Integrated data management and validation platform for phosphorylated tandem mass spectrometry data: Supplementary Material

This document contains details of the five classification methods, two supplementary figures and one supplementary table.

Classification methods

To test the importance of our features and to achieve good separation between correct and incorrect peptide assignments, we used five classifiers available in the Weka machine learning workbench [1]. Here we briefly describe the classifiers used in the case study. The selected parameters for the classifiers were default values used by Weka.

The classifiers were compared with an independent dataset and cross-validation. The cross-validation is done so that the whole training data is randomly split into ten subsets. One subset at a time is left out and a classification method is trained with the existing data. Then the trained classifier is used to predict the remaining subset of data and its performance is recorded. The next round leaves out the second subset of data and the classifier is trained with the remaining nine data sets and the trained classifier is once again used to classify the dataset that was left out. This process is done with all subsets of the data and the accuracy is the mean of the individual prediction accuracies.

Logistic regression

In the logistic regression data are assumed binomially distributed and the logistic function $f(z) = 1/(1+\exp(-z))$ is used to predict the classes. The variable z is a measure of the contributions of the features: $z = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$, where β_0 is a constant term and $\beta_1, \beta_2, \ldots, \beta_n$ are regression coefficients. The implementation of logistic regression in Weka uses a multinomial logistic regression model with a ridge estimator that includes a ridge parameter to penalize for large parameters in the model [2]. We set ridge parameter to 10^{-8} and number of iterations unlimited. The method uses z-normalization of data.

Decision tree

A decision tree uses consecutive decisions or splits so that each split utilizes the feature that maximizes

the purity of the split [3]. The leaves of a decision tree correspond to classes. The algorithm implemented in Weka is called J48, which is a version of C4.5 [4]. The following parameters were chosen: pruning confidence was set to 0.25 and minimum number of instances for a leaf was two. Data normalization is not required.

Random forest

The random forest classifier uses an ensemble of decision trees with out-of-bag sampling where each tree is built using a bootstrap sample [5]. New samples are classified using the ensemble of individual decision tree classifiers and the final label is assigned via majority vote. The features used for each decision tree are selected at random from the available features, thus in theory discarding unimportant features and using only the most informative features for classification. Data normalization is not required.

The random forest is also able to report feature importance to classification. For one feature at a time, the values are permutated and classification is performed using the feature with permutated values together with the other (not permutated) features. The number of votes for the correct class with perturbed values is subtracted from the number of votes for correct class with unperturbed, original data to obtain the average decrease in accuracy for each variable. The decrease in accuracy of the classifier for perturbation of a feature correlates to the importance of that feature to the overall classifier. The parameters for the random forest classifier were: The number of trees to build was set to 500, number of features to consider was set to zero, and random number generator seed was 1.0.

Artificial neural network

The artificial neural network (ANN) predictors mimic biological neural networks. An ANN consists of neurons or nodes that are arranged to input, hidden and output layers. Each node typically has a nonlinear activation function. Nodes are connected with weighted arcs; weights are tuned during the learning step [6]. As we used the Weka implementation for ANN, activation functions were sigmoidal and the learning algorithm was back-propagation. The parameters were as follows: learning rate was 0.3, momentum rate was 0.2, number of training epochs was 500, validation set size was zero, seed for the random number generator was 0.0, number of allowed consecutive errors for validation testing before terminating the network was 20, and number of hidden nodes was *(attributes+classes)/2*. The normalization method used by Weka was linear normalization.

Naïve Bayes classifier

Naïve Bayes classifier uses Bayes' theorem and the assumption that the features used in the classification are independent of each other given the class variable [7]. The Naïve Bayes classifier calculates the posterior probabilities for the classes given the data, and the outcome is the most probable class. In our case study a kernel estimator was used for numeric attributes rather than assuming a normal distribution and data was linearly normalized.

References

[1] Witten, I. H., Frank, E., Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco 2005, 2nd edition.

[2] le Cessie, S., van Houwelingen, J., Ridge estimators in logistic regression. *Applied Statistics* 1992, *41*, 191-201.

[3] Quinlan, R., Induction of decision trees. *Machine Learning* 1986, *1*, 81-106.

