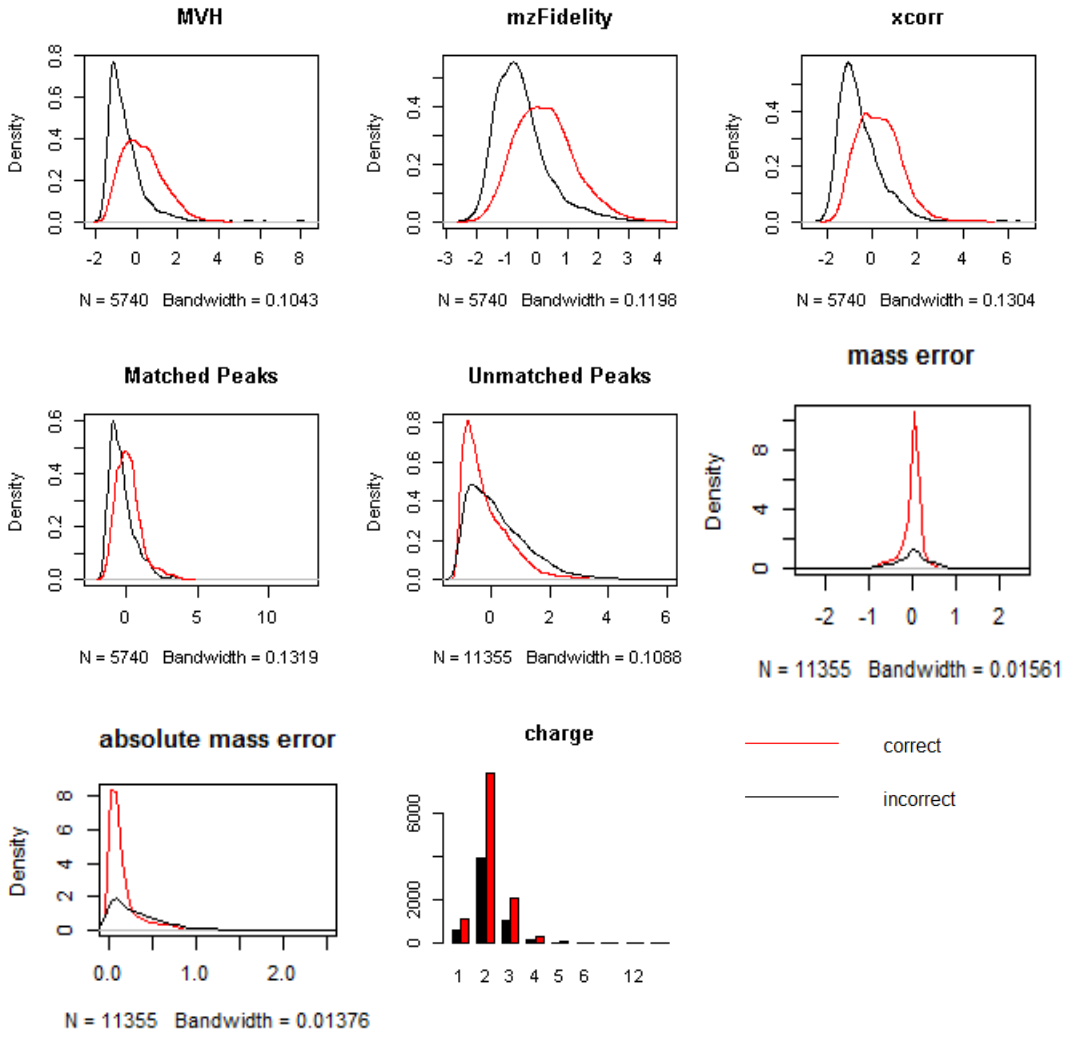
Feature 2	*unmatchedPeaks*	number of unmatched peaks
Feature 3	*massError*	the difference in calculated and observed monoisotopic mass
Feature 4	*AbsMassError*	Absolute value of the difference in calculated and observed monoisotopic mass
Feature 5	*Charge*	Categorical feature 1: charge state 1; 2: charge state 2; 3: charge state 3 4: charge state >=4
Feature 6	*enzN*	Boolean: is the peptide preceded by an enzymatic site
Feature 7	*enzC*	Boolean: does the peptide have an enzymatic C-terminus
Feature 8	*missCleavages*	Number of missed internal enzymatic sites
Feature 9	*pepLen*	The length of the matched peptide, in residues
Feature 10	*monoMass*	The monoisotopic mass of the peptide
Feature 11	*RTdiff*	Difference between predicted and observed retention time (in seconds)
Feature 12	*numMods*	Number of modifications in the peptide

III. RESULTS

3.1 Descriptive Statistics

The distributions of the predictors were segregated by the correctness of the PSMs (Figure 4, supplementary table 1) . The three scores (*MVH*, *Xcorr* and *mzFidelity*) were distributed differently in correct versus incorrect PSMs; Incorrect PSMs had a higher density of scores at the low end. However, none of these three scores was able to separate correct from incorrect PSMs very well because there was significant overlap between the scores of two groups. The distributions of the *MVH* and *Xcorr* scores had longer tails than the *mzFidelity* score. Correct PSMs had a larger number of matched peaks and a smaller number of unmatched peaks than incorrect PSMs. Distribution of both mass error and absolute mass error showed a spike around 0 for correct PSMs and a wide distribution for incorrect ones. There were 9 different charge values: 1, 2, 3, 4, 5, 6, 11, 12, and 13. Since in liquid chromatography mass spectrometry results, charge states more than 4 are very rare (Figure 4). , I collapsed PSMs with charge >=4 into one group and kept charge 1, 2, 3 as individual groups. The enzyme cleavages were more specific at N terminus and C terminus for correct PSMs. The number of missed cleavages was mostly 0 for correct PSMs, while they were distributed more widely for incorrect ones. Correct PSMs had slightly higher densities of peptide length around 15. A similar distribution was found at the monoisotopic mass distribution. The distribution of retention time differences was narrower and higher around 0 for correct PSMs, indicating that the scan time of correct PSMs is closer to the predicted retention time calculated by hydrophobicity of the peptides. An empirical cumulative distribution plot of the continuous variables in the dataset led me to the same conclusion as above (Figure 5). PSM with peptide charge 3 had the highest proportion of correctness, followed by charge 2 and charge 1. Charge more than or equal to 4 had the lowest proportion of correctness (Figure 6).

Figure 4: Stratified distributions of predictor variables. *The distribution of the predictor variables were shown segregated by the correctness of the PSMs. The red lines/bars are correct PSMs. The black lines/bars are incorrect PSMs.*



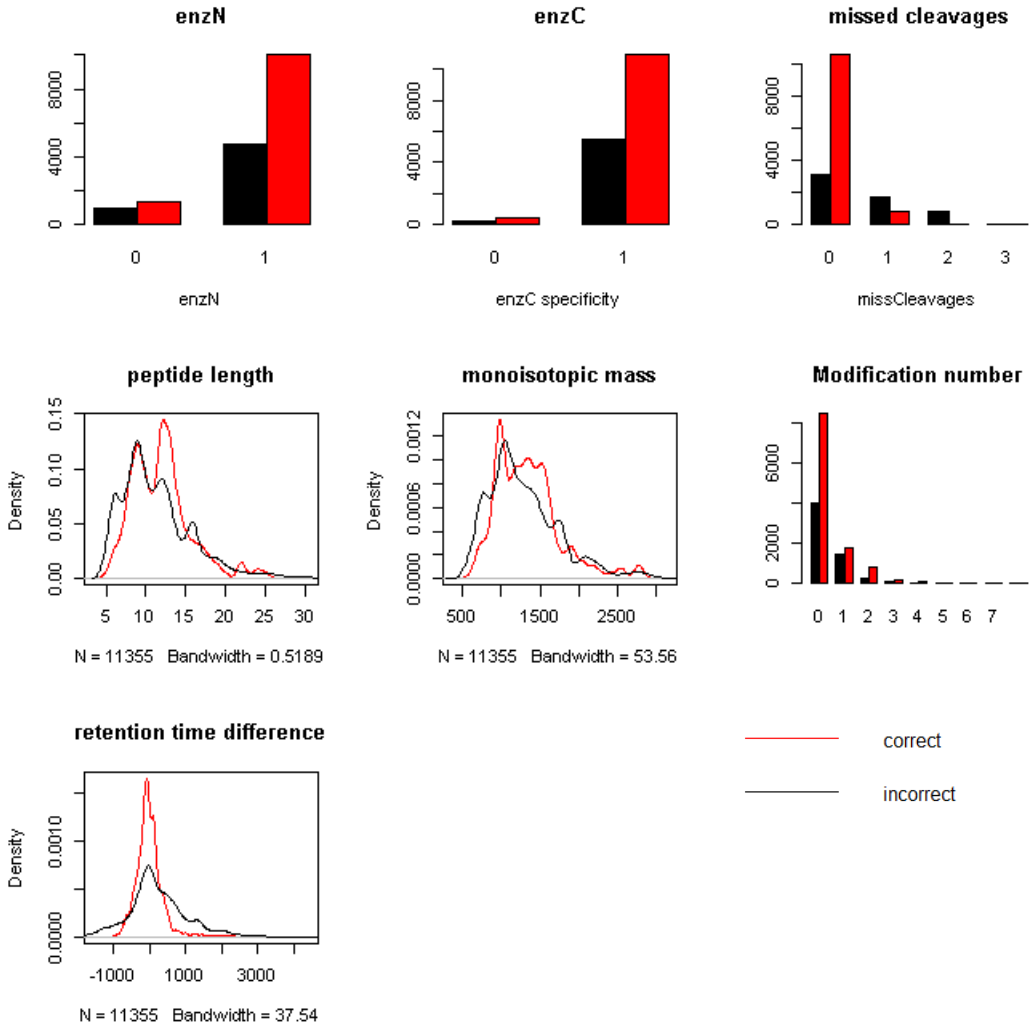
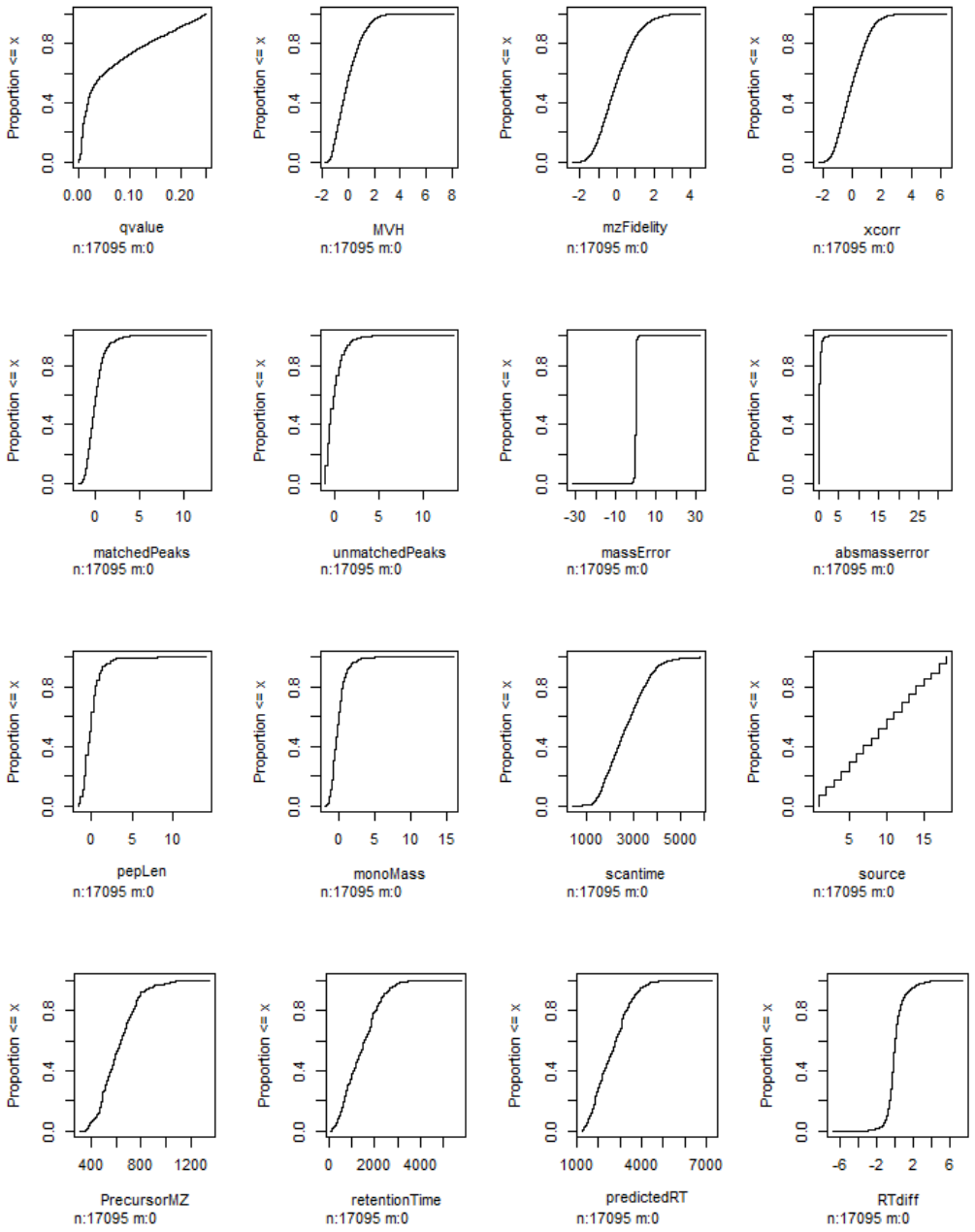

enzN
0
4000
8000
0
1
enzN
enzC
0
4000
8000
0
1
enzC specificity
missed cleavages
0
4000
8000
0
1
2
3
missCleavages
peptide length
Density
0.00
0.05
0.10
0.15
5
10
15
20
25
30
N = 11355 Bandwidth = 0.5189
monoisotopic mass
Density
0.0000
0.0006
0.0012
500
1500
2500
N = 11355 Bandwidth = 53.56
Modification number
0
2000
6000
0
1
2
3
4
5
6
7
retention time difference
Density
0.0000
0.0010
-1000
1000
3000
N = 11355 Bandwidth = 37.54
correct
incorrect

Figure 5. Empirical cumulative distribution plot of the continuous variables in the dataset



Although *MVH*, *mzFidelity* and *Xcorr* scores are all related to the quality of the same PSMs, their correlation was around 0.8 [40] (Figure 7, Figure 8). Using all three scores can provide more information

for PSM classification. The correlation between peptide length and peptide monoisotopic mass was 0.98, indicating high collinearity; therefore, only monoisotopic mass (*monomass*) was included in the models (Figure 7). R^2 of *absmasserror* and *matchedPeaks* from redundancy analysis by *redun* function in *rms* were 0.999 and 0.970 respectively, indicating these two variables could be predicted from other predictors. Therefore, these two variables were excluded from the logistic regression models to avoid matrix singularity.

Figure 6. Univariate summaries of PSM correctness. *The marginal proportion of correct PSMs is shown separately by categories of predictors.*

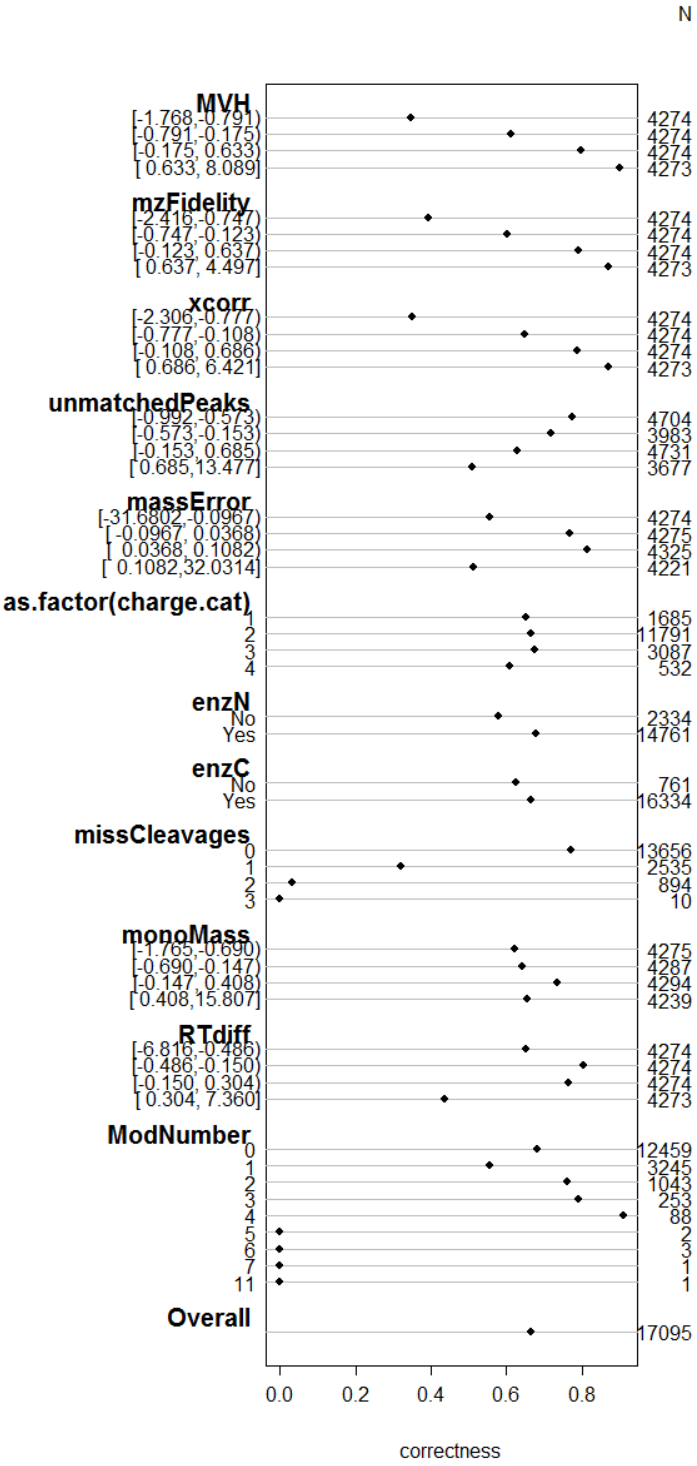


Figure 7: Correlation of all predictor variables. *The values of Spearman's rank correlation are shown for any two variables.*

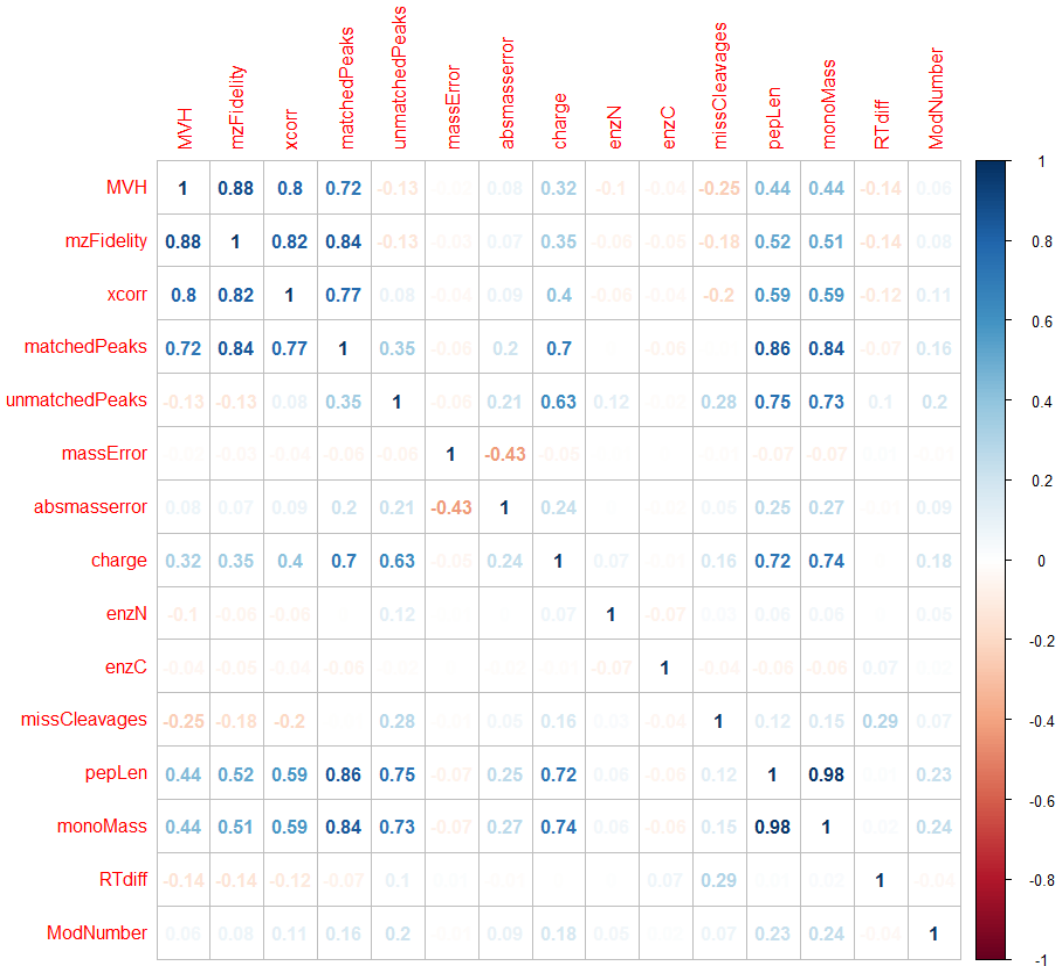
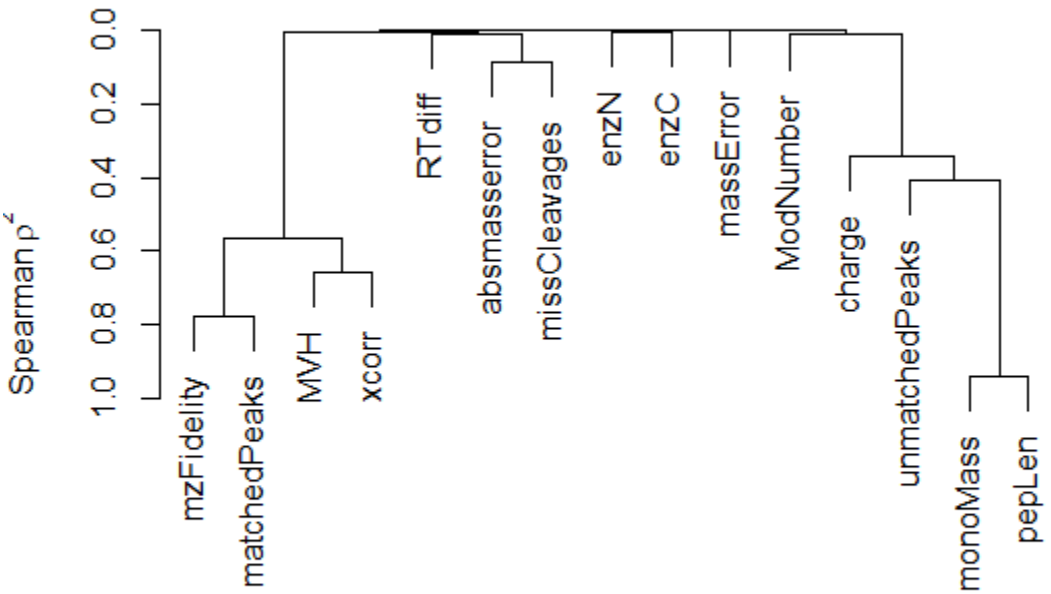


Figure 8. Variable clustering based on Spearman's rank correlation



3.2 Main Effect Logistic Regression Model Fitting and Validation

In the main effect model, all the predictors were significantly associated with the outcome label under alpha=0.05 (Table 2, Table 3, Figure 10). Among these variables, the 95% confidence interval of log odds ratios of *MVH*, *Xcorr*, *enzN*, *enzC*, *ModNumber* do not include zero, which I interpreted as evidence of a positive association between the variables and the PSM. On the other hand, as the *missCleavages*, *unmatchedPeaks*, *massError*, retention time difference increases, the odds of correctness decrease (Figure 11). The 95% confidence intervals of odds ratio of charge group 2, 3, 4 compared to charge group 1 were (0.3464, 0.4613), (0.4758, 0.7299), (0.1964, 0.4099) respectively. The spectrum with charge group 1 had the highest odds of being correct in this dataset, while the charge greater than or equal to 4 had the lowest odds of correctness. Consistent with observations in the summary statistics, the three scores, enzyme specificity, retention time difference, and mass error were good predictors to separate correct PSMs from incorrect ones. There might be interactions between predictors or non-linear relationship between predictors and the outcome. The effects of these 6 predictors (*MVH*, *massError*, *RTdiff*, *ModNumber*, *enzN*, *enzC*) on correctness of PSM vary among different charge groups (Figure 9). Therefore, it was reasonable to include interaction terms between *charge.cat* and these 6 predictors.

Figure 9. Nonparametric regression estimates of the relationship between predictors of interest and the probability of PSM correctness. *The relationship is stratified by four charge categories. The curves appear different for different charges in each predictor. Therefore, it is reasonable to include interaction between charge and other predictors.*

The effects of MVH, massError, RTdiff, ModNumber, enzN, enzC are stratified by charge (black: charge 1, red: charge 2, green: charge 3, blue: charge >=4).