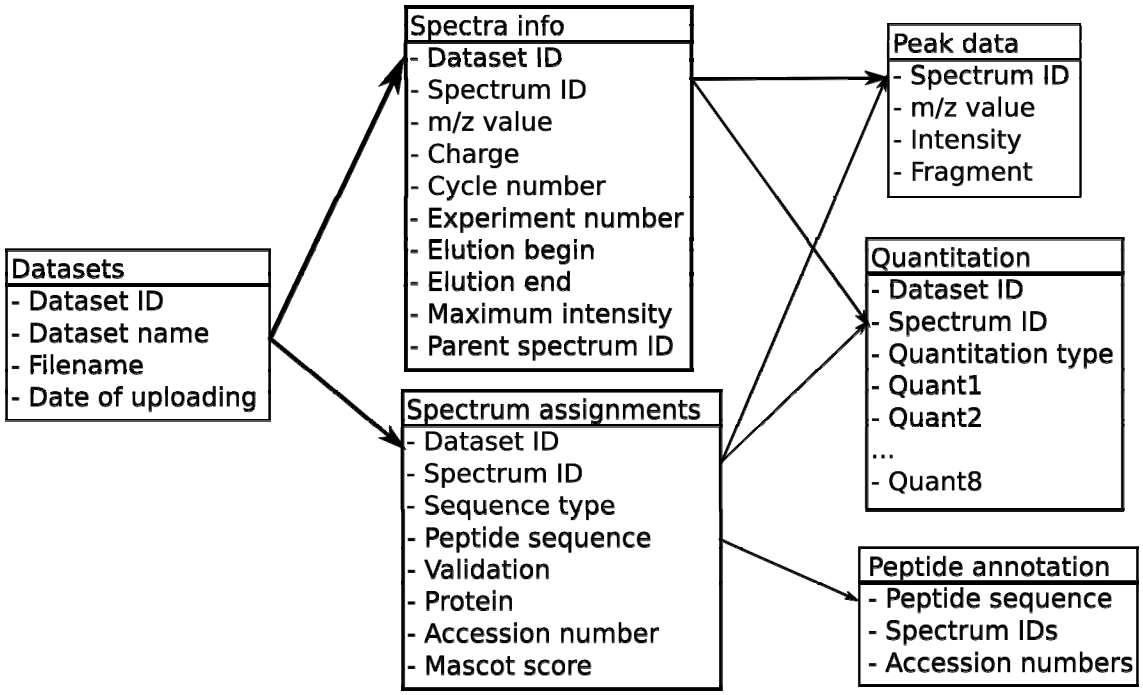
[4] Quinlan, R., C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA 1993.

[5] Breiman, L., Random forests. *Machine Learning* 2001, *45*, 5–32.

[6] Haykin, S., Neural Networks: A Comprehensive Foundation. Macmillan, New York 1994.

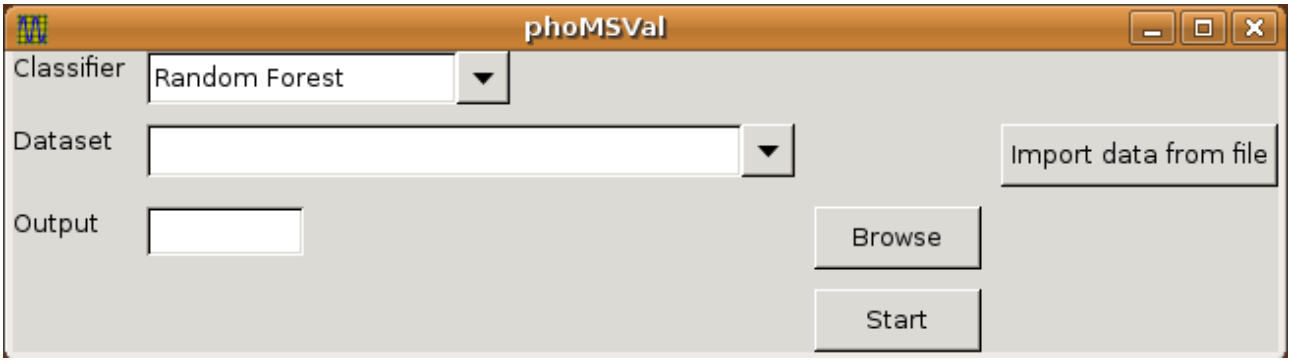
[7] John, G. H. and Langley, P., Estimating continuous distributions in Bayesian classifiers. In Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo 1995, pages 338–345.