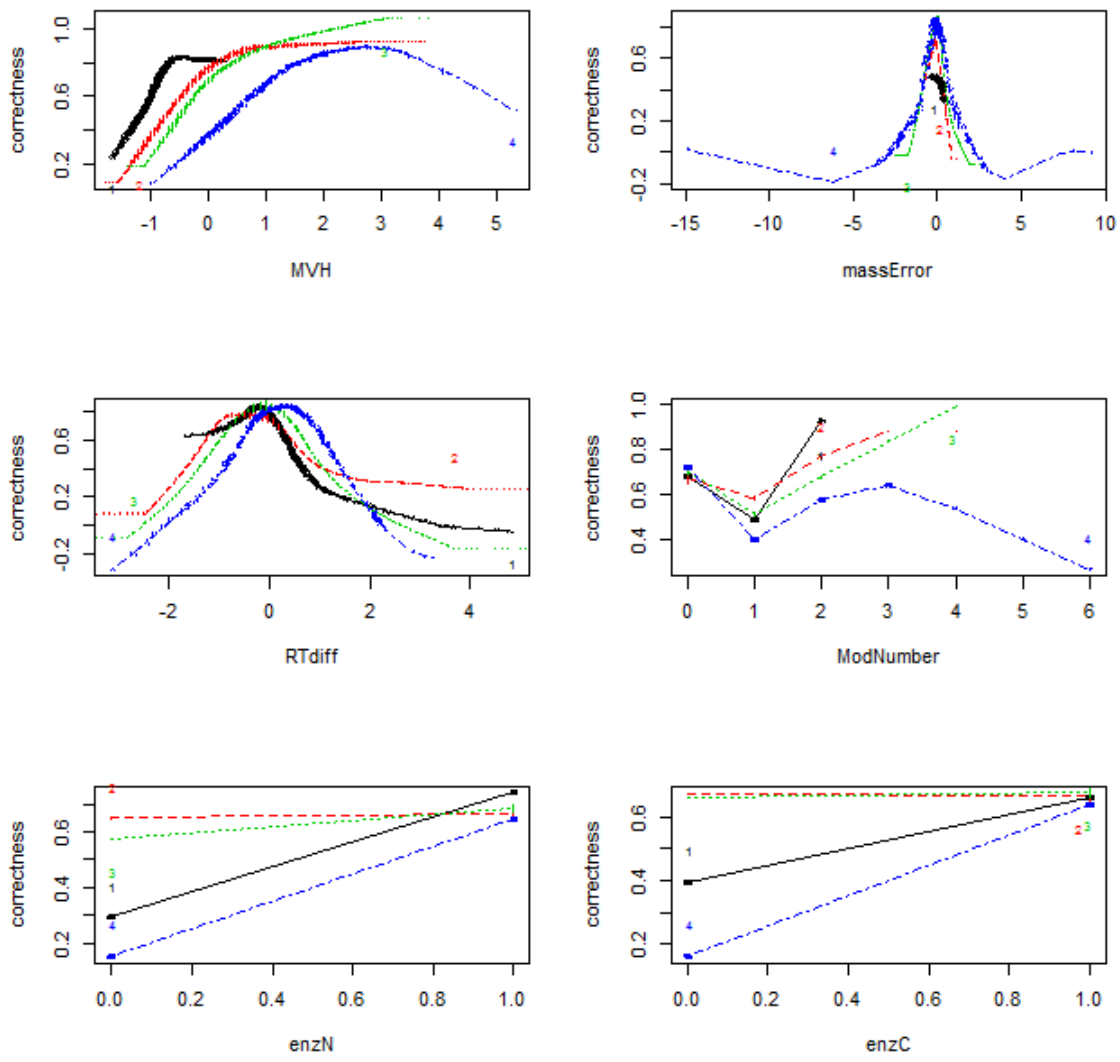


Table 2. Estimate and 95% confidence interval of correlation coefficients of the logistic regression model with main effects. *MVH, Xcorr, unmatchedPeaks, massError, charge.cat, enzN, enzC, missCleavages, RTdiff and ModNumber are highly associated with correctness of spectra.*

	Estimate	Std.Error	95% CI lower bound	95% CI upper bound
(Intercept)	0.4585	0.1241	0.2153	0.7017
MVH	0.6789	0.054	0.5732	0.7847
mzFidelity	-0.1199	0.0753	-0.2675	0.0276
Xcorr	1.0561	0.0465	0.9651	1.1472
unmatchedPeaks	-0.3869	0.0744	-0.5328	-0.241
massError	-0.0533	0.0237	-0.0998	-0.0068
as.factor(charge.cat)2	-0.9172	0.0731	-1.0605	-0.7738
as.factor(charge.cat)3	-0.5288	0.1091	-0.7427	-0.3149
as.factor(charge.cat)4	-1.2597	0.1877	-1.6276	-0.8919
enzN	1.3577	0.06	1.2401	1.4752
enzC	0.5605	0.0971	0.3701	0.7508
missCleavages	-1.6896	0.0509	-1.7895	-1.5897
monoMass	-0.159	0.0832	-0.3222	0.0041
RTdiff	-0.0542	0.0215	-0.0963	-0.0121
ModNumber	0.1269	0.0322	0.0637	0.1901

Figure 10. Interquartile-range odds ratios for continuous predictors and simple odds ratios for categorical predictors. *Numbers at left are upper quartile : lower quartile or current group : reference group. The bars represent 0:9; 0:95; 0:99 confidence limits. The intervals are drawn on the log odds ratio scale and labeled on the odds ratio scale. Ranges are on the original scale.*

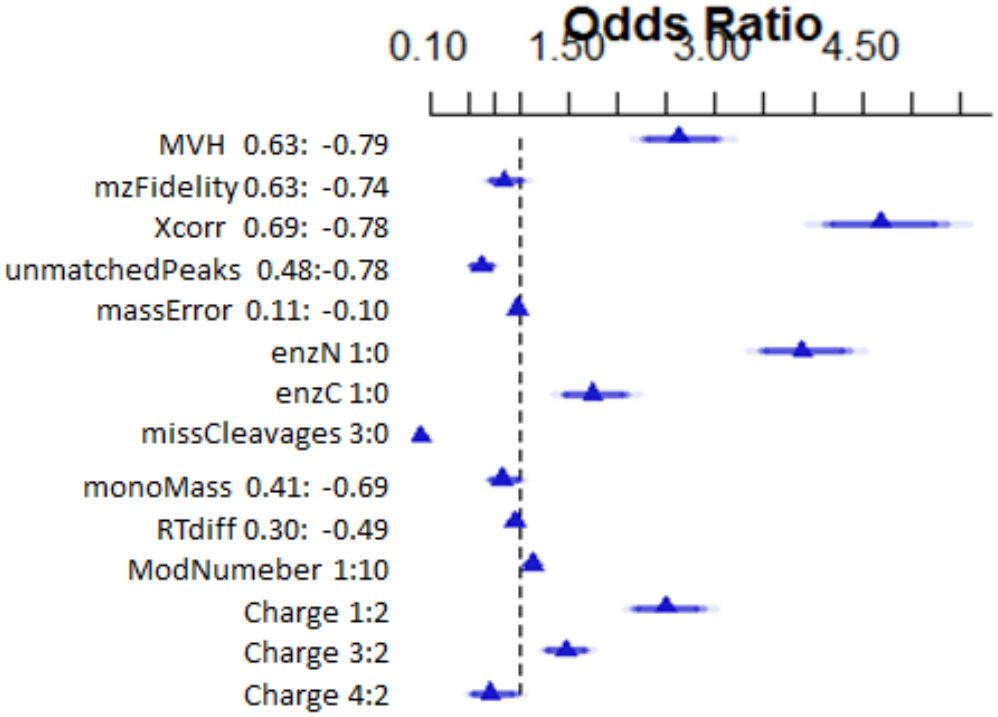


Figure 11. Plot of effects of variables estimated by main effect model. *Odds of correct PSM increase when xcorr or MVH increases, decrease when unmatchedPeaks and missCleavages increase.*

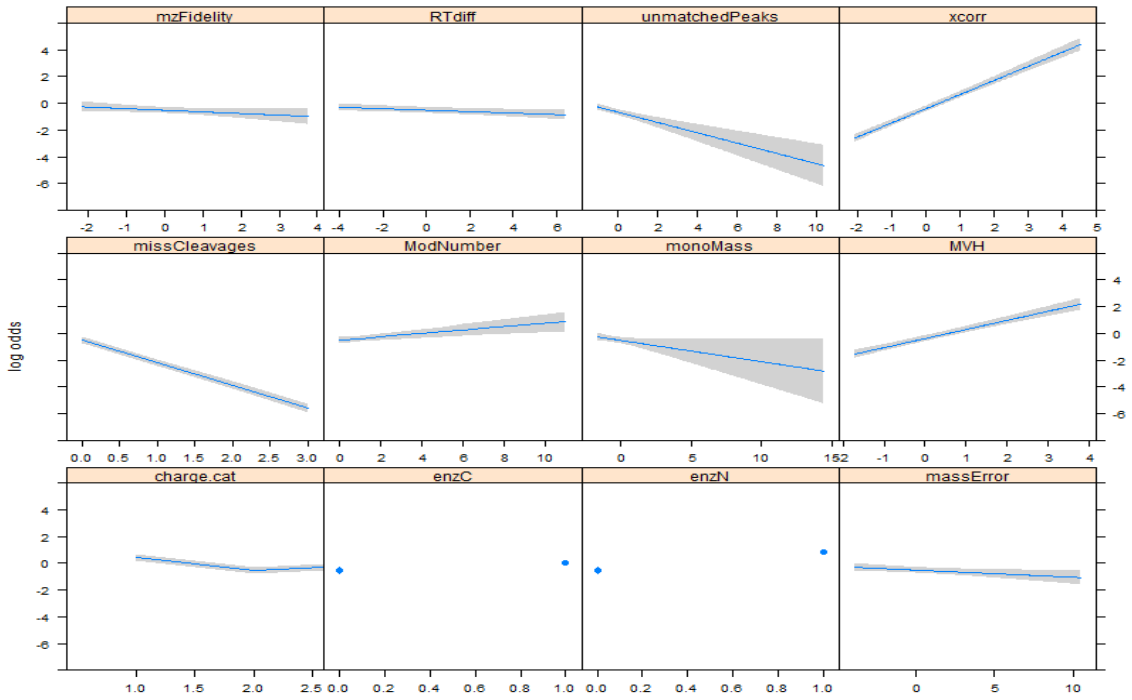
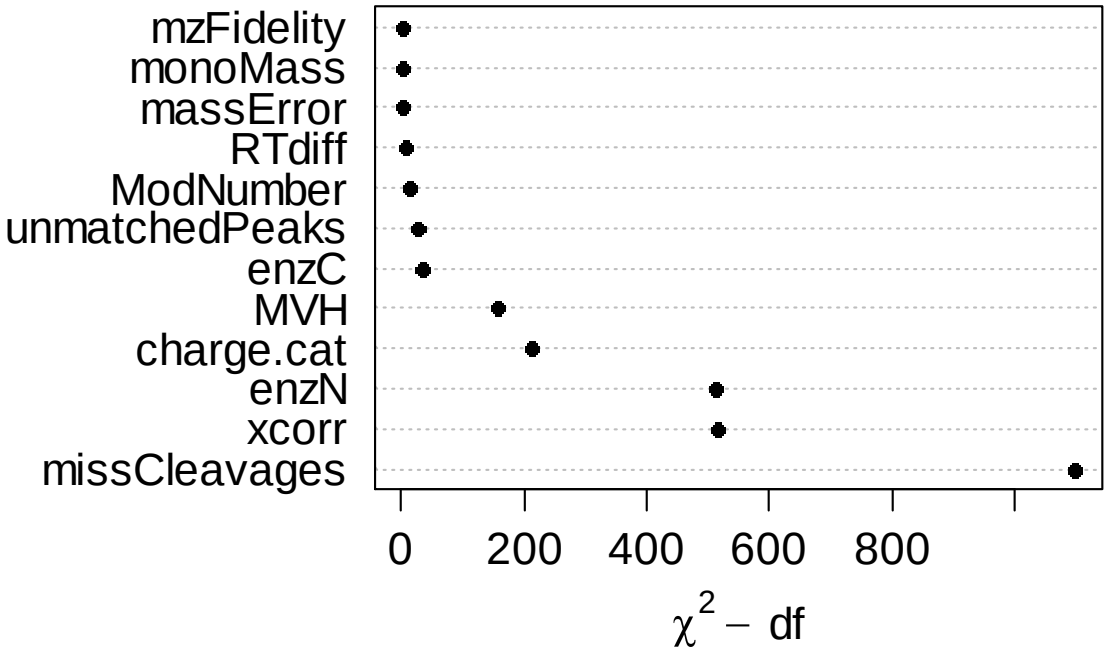


Table 3. Wald statistics and p-values of predictors in the main effect model

	χ^2	$d.f.$	P
MVH	158.33	1	< 0.0001
mzFidelity	2.54	1	0.1110
xcorr	516.49	1	< 0.0001
unmatchedPeaks	27.02	1	< 0.0001
massError	5.05	1	0.0246
charge.cat	213.52	3	< 0.0001
enzN	512.45	1	< 0.0001
enzC	33.30	1	< 0.0001
missCleavages	1099.75	1	< 0.0001
monoMass	3.65	1	0.0561
RTdiff	6.36	1	0.0117
ModNumber	15.49	1	0.0001
TOTAL	3641.98	14	< 0.0001

Figure 12. Ranking of apparent importance of predictors by χ^2 *-df* of spectrum correctness in logistic regression model with main effects. *MissCleavages has the highest apparent importance followed by xcorr, enzN and charge.cat.*



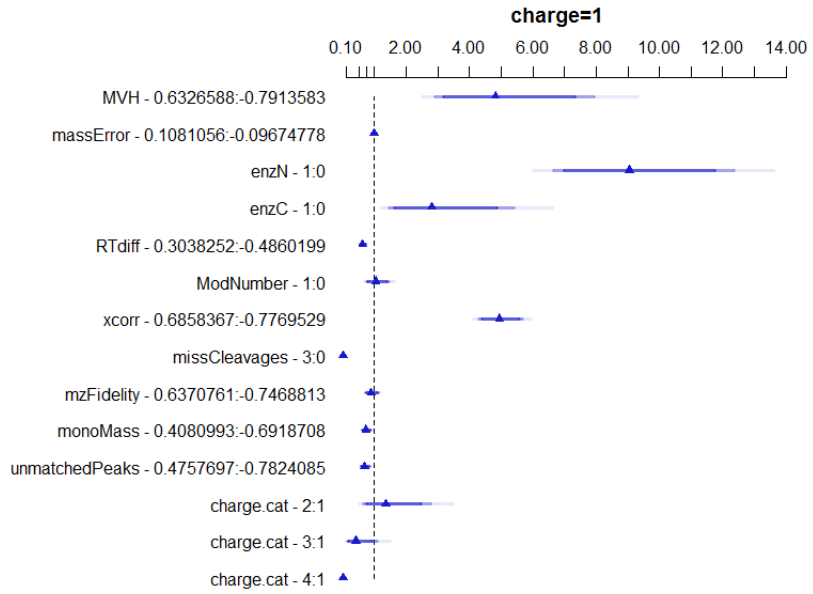
The importance of predictors were compared by $\chi 2 - df$ and ranked from lowest to highest (Figure 12). *Misscleavages* had the highest importance with $\chi 2 - df$ value over 1000. *Xcorr* had the second highest importance followed by *charge.cat*, *MVH*. *mzFidelity* and *massError* had low importance probably because their effects are masked by other correlating variables. *MonoMass* did not have high predictive importance since it was obvious that higher or lower monoisotopic mass should not determine the correctness of spectra.

3.3 Logistic Regression with Interactions Model Fitting and Validation

The estimates and confidence intervals of odds ratio of *MVH*, *enzN*, *enzC* were similar in charge group 1 and 3 (Figure 13.1, Figure 13.3, Supplementary table 2). The estimates were more precise in charge group 2 (Figure 13.2). In charge >=4 group, the confidence interval of odds ratio of charge 1, 2 or 3 vs. charge 4 was very wide. This might be due to the data of PSMs in the charge >=4 group were very noisy.

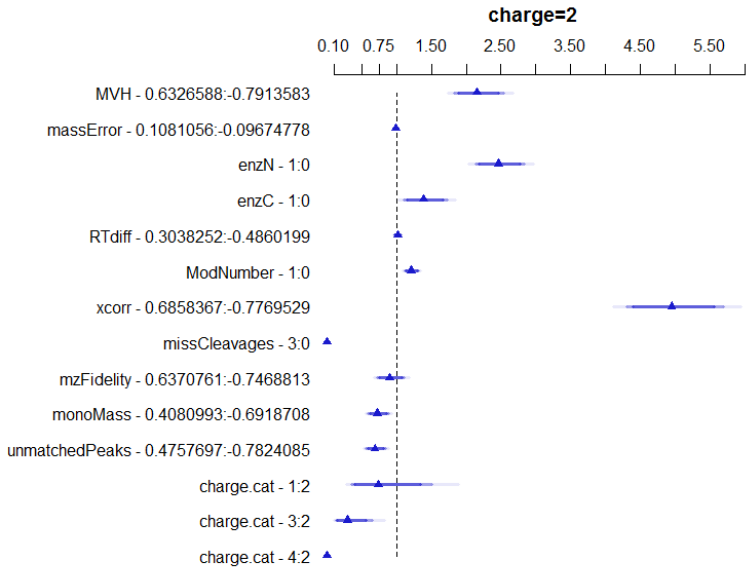
Additionally, the number of observations in this group was small and the odds of correctness differed a lot across individual charge groups.

Figure 13. Interquartile-range odds ratios for continuous predictors and simple odds ratios for categorical predictors in logistic regression model with interactions. *The odds ratios were plotted separately for charge=1, charge=2 and charge =3 groups. To evaluate the effects of different charge groups, continuous variables were adjusted to its medium, and categorical variables were adjusted to the default value 0. The odds ratios of charge.cat were plotted with interacting variables adjusted to: MVH=-0.1754578 massError=0.03677613 enzN=0 enzC=0 RTdiff=-0.1505191 ModNumber=0. The effects were similar for charge =1 and charge =3. The confidence intervals of the effects in charge=2 group were narrower, indicating higher precision. Charge>=4 group was not shown because estimates and confidence intervals were obtained from data with a small number of noisy observations.*
14.1



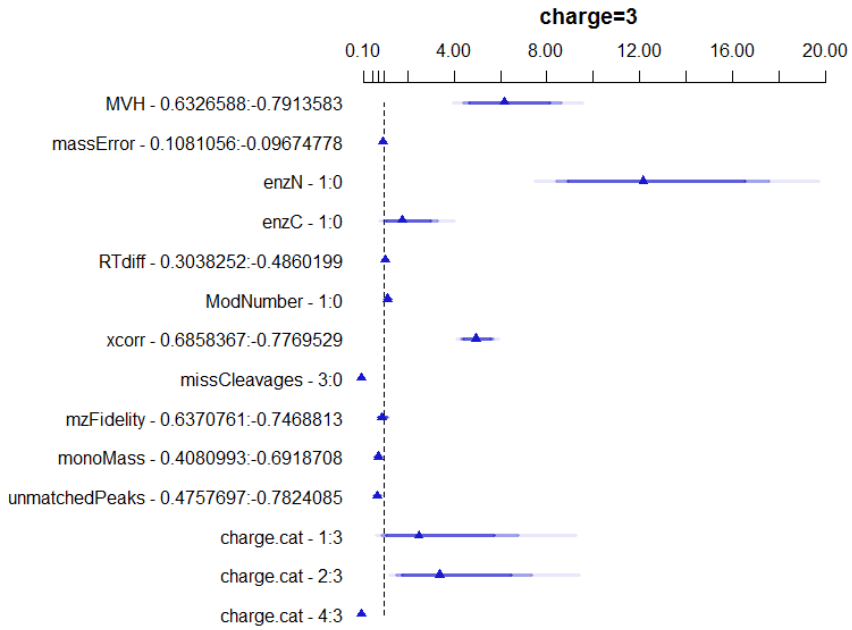
arge.cat=1 MVH=-0.1754578 massError=0.03677613 enzN=0 enzC=0 RTdiff=-0.1505191 ModNumber=0

14.2



arge.cat=2 MVH=-0.1754578 massError=0.03677613 enzN=0 enzC=0 RTdiff=-0.1505191 ModNumber=0

14.3

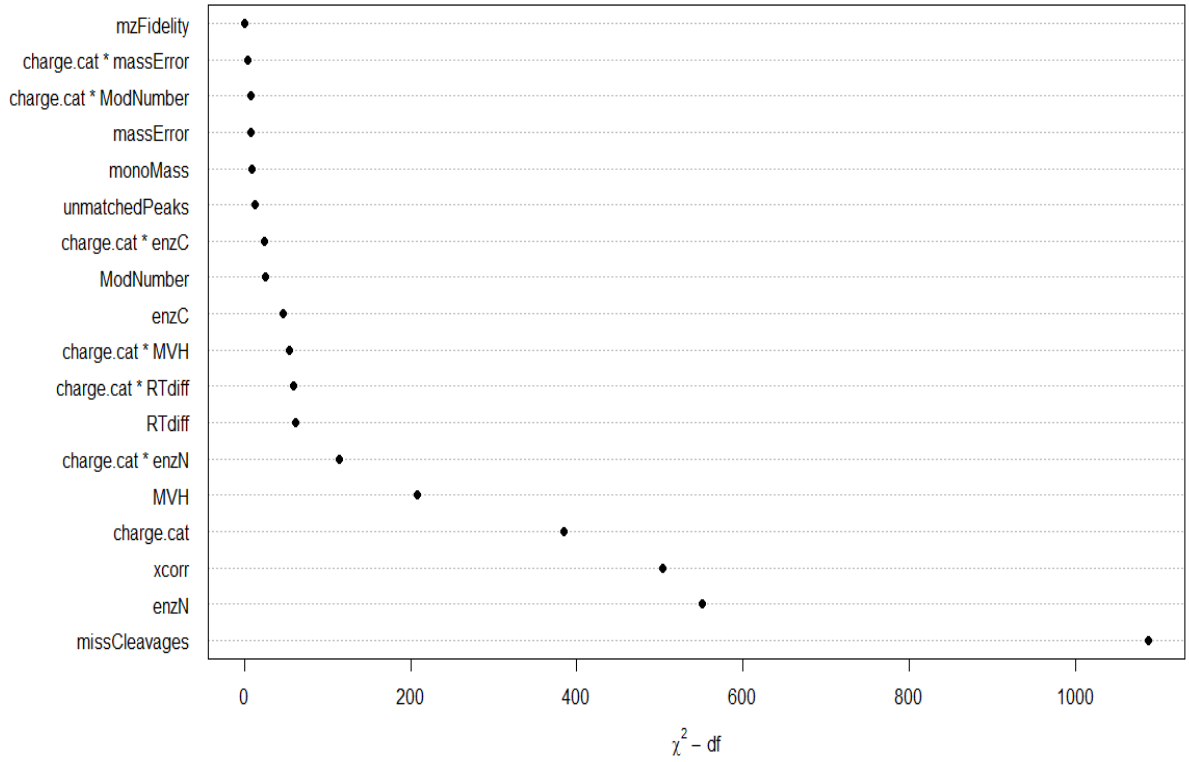


arge.cat=3 MVH=-0.1754578 massError=0.03677613 enzN=0 enzC=0 RTdiff=-0.1505191 ModNumber=0

Table 4. Wald statistics of predictors in logistic regression model with interactions

	χ^2	$d.f.$	P
charge.cat (Factor+Higher Order Factors)	405.25	21	< 0.0001
All Interactions	225.51	18	< 0.0001
MVH (Factor+Higher Order Factors)	211.39	4	< 0.0001
All Interactions	56.56	3	< 0.0001
massError (Factor+Higher Order Factors)	11.50	4	0.0215
All Interactions	6.96	3	0.0732
enzN (Factor+Higher Order Factors)	554.74	4	< 0.0001
All Interactions	116.64	3	< 0.0001
enzC (Factor+Higher Order Factors)	50.34	4	< 0.0001
All Interactions	27.12	3	< 0.0001
RTdiff (Factor+Higher Order Factors)	64.96	4	< 0.0001
All Interactions	62.60	3	< 0.0001
ModNumber (Factor+Higher Order Factors)	29.34	4	< 0.0001
All Interactions	10.34	3	0.0159
xcorr	504.82	1	< 0.0001
missCleavages	1089.35	1	< 0.0001
mzFidelity	0.97	1	0.3243
monoMass	10.35	1	0.0013
unmatchedPeaks	14.23	1	0.0002
charge.cat $\times$ MVH (Factor+Higher Order Factors)	56.56	3	< 0.0001
charge.cat $\times$ massError (Factor+Higher Order Factors)	6.96	3	0.0732
charge.cat $\times$ enzN (Factor+Higher Order Factors)	116.64	3	< 0.0001
charge.cat $\times$ enzC (Factor+Higher Order Factors)	27.12	3	< 0.0001
charge.cat $\times$ RTdiff (Factor+Higher Order Factors)	62.60	3	< 0.0001
charge.cat $\times$ ModNumber (Factor+Higher Order Factors)	10.34	3	0.0159
TOTAL INTERACTION	225.51	18	< 0.0001
TOTAL	3715.12	32	< 0.0001

Figure 14. Ranking of apparent importance of predictors by χ^2 *-df* of spectrum correctness in logistic regression model with interactions.



The Wald statistics of predictors in this model showed that all were significantly associated with the outcome label under alpha=0.05, except interaction between charge.cat and massError (Table 4). Consistent with the main effect model, missCleavages was the most important predictor followed by *enzN* and *xcorr* (Figure 14). The interaction between *charge.cat* and *enzN* was most important among all the interactions terms with χ2 –df value over 150. *RTdiff* had higher importance in this model than in the main effect model.

3.4 Logistic Regression with Splines Model Fitting and Validation

The Wald statistics of predictors in this model showed that all except non-linear effect of unmatched peaks were significantly associated with the outcome label under alpha=0.05 (Supplementary table 4, Table 5, Figure 15). *MissCleavages* was the most important predictor followed by *RTdiff* and *xcorr* (Table 5, Figure 16). The effect of *RTdiff* on the outcome was non-linear. *RTdiff* and its non-linear terms had the second highest importance with $\chi 2 - df$ value over 400. The absolute difference between the observed retention time of a spectrum and the expected retention time of the matching peptide was negatively associated with correctness of the match. It did not matter if the observed time was higher or lower than the calculated time, therefore the relationship was non-linear. The log odds of correctness of spectra decreased when RTdiff was farther away from zero (Figure 17).

Table 5. Wald statistics of logistic regression model with restricted cubic spline of predictors with 3 knots

	χ^2	$d.f.$	P
MVH	71.34	4	< 0.0001
Nonlinear	22.67	3	< 0.0001
mzFidelity	113.68	4	< 0.0001
Nonlinear	86.95	3	< 0.0001
xcorr	428.63	4	< 0.0001
Nonlinear	137.54	3	< 0.0001
unmatchedPeaks	92.22	4	< 0.0001
Nonlinear	2.63	3	0.4529
massError	294.20	4	< 0.0001
Nonlinear	290.94	3	< 0.0001
enzN	350.03	1	< 0.0001
enzC	41.09	1	< 0.0001
missCleavages	857.73	1	< 0.0001
monoMass	62.20	4	< 0.0001
Nonlinear	21.59	3	0.0001
RTdiff	416.66	4	< 0.0001
Nonlinear	389.90	3	< 0.0001
ModNumber	22.14	1	< 0.0001
charge.cat	195.37	3	< 0.0001
TOTAL NONLINEAR	1352.81	21	< 0.0001
TOTAL	4222.83	35	< 0.0001

Figure 15. Interquartile-range odds ratios for continuous predictors and simple odds ratios for categorical predictors in logistic regression model with splines. *This plot is similar as Figure 11 for main effect model*

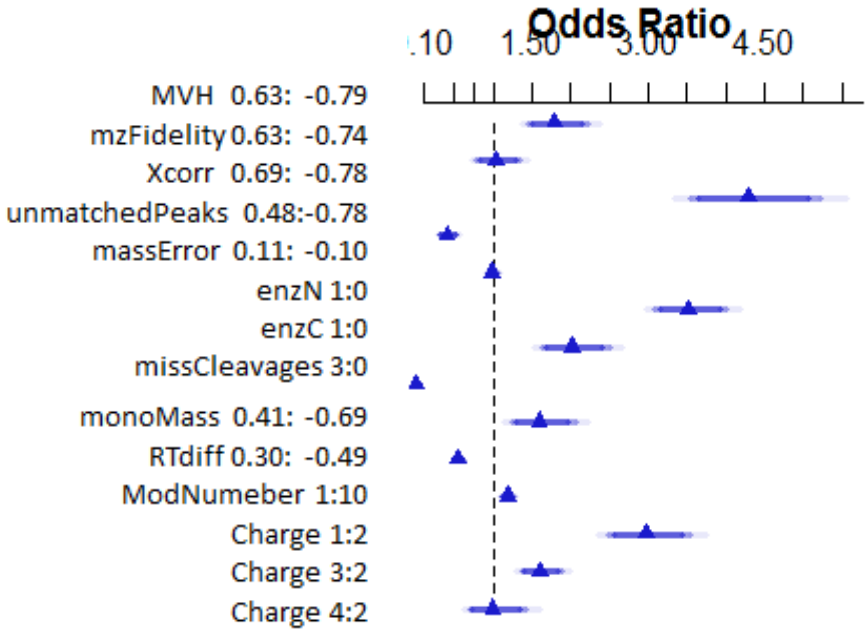


Figure 16. Ranking of apparent importance of predictors by χ^2 *-df* of spectrum correctness in logistic regression model with splines