Supplementary Figure S1



Tables and dependencies for the MySQL database format used with phoMSVal. The database consists of six tables. The "Datasets" table contains identifiers for each dataset entered into the database. The "Spectra Info" table includes additional information about each spectrum. The "Spectrum Assignments" table contains information regarding the peptide assignment for each spectrum. This table also contains manual validation information for the given peptide assignment, if such is available. There are four validation classes: "1" means that the spectrum was correctly assigned, "-1" means incorrect assignment, "2" means that the manual assessment was inconclusive, and "0" that the spectrum has not been validated. The "Quantitation" table contains quantitation information for each spectrum, if applicable. The "Peak Data" table holds all of the peak list data (e.g. *m/z*, intensity) for each MS/MS spectrum. If a peak has been assigned as a particular fragment ion, they are also included here. The sixth table is a “Peptide Annotation” table that can be used for possible storing more informative peptide to protein annotations. Spectrum IDS and protein accession numbers can be stored as concatenated lists.

Supplementary Figure S2



Screenshot of the graphical user interface used with phoMSVal. The user can select with classifier to use, which dataset to classify, whether to import new data from a file, and where to write the resulting output file.

Supplementary Table S1. Correlation coefficients for the set of 17 features. Features in red were retained for the analysis.

	n	N	sigma	I_{max}	mz_{Imax}	mz_{max}	I_{tot}	I_{avg}	I_{bal}	S	N_{noID}	I_{blon}	I_{ylon}	I_{noID}	N_{NL}	I_{NL}	NoID_{NL}
n	1.000	0.999	0.730	0.794	0.125	0.451	0.744	0.693	-0.651	0.243	0.224	0.631	0.615	0.691	0.562	0.548	0.139
N	0.999	1.000	0.729	0.792	0.123	0.447	0.746	0.694	-0.655	0.244	0.224	0.632	0.616	0.693	0.563	0.550	0.139
sigma	0.730	0.729	1.000	0.968	0.052	0.225	0.903	0.982	-0.667	0.212	0.142	0.871	0.896	0.948	0.402	0.813	0.160
I_{max}	0.794	0.792	0.968	1.000	0.063	0.224	0.950	0.928	-0.749	0.188	0.169	0.862	0.875	0.912	0.401	0.799	0.134
mz_{Imax}	0.125	0.123	0.052	0.063	1.000	0.107	0.050	0.028	-0.033	0.123	-0.162	0.074	0.030	0.001	0.131	0.033	0.126
mz_{max}	0.451	0.447	0.225	0.224	0.107	1.000	0.173	0.203	-0.147	0.365	-0.054	0.174	0.155	0.185	0.386	0.101	0.017
I_{tot}	0.744	0.746	0.903	0.950	0.050	0.173	1.000	0.880	-0.849	0.153	0.160	0.866	0.860	0.894	0.355	0.819	0.081
I_{avg}	0.693	0.694	0.982	0.928	0.028	0.203	0.880	1.000	-0.655	0.214	0.136	0.857	0.859	0.977	0.419	0.796	0.163
I_{bal}	-0.651	-0.655	-0.667	-0.749	-0.033	-0.147	-0.849	-0.655	1.000	-0.098	-0.144	-0.694	-0.609	-0.688	-0.294	-0.637	-0.049
S	0.243	0.244	0.212	0.188	0.123	0.365	0.153	0.214	-0.098	1.000	-0.341	0.205	0.207	0.152	0.367	0.115	0.187
N_{noID}	0.224	0.224	0.142	0.169	-0.162	-0.054	0.160	0.136	-0.144	-0.341	1.000	0.088	0.126	0.209	-0.118	0.111	-0.303
I_{blon}	0.631	0.632	0.871	0.862	0.074	0.174	0.866	0.857	-0.694	0.205	0.088	1.000	0.860	0.817	0.332	0.890	0.155
I_{ylon}	0.615	0.616	0.896	0.875	0.030	0.155	0.860	0.859	-0.609	0.207	0.126	0.860	1.000	0.818	0.224	0.892	0.128
I_{noID}	0.691	0.693	0.948	0.912	0.001	0.185	0.894	0.977	-0.688	0.152	0.209	0.817	0.818	1.000	0.404	0.776	0.091
N_{NL}	0.562	0.563	0.402	0.401	0.131	0.386	0.355	0.419	-0.294	0.367	-0.118	0.332	0.224	0.404	1.000	0.205	0.424
I_{NL}	0.548	0.550	0.813	0.799	0.033	0.101	0.819	0.796	-0.637	0.115	0.111	0.890	0.892	0.776	0.205	1.000	0.201
NoID_{NL}	0.139	0.139	0.160	0.134	0.126	0.017	0.081	0.163	-0.049	0.187	-0.303	0.155	0.128	0.091	0.424	0.201	1.000