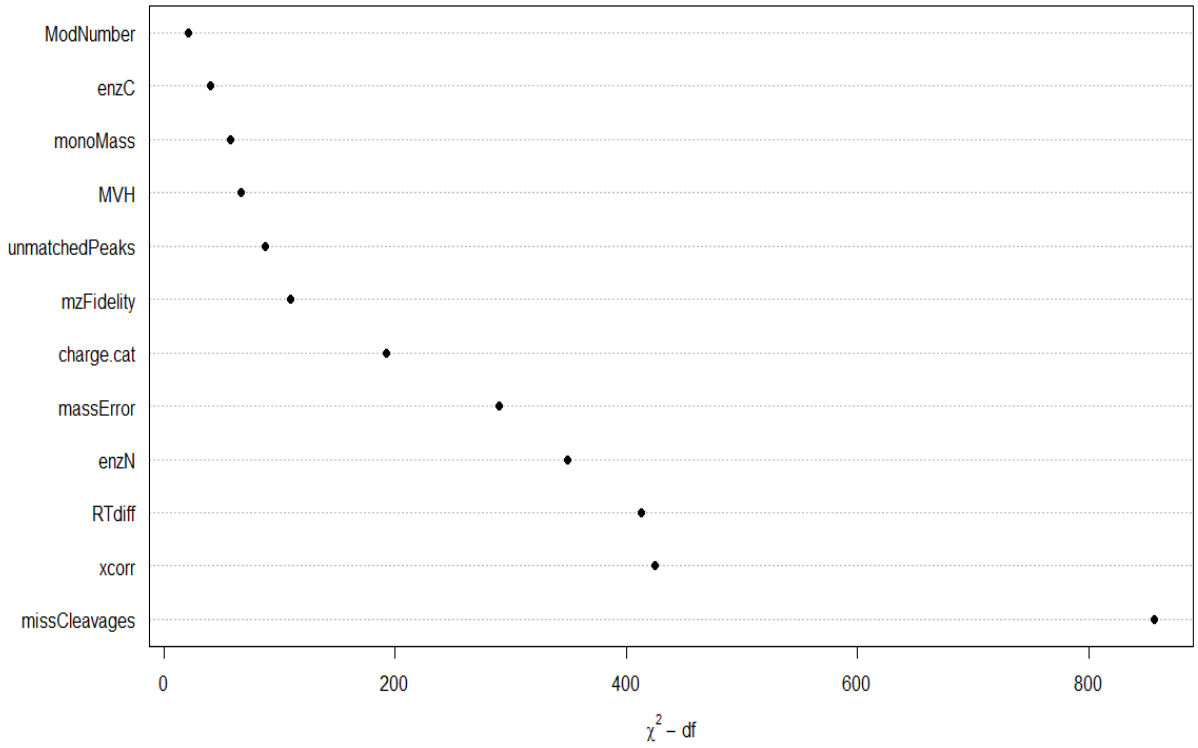
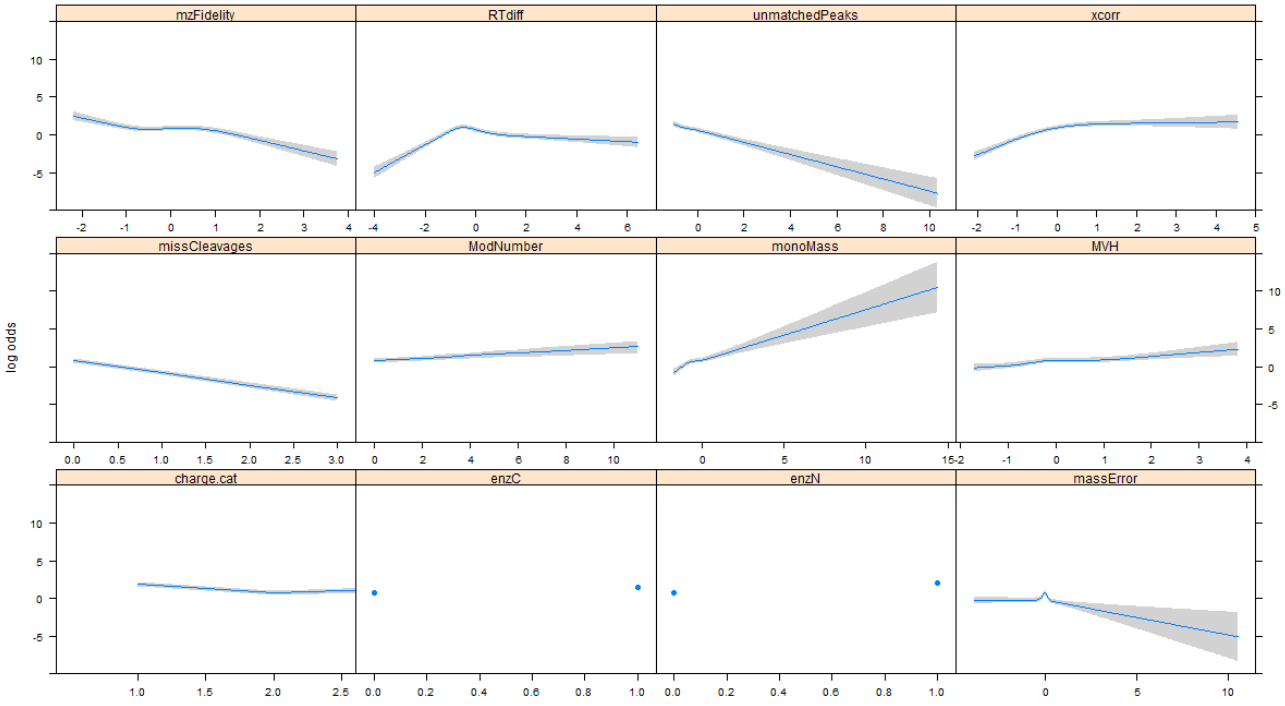


Figure 17. Plot of effects of variables estimated by main effect model. *Odds of correct PSM increase when xcorr or MVH increases, decrease when unmatchedPeaks and missCleavages increase. The odds of correct PSM increase with an increase in RTdiff when RTdiff is below 0, decrease with an increase in RTdiff when RTdiff is above 0. It is consistent with the fact that smaller absolute retention time difference in observation is associated with higher odds of correct PSMs. This plot demonstrates the existence of non-linear effect for some predictors.*



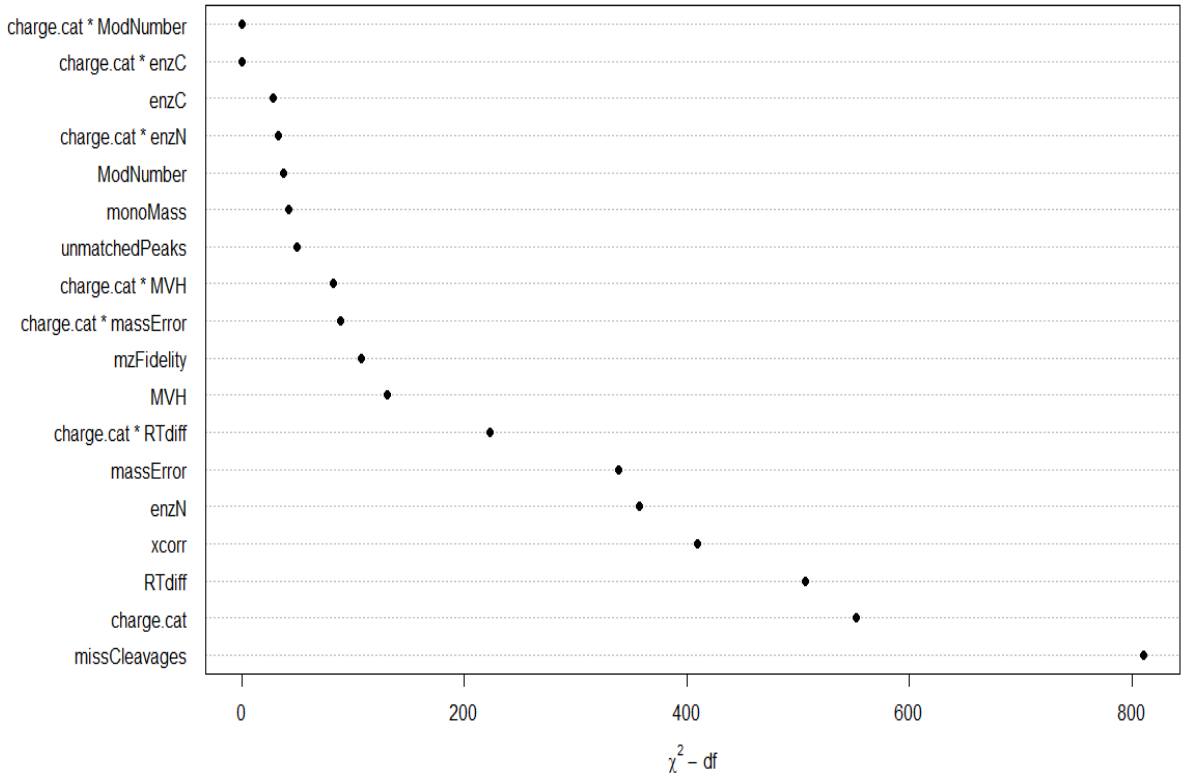
3.5 Logistic Regression with Splines and Interactions Model Fitting and Validation

The Wald statistics of predictors in this model showed that all except interactions between *charge.cat* and *modNumber*/*enzC* were significantly associated with the outcome label under alpha=0.05 (Supplementary table 5, Table 6, Figure 18). MissCleavages was the most important predictor followed by charge.cat and RTdiff (Figure 18). The results of this model also demonstrated the effect of RTdiff on the outcome was non-linear (Table 6).

Table 6. Wald statistics of predictors of logistic regression model with restricted cubic spline of predictors and interactions

	χ^2	*d.f.*	*P*
charge.cat (Factor+Higher Order Factors)	599.95	48	< 0.0001
All Interactions	442.59	45	< 0.0001
MVH (Factor+Higher Order Factors)	147.17	16	< 0.0001
All Interactions	94.04	12	< 0.0001
Nonlinear (Factor+Higher Order Factors)	66.80	12	< 0.0001
massError (Factor+Higher Order Factors)	354.44	16	< 0.0001
All Interactions	100.54	12	< 0.0001
Nonlinear (Factor+Higher Order Factors)	344.96	12	< 0.0001
RTdiff (Factor+Higher Order Factors)	522.67	16	< 0.0001
All Interactions	234.81	12	< 0.0001
Nonlinear (Factor+Higher Order Factors)	456.53	12	< 0.0001
ModNumber (Factor+Higher Order Factors)	41.39	4	< 0.0001
All Interactions	2.99	3	0.3924
enzN (Factor+Higher Order Factors)	360.98	4	< 0.0001
All Interactions	35.57	3	< 0.0001
enzC (Factor+Higher Order Factors)	31.75	4	< 0.0001
All Interactions	3.50	3	0.3208
missCleavages	811.77	1	< 0.0001
mzFidelity	111.38	4	< 0.0001
Nonlinear	97.88	3	< 0.0001
xcorr	413.76	4	< 0.0001
Nonlinear	89.47	3	< 0.0001
monoMass	46.18	4	< 0.0001
Nonlinear	41.62	3	< 0.0001
unmatchedPeaks	53.17	4	< 0.0001
Nonlinear	36.89	3	< 0.0001
charge.cat × MVH (Factor+Higher Order Factors)	94.04	12	< 0.0001
Nonlinear	27.89	9	0.0010
Nonlinear Interaction : f(A,B) vs. AB	27.89	9	0.0010
charge.cat × massError (Factor+Higher Order Factors)	100.54	12	< 0.0001
Nonlinear	89.76	9	< 0.0001
Nonlinear Interaction : f(A,B) vs. AB	89.76	9	< 0.0001
charge.cat × RTdiff (Factor+Higher Order Factors)	234.81	12	< 0.0001
Nonlinear	190.35	9	< 0.0001
Nonlinear Interaction : f(A,B) vs. AB	190.35	9	< 0.0001
charge.cat × ModNumber (Factor+Higher Order Factors)	2.99	3	0.3924
charge.cat × enzN (Factor+Higher Order Factors)	35.57	3	< 0.0001
charge.cat × enzC (Factor+Higher Order Factors)	3.50	3	0.3208
TOTAL NONLINEAR	1542.33	48	< 0.0001
TOTAL INTERACTION	442.59	45	< 0.0001
TOTAL NONLINEAR + INTERACTION	1677.07	66	< 0.0001
TOTAL	4043.62	80	< 0.0001

Figure 18. Ranking of apparent importance of predictors by χ^2 *-df* of spectrum correctness in logistic regression model with splines and interactions



The AIC's of *modelReg*, *modelReg.inter*, *modelReg.spline*, *modelReg.spline.inter1* and *modelReg. Spline.inter 2* were 14500, 13980, 12890, 12170 and 11546 respectively. The model with splines with the most interaction terms (modelReg. Spline.inter 2) had the lowest AIC. The log odds ratios of these predictors and their confidence intervals from the four models were shown in supplementary tables. Because there were too many predictors in modelReg.spline.inter2 model, the estimates and confidence intervals of the log odds ratio were not shown.

3.6 Logistic Regression Model Validation

The main effect model worked comparably well with an average AUC of 0.86 in both training and test sets (Table 7, Table 8). Linear model with interaction had slightly better AUC (0.87) in training sets but the same AUC 0.86 in test sets. Spline model without interaction had higher AUC -0.88, 0.87 in training and test sets. Spline model with interactions had the highest AUC 0.90 in training sets but the number slightly decreased in test sets (0.87). The performances of all these models in training sets were approximately the same as the test sets, indicating there was not much overfitting.

Table 7. Performance measures of logistic regression models in training sets

	Sensitivity	specificity	precision	Accuracy	F-measure	AUC
linear main effect model	0.92 (0.92,0.93)	0.62 (0.60,0.64)	0.83 (0.82,0.83)	0.82 (0.82,0.83)	0.87 (0.87,0.88)	0.86 (0.85,0.87)
linear model with interaction	0.93 (0.92,0.93)	0.63 (0.61,0.65)	0.83 (0.83,0.84)	0.83 (0.82,0.83)	0.88 (0.87,0.88)	0.87 (0.86,0.87)
spline model	0.94 (0.93,0.94)	0.70 (0.68,0.71)	0.86 (0.85,0.86)	0.86 (0.85,0.86)	0.90 (0.89,0.90)	0.88 (0.88,0.89)
spline model with interaction	0.94 (0.94,0.94)	0.72 (0.71,0.73)	0.87 (0.86,0.87)	0.87 (0.86,0.87)	0.90 (0.90,0.91)	0.90 (0.89,0.90)

Table 8. Performance measures of logistic regression models in test sets

	sensitivity	specificity	precision	accuracy	F-measure	AUC
linear main effect model	0.92 (0.91,0.93)	0.62 (0.60,0.64)	0.83 (0.82,0.84)	0.82 (0.81,0.83)	0.87 (0.87,0.88)	0.86 (0.85,0.87)
linear model with interaction	0.92 (0.91,0.93)	0.63 (0.60,0.65)	0.83 (0.82,0.84)	0.82 (0.82,0.83)	0.87 (0.87,0.88)	0.86 (0.85,0.87)
spline model	0.93 (0.87,0.96)	0.67 (0.54,0.75)	0.85 (0.81,0.88)	0.84 (0.81,0.86)	0.89 (0.86,0.90)	0.87 (0.82,0.89)
spline model with interaction	0.92 (0.85,0.96)	0.69 (0.56,0.76)	0.85 (0.81,0.88)	0.84 (0.80,0.86)	0.88 (0.85,0.90)	0.87 (0.83,0.89)

I validated these four models with 200 bootstrap validation using *validate* function in *rms* package (Table 9). The most flexible model *ModelReg.spline.inter* (Table 9.4, Table 9.5) had the highest D_{xy}, R^2 , D, Q and g-index. Overall, the simplest main effect model *modelReg* had the worst predictive accuracy indices

compared to other models. Additive model with splines (*modelReg.spline*, Table 9.3) had higher accuracy indices than the one with interactions (*modelReg.inter*, Table 9.2). The optimism values were small in the four models, also indicating there was not much overfitting in the models. The model with most interactions terms (*ModelReg.spline.inter2*, Table 9.5) had the highest corrected D_{xy}: 0.8037. The main effect model had the best slope shrinkage factor 0.9973 (Table 9.1).

Table 9. Indices of predictive accuracy in logistic regression models

9.1. Indices of predictive accuracy in linear main effect model

	index.orig	training	test	optimism	index.corrected
Dxy	0.72	0.72	0.72	0.00	0.72
R2	0.48	0.49	0.48	0.00	0.48
Intercept	0.00	0.00	0.00	0.00	0.00
Slope	1.00	1.00	1.00	0.00	1.00
Emax	0.00	0.00	0.00	0.00	0.00
D	0.43	0.43	0.43	0.00	0.43
U	0.00	0.00	0.00	0.00	0.00
Q	0.43	0.43	0.43	0.00	0.43
B	0.13	0.13	0.13	0.00	0.13
g	2.26	2.26	2.25	0.01	2.25
gp	0.32	0.32	0.32	0.00	0.32

9.2 Indices of predictive accuracy in Linear model with interactions

	index.orig	training	test	optimism	index.corrected
Dxy	0.73	0.73	0.73	0.00	0.73
R2	0.50	0.50	0.50	0.00	0.49
Intercept	0.00	0.00	0.01	-0.01	0.01
Slope	1.00	1.00	0.99	0.01	0.99
Emax	0.00	0.00	0.00	0.00	0.00
D	0.45	0.45	0.44	0.00	0.44
U	0.00	0.00	0.00	0.00	0.00
Q	0.45	0.45	0.44	0.00	0.44
B	0.13	0.13	0.13	0.00	0.13
g	2.36	2.37	2.35	0.03	2.33
gp	0.33	0.33	0.33	0.00	0.33

9.3. Indices of predictive accuracy in additive spline model

	index.orig	training	test	optimism	index.corrected
Dxy	0.76	0.76	0.76	0.00	0.76
R2	0.57	0.57	0.57	0.01	0.56

Intercept	0.00	0.00	0.01	-0.01	0.01
Slope	1.00	1.00	0.98	0.02	0.98
Emax	0.00	0.00	0.01	0.01	0.01
D	0.53	0.53	0.53	0.01	0.52
U	0.00	0.00	0.00	0.00	0.00
Q	0.53	0.53	0.52	0.01	0.52
B	0.11	0.11	0.11	0.00	0.11
g	2.43	2.49	2.44	0.05	2.38
gp	0.35	0.35	0.35	0.00	0.35

9.4 Indices of predictive accuracy in spline model with interactions 1

	index.orig	training	test	optimism	index.corrected
Dxy	0.79	0.79	0.79	0.01	0.78
R2	0.61	0.61	0.60	0.01	0.59
Intercept	0.00	0.00	0.04	-0.04	0.04
Slope	1.00	1.00	0.94	0.06	0.94
Emax	0.00	0.00	0.02	0.02	0.02
D	0.57	0.58	0.57	0.01	0.56
U	0.00	0.00	Inf	-Inf	Inf
Q	0.57	0.58	-Inf	Inf	-Inf
B	0.11	0.10	0.11	0.00	0.11
g	3.25	3.42	3.24	0.18	3.07
gp	0.36	0.36	0.35	0.01	0.35

*In 32 out of 200 iterations, this model does not converge, leaving the corrected index of U and Q infinite.

9.5. Indices of predictive accuracy in spline model with interactions 2

	index.orig	training	test	optimism	index.corrected
Dxy	0.81	0.82	0.81	0.01	0.80
R2	0.64	0.64	0.63	0.02	0.62
Intercept	0.00	0.00	0.08	-0.08	0.08
Slope	1.00	1.00	0.91	0.09	0.91
Emax	0.00	0.00	0.03	0.03	0.03
D	0.61	0.62	0.60	0.02	0.59
U	0.00	0.00	0.02	-0.02	0.02
Q	0.61	0.62	0.58	0.04	0.57
B	0.10	0.10	0.10	0.00	0.10
g	4.40	4.65	4.29	0.36	4.04
gp	0.36	0.37	0.36	0.01	0.35

Dxy: Somers' rank correlation between predicted probability that Y = 1 vs. the binary Y values. This equals 2(C – 0.5) where C is the “ROC Area".

D: Discrimination index, likelihood ratio χ^2 divided by the sample size

Q: Logarithmic accuracy score, a scaled version of the log-likelihood achieved by the predictive model

g: g-index, measure of model's predictive discrimination based on Gini's mean difference for a variable Z.

gp: g-index on the probability scale, Gini's mean difference of $\hat{P}$

3.7 SVM and Random Forest Model Fitting and Validation

SVM with a Gaussian kernel outperformed SVM with a linear kernel as in Percolator[24, 25] with a higher mean AUC and F-measure in both training and test sets (Table 10, Table 11). Random forest and SVM with Gaussian kernels performed the best among these models. The AUC of the random forest was the highest at -1.0 and 0.99 in training and test sets respectively. The AUC of SVM Gaussian model was also very high at -1.00 and 0.97 in training and test sets respectively. SVM linear model performed similarly to logistic regression with splines and interactions, with AUC 0.87 in both training and test sets. Comparably, logistic regression with interactions achieved higher AUC in training set -0.90 . These accuracy measures were very precise with narrow bootstrap confidence intervals. Validated by 1000 bootstrap simulations, retention time difference and monoisotopic mass were most important for classification of the PSMs in random forests (Figure 19), followed by number of missed cleavages and Xcorr score. MVH score, charge (*charge.cat*), mass error (*massError*), enzyme specificity at N terminus (*enzN*) were moderately important in predictions. Enzyme specificity at C terminus (*enzC*) was the least important for prediction. I compared the apparent importance estimated by $\chi 2\text{-}df$ in Figure 12, 14, 16, 18 in logistic regression models. *MissCleavages*, *xcorr*, *enzN*, charge, RT diff are the four most important variables in the logistic regression models. According to our previous knowledge, monoisotopic mass might not be a good predictor for correctness of peptide-spectrum matches, therefore, the importance evaluated by $\chi 2\text{-}df$ in logistic regression models were more reasonable.

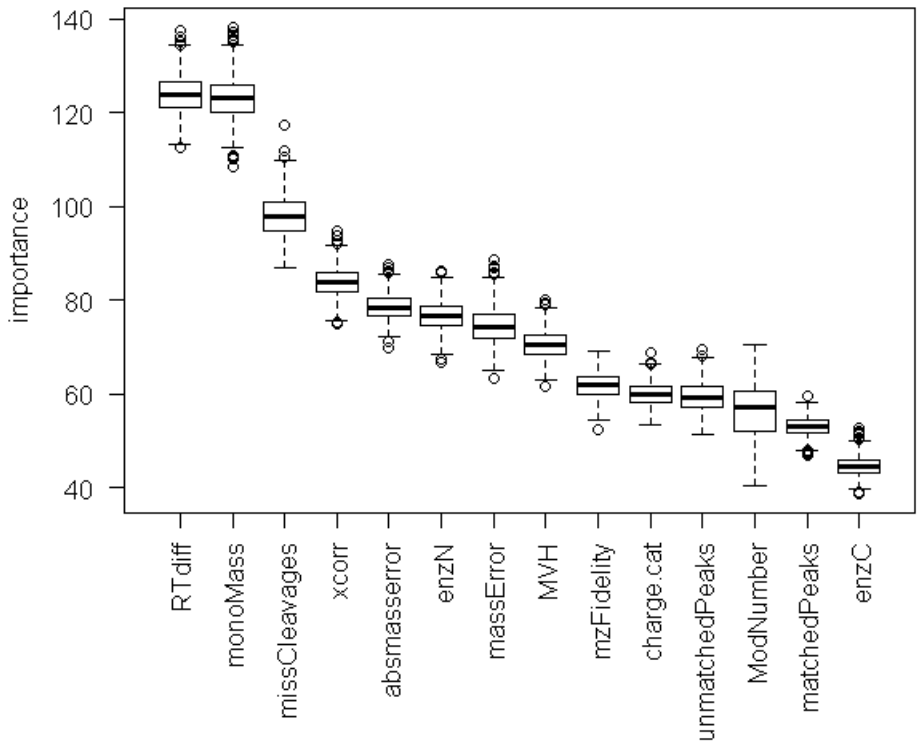
Table 10. Performance measures of the SVM model with a linear or a Gaussian kernel and random forests in training set.

	Sensitivity	specificity	precision	accuracy	Fmeasure	Auc
Spline model with interaction	0.94 (0.94,0.94)	0.72 (0.71,0.73)	0.87 (0.86,0.87)	0.87 (0.86,0.87)	0.90 (0.90,0.91)	0.90 (0.89,0.90)
SVM Gaussian	0.99 (0.99,0.99)	0.98 (0.97,0.98)	0.99 (0.98,0.99)	0.99 (0.99,0.99)	0.99 (0.99,0.99)	1.00 (0.99,1.00)
SVM linear	0.94 (0.94,0.95)	0.65 (0.62,0.67)	0.85 (0.83,0.86)	0.84 (0.83,0.85)	0.90 (0.88,0.92)	0.87 (0.87,0.88)
RF	1.00 (1.00,1.00)	1.00 (1.00,1.00)	1.00 (1.00,1.00)	1.00 (1.00,1.00)	1.00 (1.00,1.00)	1.00 (1.00,1.00)

Table 11. Performance measures of the SVM model with a linear or a Gaussian kernel and random forests in test set (OOB set).

	Sensitivity	specificity	precision	accuracy	Fmeasure	Auc
Spline model with interaction	0.92 (0.85,0.96)	0.69 (0.56,0.76)	0.85 (0.81,0.88)	0.84 (0.80,0.86)	0.88 (0.85,0.90)	0.87 (0.83,0.89)
SVM Gaussian	0.95 (0.94,0.96)	0.91 (0.89,0.92)	0.94 (0.89,0.96)	0.94 (0.93,0.94)	0.95 (0.92,0.96)	0.97 (0.96,0.97)
SVM linear	0.94 (0.93,0.95)	0.64 (0.62,0.66)	0.85 (0.83,0.88)	0.84 (0.84,0.85)	0.90 (0.88,0.93)	0.87 (0.86,0.88)
RF	0.97 (0.96,0.98)	0.88 (0.87,0.90)	0.94 (0.94,0.95)	0.94 (0.94,0.95)	0.96 (0.95,0.96)	0.99 (0.98,0.99)

Figure 19. Variable importance boxplot by the random forests model. *Variable importance values were computed using the mean decrease in the Gini index, and expressed relative to the maximum.*



IV. DISCUSSION

This study is the first attempt to compare the performances of logistic regression, support vector machines, and random forests methods for spectrum discrimination. This study is also novel for applying logistic regression to this type of study and demonstrating its advantages compared to other methods. It is also the first study that demonstrates that a logistic regression model with spline functions and interactions perform as well as (if not better than) SVM with a linear kernel which is currently widely used on proteomics data and is preferable due to faster computing time and better interpretability.

In a production proteomics laboratory, researchers must often create large lists of peptide identifications based on various confidence statistics generated by search engines. The common methodology for selecting correct peptide-spectrum matches from incorrect ones is based on arbitrary thresholds. This could lead to a lot of false discoveries. Statistical learning algorithms such as logistic regression and random forests provide a more accurate method for spectrum classification. Additional newer scoring criteria continue to be published to improve discrimination of correct search results from incorrect ones. These newer scores can complement the current scoring measures generated by popular search engines such as Mascot and Sequest. Flexible methods that allow incorporation of these new scoring measures are necessary for current proteomics research. All of the three statistical learning methods discussed in my study can be applied to incorporate newer search measures. They can not only combine sub-scores from one search engine but also scores from different search engines during the statistical evaluation of the quality of the matches.

Using logistic regression I was able to interpret outcome probabilities and correlation coefficients for the predictor variables. Since there were usually adequate observations in proteomics studies, logistic regression model with splines and interactions had better performance than less complex models that I tested. The inferences of features and scores derived from logistic regression will be useful for

understanding the mechanism of peptide-spectrum matches. Interactions exist between missCleavages or charge with other predictors. The new features such as retention time difference (*RTdiff*) found important by logistic regression in this study can be integrated in current classification algorithms to improve spectrum classifications. To understand the effect of each variable or to select variables for prediction in spectral classification, I recommend logistic regression.

Compared to support vector machines with Gaussian kernels or random forest, logistic regression may not be as flexible – there is always a trade-off between performance and interpretability. If the purpose of a study is to understand the importance and effects of certain predictors (e.g. retention time difference, a certain score of interest), logistic regression is the method of choice. Compared to random forests, it gave a better and more reasonable estimation of importance of predictors. However, if the purpose is to build an accurate model for spectrum classification, random forest is preferred.

SVM with a non-linear kernel function has high accuracy even when the data are not linearly separable in the base feature space. However, it is memory intensive and time-consuming to tune its parameters. Most the proteomics datasets are very large with thousands of thousands of spectra, therefore, although Percolator is widely used, from my testing and validation results, SVM is not recommended compared to the two other methods (Table 12).

Random forests outperform other methods in analyzing proteomics data, perhaps because proteomics datasets usually contain millions of spectra but only a few predictors. We need a classifier which is very flexible, easy to construct, and accurate. Random forests are non-parametric and can easily handle feature interactions. It is robust to outliers, as well as being fast and scalable. It can deal with classification problems of unbalanced, multiclass, and small sample data without data preprocessing procedures. In this study, compared to other methods, random forests was robust, easy to tune, and has the best performance in PSM classification (Table 12). Overall, I recommend random forests since it

has the highest accuracy while the interpretability is not of concern in the study. It has a great potential to be widely applied in shotgun proteomics analysis.

I stress that the methods used in this study are not restricted to any mass spectrometry instrument. However, improved results could be obtained by using datasets from particular instruments, e.g. for Orbitrap spectrum classification, better models can be obtained by training on Orbitrap datasets than training on datasets from a different instrument. In the future, it remains to be seen that if the recommended method can perform well across the rich variety of experimental datasets. It will also be valuable to see whether the protein identification tools such as ProteinProphet, which could naturally fit into this recommended method, could perform robustly in finding the true proteins when applied to various experimental datasets.

Table 12. Advantages and disadvantages of three methods found when applying to proteomics data

	Advantage	Disadvantages
Logistic regression	Interpretable Fast Scalable	Main effect model is not as flexible
Support vector machines	High accuracy Optimal solution Flexible with non-linear kernels	Memory-intensive, not suitable for large datasets Hard to interpret Tuning its parameters takes a lot of effort
Random forests	Robust to outliers Fast Scalable Easy to tune	Hard to interpret

APPENDIX

Supplementary Table 1. Summary statistics of predictor variables by correctness of spectra

	FALSE $N = 5740$	TRUE $N = 11355$	Test Statistic
MVH	-1.1222 -0.7672 -0.2454	-0.4574 0.1884 0.9110	$F_{1,17093} = 4374,\ P < 0.001^1$
mzFidelity	-1.1231 -0.6611 -0.1162	-0.4435 0.2005 0.8601	$F_{1,17093} = 3207,\ P < 0.001^1$
xcorr	-1.19313 -0.74563 -0.09645	-0.42528 0.22466 0.91008	$F_{1,17093} = 3746,\ P < 0.001^1$
unmatchedPeaks	-0.57271 0.05638 0.89516	-0.78241 -0.36302 0.26607	$F_{1,17093} = 800.3,\ P < 0.001^1$
massError	-0.25246 0.03605 0.28847	-0.06094 0.03712 0.08952	$F_{1,17093} = 16.23,\ P < 0.001^1$
as.factor(charge.cat)			$\chi^2_3 = 10.01,\ P = 0.019^2$
1	10% (586)	10% (1099)	
2	69% (3942)	69% (7849)	
3	17% (1004)	18% (2083)	
4	4% (208)	3% (324)	
enzN	83% (4761)	88% (10000)	$\chi^2_1 = 84.87,\ P < 0.001^2$
enzC	95% (5456)	96% (10878)	$\chi^2_1 = 5,\ P = 0.025^2$
missCleavages			$\chi^2_3 = 3632,\ P < 0.001^2$
0	55% (3146)	93% (10510)	
1	30% (1719)	7% (816)	
2	15% (865)	0% (29)	
3	0% (10)	0% (0)	
monoMass	-0.74422 -0.27648 0.41659	-0.66892 -0.05117 0.40810	$F_{1,17093} = 93.12,\ P < 0.001^1$
RTdiff	-0.50533 0.06665 0.94612	-0.47876 -0.19737 0.11585	$F_{1,17093} = 461,\ P < 0.001^1$
ModNumber			$\chi^2_8 = 287.8,\ P < 0.001^2$
0	69% (3981)	75% (8478)	
1	25% (1443)	16% (1802)	
2	4% (248)	7% (795)	
3	1% (53)	2% (200)	
4	0% (8)	1% (80)	
5	0% (2)	0% (0)	
6	0% (3)	0% (0)	
7	0% (1)	0% (0)	
11	0% (1)	0% (0)	

$_a\ b\ _c$ represent the lower quartile a, the median b, and the upper quartile c for continuous variables. Numbers after percents are frequencies. Tests used: [1]Wilcoxon test; [2]Pearson test

Supplementary Table 2. Estimate and confidence interval of correlation coefficients of the logistic regression model with restricted cubic spline of predictors with 3 knots (modelReg.spline).

	Estimate	Std.Error	95% CI lb	95% CI ub
(Intercept)	2.19	0.43	1.35	3.03
rcs(MVH)MVH	0.38	0.24	-0.08	0.84
rcs(MVH)MVH'	4.88	2.26	0.45	9.32
rcs(MVH)MVH''	-17.63	6.17	-29.71	-5.54
rcs(MVH)MVH'''	20.79	5.85	9.33	32.26

rcs(mzFidelity)mzFidelity	-1.28	0.20	-1.66	-0.90
rcs(mzFidelity)mzFidelity'	5.95	1.29	3.43	8.47
rcs(mzFidelity)mzFidelity''	-14.34	4.68	-23.51	-5.17
rcs(mzFidelity)mzFidelity'''	4.42	5.65	-6.65	15.49
rcs(xcorr)xcorr	2.11	0.20	1.71	2.50
rcs(xcorr)xcorr'	-2.65	1.29	-5.17	-0.12
rcs(xcorr)xcorr''	2.83	4.15	-5.30	10.96
rcs(xcorr)xcorr'''	0.87	4.72	-8.38	10.12
rcs(unmatchedPeaks)unmatchedPeaks	-1.07	0.26	-1.58	-0.56
rcs(unmatchedPeaks)unmatchedPeaks'	14.60	10.58	-6.14	35.35
rcs(unmatchedPeaks)unmatchedPeaks''	-24.50	18.34	-60.45	11.45
rcs(unmatchedPeaks)unmatchedPeaks'''	11.60	9.91	-7.82	31.02
rcs(massError)massError	0.00	0.05	-0.10	0.11
rcs(massError)massError'	5.49	0.43	4.65	6.33
rcs(massError)massError''	-325.33	22.47	-369.36	-281.30
rcs(massError)massError'''	924.57	66.03	795.15	1053.99
enzN	1.26	0.07	1.13	1.39
enzC	0.70	0.11	0.49	0.91
missCleavages	-1.63	0.06	-1.73	-1.52
rcs(monoMass)monoMass	1.50	0.23	1.05	1.96
rcs(monoMass)monoMass'	-7.41	1.95	-11.24	-3.59
rcs(monoMass)monoMass''	19.91	6.26	7.63	32.18
rcs(monoMass)monoMass'''	-15.66	6.83	-29.05	-2.26
rcs(RTdiff)RTdiff	1.82	0.12	1.59	2.06
rcs(RTdiff)RTdiff'	-10.55	1.22	-12.94	-8.17
rcs(RTdiff)RTdiff''	37.85	10.18	17.90	57.80
rcs(RTdiff)RTdiff'''	-22.22	14.08	-49.81	5.38
ModNumber	0.16	0.04	0.10	0.23
(charge.cat)2	-1.09	0.09	-1.27	-0.91
(charge.cat)3	-0.61	0.13	-0.87	-0.36
(charge.cat)4	-1.09	0.21	-1.51	-0.68

Supplementary Table 3. Estimate and confidence interval of correlation coefficients of the logistic regression model with restricted cubic spline of predictors and interactions (*modelReg.spline.inter*).

	Estimate	Std.Error	95% CI lb	95% CI ub
(Intercept)	0.56	0.92	-1.24	2.37
(charge.cat)2	1.06	0.93	-0.77	2.89
(charge.cat)3	4.34	1.75	0.91	7.77
(charge.cat)4	55.98	39.78	-21.98	133.94
rcs(MVH)MVH	0.95	0.48	0.01	1.89

rcs(MVH)MVH'	1.42	8.46	-15.16	18.01
rcs(MVH)MVH''	-52.21	40.17	-130.94	26.51
rcs(MVH)MVH'''	377.92	218.19	-49.73	805.56
rcs(massError)massError	-0.10	0.05	-0.20	-0.01
rcs(massError)massError'	7.38	2.25	2.96	11.80
rcs(massError)massError''	-313.69	96.56	-502.96	-124.42
rcs(massError)massError'''	835.60	270.21	305.98	1365.22
rcs(RTdiff)RTdiff	-0.45	0.43	-1.30	0.40
rcs(RTdiff)RTdiff'	10.40	3.94	2.68	18.12
rcs(RTdiff)RTdiff''	-125.02	32.25	-188.23	-61.81
rcs(RTdiff)RTdiff'''	189.37	44.38	102.38	276.36
ModNumber	0.36	0.19	-0.01	0.73
enzN	1.83	0.18	1.48	2.19
enzC	0.17	0.35	-0.51	0.86
missCleavages	-1.75	0.06	-1.87	-1.63
rcs(mzFidelity)mzFidelity	-1.12	0.20	-1.51	-0.72
rcs(mzFidelity)mzFidelity'	5.82	1.33	3.20	8.43
rcs(mzFidelity)mzFidelity''	-13.38	4.88	-22.94	-3.81
rcs(mzFidelity)mzFidelity'''	2.05	5.93	-9.58	13.68
rcs(xcorr)xcorr	2.24	0.21	1.82	2.65
rcs(xcorr)xcorr'	-4.49	1.37	-7.17	-1.82
rcs(xcorr)xcorr''	11.23	4.43	2.54	19.91
rcs(xcorr)xcorr'''	-11.31	5.13	-21.38	-1.25
rcs(monoMass)monoMass	1.47	0.26	0.97	1.98
rcs(monoMass)monoMass'	-9.12	2.14	-13.31	-4.93
rcs(monoMass)monoMass''	27.33	6.88	13.83	40.82
rcs(monoMass)monoMass'''	-28.64	7.68	-43.70	-13.58
rcs(unmatchedPeaks)unmatchedPeaks	-1.08	0.27	-1.60	-0.56
rcs(unmatchedPeaks)unmatchedPeaks'	6.38	10.84	-14.87	27.63
rcs(unmatchedPeaks)unmatchedPeaks''	-9.36	18.86	-46.33	27.61
rcs(unmatchedPeaks)unmatchedPeaks'''	4.35	10.34	-15.91	24.61
(charge.cat)2:rcs(MVH)MVH	-0.63	0.53	-1.66	0.41
(charge.cat)3:rcs(MVH)MVH	0.28	1.59	-2.83	3.38
(charge.cat)4:rcs(MVH)MVH	40.04	51.77	-61.43	141.51
(charge.cat)2:rcs(MVH)MVH'	5.25	8.84	-12.08	22.58
(charge.cat)3:rcs(MVH)MVH'	0.28	14.25	-27.65	28.21
(charge.cat)4:rcs(MVH)MVH'	-248.96	305.86	-848.46	350.53
(charge.cat)2:rcs(MVH)MVH''	27.12	40.81	-52.86	107.11
(charge.cat)3:rcs(MVH)MVH''	41.48	49.00	-54.57	137.52
(charge.cat)4:rcs(MVH)MVH''	592.00	646.85	-675.84	1859.83
(charge.cat)2:rcs(MVH)MVH'''	-347.85	218.29	-775.70	80.00
(charge.cat)3:rcs(MVH)MVH'''	-357.00	219.40	-787.03	73.03
(charge.cat)4:rcs(MVH)MVH'''	-706.88	430.35	-1550.36	136.60

(charge.cat)2:rcs(massError)massError	0.03	0.11	-0.18	0.24
(charge.cat)3:rcs(massError)massError	2.11	0.40	1.33	2.89
(charge.cat)4:rcs(massError)massError	2.61	0.71	1.22	4.00
(charge.cat)2:rcs(massError)massError'	-1.61	2.31	-6.13	2.91
(charge.cat)3:rcs(massError)massError'	-10.12	2.83	-15.68	-4.57
(charge.cat)4:rcs(massError)massError'	-3.22	6.87	-16.70	10.25
(charge.cat)2:rcs(massError)massError''	-24.87	99.34	-219.58	169.84
(charge.cat)3:rcs(massError)massError''	241.95	120.34	6.09	477.80
(charge.cat)4:rcs(massError)massError''	-20.67	371.07	-747.98	706.63
(charge.cat)2:rcs(massError)massError'''	133.75	278.34	-411.79	679.30
(charge.cat)3:rcs(massError)massError'''	-584.92	340.50	-1252.29	82.46
(charge.cat)4:rcs(massError)massError'''	56.27	1099.45	-2098.66	2211.20
(charge.cat)2:rcs(RTdiff)RTdiff	2.20	0.46	1.30	3.09
(charge.cat)3:rcs(RTdiff)RTdiff	3.72	0.58	2.58	4.85
(charge.cat)4:rcs(RTdiff)RTdiff	31.96	10.29	11.79	52.13
(charge.cat)2:rcs(RTdiff)RTdiff'	-24.19	4.19	-32.41	-15.98
(charge.cat)3:rcs(RTdiff)RTdiff'	-30.69	5.47	-41.42	-19.97
(charge.cat)4:rcs(RTdiff)RTdiff'	-192.81	60.56	-311.51	-74.11
(charge.cat)2:rcs(RTdiff)RTdiff''	201.92	34.40	134.50	269.35
(charge.cat)3:rcs(RTdiff)RTdiff''	243.12	45.45	154.03	332.21
(charge.cat)4:rcs(RTdiff)RTdiff''	1179.44	353.32	486.92	1871.96
(charge.cat)2:rcs(RTdiff)RTdiff'''	-269.23	47.37	-362.08	-176.38
(charge.cat)3:rcs(RTdiff)RTdiff'''	-333.81	63.28	-457.85	-209.78
(charge.cat)4:rcs(RTdiff)RTdiff'''	-1397.98	403.60	-2189.03	-606.94
(charge.cat)2:ModNumber	-0.10	0.19	-0.48	0.29
(charge.cat)3:ModNumber	-0.21	0.21	-0.62	0.19
(charge.cat)4:ModNumber	-0.46	0.38	-1.21	0.29
(charge.cat)2:enzN	-0.78	0.20	-1.17	-0.40
(charge.cat)3:enzN	0.32	0.29	-0.25	0.90
(charge.cat)4:enzN	1.66	1.27	-0.84	4.15
(charge.cat)2:enzC	0.50	0.37	-0.22	1.23
(charge.cat)3:enzC	0.08	0.54	-0.97	1.13
(charge.cat)4:enzC	1.85	1.56	-1.21	4.91

Supplementary Table 4. Estimate and confidence interval of correlation coefficients of the logistic regression model with linear main effect and interactions (*modelReg.inter*).

	Estimate	Std.Error	95% CI lb	95% CI ub
(Intercept)	-0.21	0.35	-0.90	0.47
(charge.cat)2	0.30	0.37	-0.43	1.03
(charge.cat)3	-0.78	0.51	-1.78	0.22

(charge.cat)4	-5.35	1.06	-7.42	-3.28
MVH	1.10	0.18	0.75	1.46
massError	-0.04	0.06	-0.15	0.07
enzN	2.20	0.16	1.89	2.51
enzC	1.03	0.34	0.37	1.69
RTdiff	-0.59	0.09	-0.77	-0.40
ModNumber	0.04	0.17	-0.30	0.38
xcorr	1.09	0.05	1.00	1.19
missCleavages	-1.72	0.05	-1.82	-1.62
mzFidelity	-0.08	0.08	-0.23	0.08
monoMass	-0.29	0.09	-0.47	-0.11
unmatchedPeaks	-0.30	0.08	-0.45	-0.14
(charge.cat)2:MVH	-0.56	0.18	-0.92	-0.21
(charge.cat)3:MVH	0.17	0.21	-0.23	0.58
(charge.cat)4:MVH	0.16	0.26	-0.34	0.66
(charge.cat)2:massError	0.00	0.07	-0.13	0.13
(charge.cat)3:massError	-0.23	0.11	-0.44	-0.03
(charge.cat)4:massError	0.04	0.09	-0.13	0.22
(charge.cat)2:enzN	-1.30	0.17	-1.64	-0.96
(charge.cat)3:enzN	0.29	0.25	-0.19	0.77
(charge.cat)4:enzN	1.10	0.59	-0.06	2.26
(charge.cat)2:enzC	-0.70	0.35	-1.40	-0.01
(charge.cat)3:enzC	-0.45	0.46	-1.35	0.45
(charge.cat)4:enzC	2.77	0.78	1.24	4.29
(charge.cat)2:RTdiff	0.61	0.10	0.42	0.80
(charge.cat)3:RTdiff	0.62	0.11	0.41	0.83
(charge.cat)4:RTdiff	0.00	0.15	-0.30	0.29
(charge.cat)2:ModNumber	0.15	0.18	-0.20	0.50
(charge.cat)3:ModNumber	0.09	0.18	-0.27	0.46
(charge.cat)4:ModNumber	-0.26	0.21	-0.68	0.16

REFERENCES

1. Allmer, J., *Existing bioinformatics tools for the quantitation of post-translational modifications.* Amino Acids, 2012. **42**(1): p. 129-38.

2. Han, X., A. Aslanian, and J.R. Yates, 3rd, *Mass spectrometry for proteomics.* Curr Opin Chem Biol, 2008. **12**(5): p. 483-90.

3. Witze, E.S., et al., *Mapping protein post-translational modifications with mass spectrometry.* Nat Methods, 2007. **4**(10): p. 798-806.

4. Lin, D., D.L. Tabb, and J.R. Yates, 3rd, *Large-scale protein identification using mass spectrometry.* Biochim Biophys Acta, 2003. **1646**(1-2): p. 1-10.

5. McConnell, R.E., et al., *Proteomic analysis of the enterocyte brush border.* Am J Physiol Gastrointest Liver Physiol, 2011.

6. McFarland, M.A., et al., *Proteomics analysis identifies phosphorylation-dependent alpha-synuclein protein interactions.* Mol Cell Proteomics, 2008. **7**(11): p. 2123-37.

7. Schrimpf, S.P., et al., *Comparative functional analysis of the Caenorhabditis elegans and Drosophila melanogaster proteomes.* PLoS Biol, 2009. **7**(3): p. e48.

8. Aebersold, R. and M. Mann, *Mass spectrometry-based proteomics.* Nature, 2003. **422**(6928): p. 198-207.

9. Choi, H., D. Ghosh, and A.I. Nesvizhskii, *Statistical validation of peptide identifications in large-scale proteomics using the target-decoy database search strategy and flexible mixture modeling.* J Proteome Res, 2008. **7**(1): p. 286-92.

10. Tabb, D.L., C.G. Fernando, and M.C. Chambers, *MyriMatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis.* J Proteome Res, 2007. **6**(2): p. 654-61.

11. Perkins, D.N., et al., *Probability-based protein identification by searching sequence databases using mass spectrometry data.* Electrophoresis, 1999. **20**(18): p. 3551-67.

12. Alves, G., et al., *Enhancing peptide identification confidence by combining search methods.* J Proteome Res, 2008. **7**(8): p. 3102-13.

13. Bakalarski, C.E., et al., *The effects of mass accuracy, data acquisition speed, and search algorithm choice on peptide identification rates in phosphoproteomics.* Anal Bioanal Chem, 2007. **389**(5): p. 1409-19.

14. Elias, J.E. and S.P. Gygi, *Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry.* Nat Methods, 2007. **4**(3): p. 207-14.

15. Jaffe, J.D., et al., *PEPPeR, a platform for experimental proteomic pattern recognition.* Mol Cell Proteomics, 2006. **5**(10): p. 1927-41.

16. Li, J., et al., *A bioinformatics workflow for variant peptide detection in shotgun proteomics.* Mol Cell Proteomics, 2011.

17. Gerster, S., et al., *Protein and gene model inference based on statistical modeling in k-partite graphs.* Proc Natl Acad Sci U S A, 2010. **107**(27): p. 12101-6.

18. Karpievitch, Y., et al., *A statistical framework for protein quantitation in bottom-up MS-based proteomics.* Bioinformatics, 2009. **25**(16): p. 2028-34.

19. Li, Y.F., et al., *A bayesian approach to protein inference problem in shotgun proteomics.* J Comput Biol, 2009. **16**(8): p. 1183-93.

20. Nesvizhskii, A.I. and R. Aebersold, *Interpretation of shotgun proteomic data: the protein inference problem.* Mol Cell Proteomics, 2005. **4**(10): p. 1419-40.

21. Keller, A., et al., *Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search.* Anal Chem, 2002. **74**(20): p. 5383-92.

22. Ma, Z.Q., et al., *IDPicker 2.0: Improved protein assembly with high discrimination peptide identification filtering.* J Proteome Res, 2009. **8**(8): p. 3872-81.

23. Zhang, B., M.C. Chambers, and D.L. Tabb, *Proteomic parsimony through bipartite graph analysis improves accuracy and transparency.* J Proteome Res, 2007. **6**(9): p. 3549-57.

24. Kall, L., et al., *Semi-supervised learning for peptide identification from shotgun proteomics datasets.* Nat Methods, 2007. **4**(11): p. 923-5.

25. Spivak, M., et al., *Improvements to the percolator algorithm for Peptide identification from shotgun proteomics data sets.* J Proteome Res, 2009. **8**(7): p. 3737-45.

26. Anderson, D.C., Li, W., Payan, D. G., and Noble, W. S., *A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores.* J. Proteome Res., 2003. **2**: p. 137–146.

27. Peter J. Ulintz, J.Z., Zhaohui S. Qin, Phillip C. Andrews, *Improved classification of mass spectrometry ddatabase search results using new machine learning approaches.* Molecular cellular proteomics, 2006.

28. Cox., D.R., *The regression analysis of binary sequences (with discussion).* J Roy Stat Soc B, 1958: p. 20:215-242

29. S. H. Walker, D.B.D., *Estimation of the probability of an event as a function of several independent variables.* Biometrika, 1967. **54:167-178**.

30. Breiman, L., *Machine Learning*. Bagging Predictors. Vol. 24(2). 1996.

31. F. E. Harrell, K.L.L., R. M. Cali, D. B. Pryor, R. A. Rosati., *Regression modeling strategies for improved prognostic prediction.* Stat Med, 1984. **3:143-152**.

32. Harrell, F.E. *R package: Hmics*. 2014; Available from: http://cran.r-project.org/web/packages/Hmisc/index.html.

33. James, G., Witten, D., Hastie, T., Tibshirani, R., *An introduction to statistical learning*. 2013.

34. Breiman, L., *Random Forests*, in *Machine Learning*. 2001. p. 5-32.

35. *Pruning decision tree*. Available from: http://en.wikipedia.org/wiki/Pruning_(decision_trees).

36. Varian, H.R., *Big data: new tricks for econometrics.* 2013.

37. Mansour, Y., *Pessimistic decision tree pruning based on tree size.* Proc. 14th International Conference on Machine learning, 1997: p. 195-201.

38. Yin-Wen Chang, C.-J.H., Kai-Wei Chang, *Training and Testing Low-degree Polynomial Data Mappings via Linear SVM.* Journal of Machine Learning Research, 2010.

39. Trevor Hastie, R.T., Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2009.

40. Harrell, F., *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. 2001.

41. Efron B, T.R., *Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy.* Statistical Sci 1986. **1:54-77**.

42. Dicker, L., X. Lin, and A.R. Ivanov, *Increased power for the analysis of label-free LC-MS/MS proteomics data by combining spectral counts and peptide peak attributes.* Mol Cell Proteomics, 2010. **9**(12): p. 2704-18.

43. Chen, Y.Y., et al., *IDPQuantify: Combining Precursor Intensity with Spectral Counts for Protein and Peptide Quantification.* J Proteome Res, 2013.

44. Kessner, D., et al., *ProteoWizard: open source software for rapid proteomics tools development.* Bioinformatics, 2008. **24**(21): p. 2534-6.

45. Li, M., et al., *Comparative shotgun proteomics using spectral count data and quasi-likelihood modeling.* J Proteome Res, 2010. **9**(8): p. 4295-305.

46. Krokhin, O.V., *Sequence-specific retention calculator. Algorithm for peptide retention prediction in ion-pair RP-HPLC: application to 300- and 100-A pore size C18 sorbents.* Anal Chem, 2006. **78**(22): p. 7785-95.

47. Smith., P.L., *Splines as a useful and convenient statistical tool.* Am Statistician, 1979. **33:57-62**.

48. Xuming He, L.S., *Linear regression after spline transformation* Biometrika, 1997. **84:474-481**.

49. Harrell, F. *R package: rms*. 2014.

50. Dasari, S., et al., *Sequence tagging reveals unexpected modifications in toxicoproteomics.* Chem Res Toxicol, 2011. **24**(2): p. 204-16.

51. Dasari, S., et al., *Pepitome: evaluating improved spectral library search for identification complementarity and quality assessment.* J Proteome Res, 2012. **11**(3): p. 1686-95.

52. Dasari, S., et al., *TagRecon: high-throughput mutation identification through sequence tagging.* J Proteome Res, 2010. **9**(4): p. 1716-26.

53. Eng, J.K., A.L. Mccormack, and J.R. Yates, *An Approach to Correlate Tandem Mass-Spectral Data of Peptides with Amino-Acid-Sequences in a Protein Database.* Journal of the American Society for Mass Spectrometry, 1994. **5**(11): p. 976-989.

54. http://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html

55. Houtao Deng, G.R., *Feature Selection via Regularized Trees.* 2012.

56. Breiman, L.C., A. *Random Forests*. Available from: www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#varimp.

57. Deng, H.R., G.; Tuv, E., *Bias of importance measures for multi-valued attributes and solutions.* Proceedings of the 21st International Conference on Artificial Neural Networks, 2011: p. 293-300.