A Symbolic Regression Based Scoring System Improving Peptide Identification for MS Amanda

Viktoria Dorfer*
University of Applied Sciences
Upper Austria
Bioinformatics; Heuristic and
Evolutionary Algorithms
Softwarepark 13
4232 Hagenberg, Austria
viktoria.dorfer
@fh-hagenberg.at

Sergey Maltsev*
Research Institute of
Molecular Pathology
Protein Chemistry Facility
Dr. Bohr-Gasse 3
1030 Vienna, Austria
sergey.maltsev@imp.ac.at

Stephan Dreiseitl
University of Applied Sciences
Upper Austria
Bioinformatics; Heuristic and
Evolutionary Algorithms
Softwarepark 11
4232 Hagenberg, Austria
stephan.dreiseitl
@fh-hagenberg.at

Karl Mechtler
Research Institute of
Molecular Pathology;
Institute of Molecular
Biotechnology of the Austrian
Academy of Sciences
Protein Chemistry Facility
Dr. Bohr-Gasse 3
1030 Vienna, Austria
karl.mechtler@imp.ac.at

Stephan M. Winkler†
University of Applied Sciences
Upper Austria
Bioinformatics; Heuristic and
Evolutionary Algorithms
Softwarepark 11
4232 Hagenberg, Austria
stephan.winkler
@fh-hagenberg.at

ABSTRACT

Peptide search engines are algorithms that are able to identify peptides (i.e., short proteins or parts of proteins) from mass spectra of biological samples. These identification algorithms report the best matching peptide for a given spectrum and a score that represents the quality of the match; usually, the higher this score, the higher is the reliability of the respective match. In order to estimate the specificity and sensitivity of search engines, sets of target sequences are given to the identification algorithm as well as so-called decoy sequences that are randomly created or scrambled versions of real sequences; decoy sequences should be assigned low scores whereas target sequences should be assigned high scores.

In this paper we present an approach based on symbolic regression (using genetic programming) that helps to distinguish between target and decoy matches. On the basis of features calculated for matched sequences and using the information on the original sequence set (target or decoy) we learn mathematical models that calculate updated scores. As an alternative to this white box modeling approach we also use a black box modeling method, namely random forests.

As we show in the empirical section of this paper, this approach leads to scores that increase the number of reliably identified samples that are originally scored using the MS Amanda identification algorithm for high resolution as well as for low resolution mass spectra.

Keywords

Proteomics; peptide identification; symbolic regression

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining; I.2.8 [**Artificial Intelligence**]: Heuristic methods; J.3 [**Life and Medical Sciences**]: Biology and genetics

1. INTRODUCTION

Mass spectrometry based proteomics has emerged to a powerful and widely used technique in the analysis of biological samples [2]. Obtained so-called tandem mass spectra contain peaks as mass-to-charge ratios and respective ion intensities of peptide fragments. Peptide search engines are used to identify peptides (i.e., short proteins or parts of proteins) from those mass spectra. These identification algorithms report the best matching peptide for a given spectrum and a score that represents the quality of the match. A score dependent on an identification algorithm is assigned to each peptide spectrum match (PSM); usually, the higher this score, the higher is the reliability of the respective match. There are several scoring algorithms that are frequently used in modern proteomics incorporating various strategies to evaluate the quality of a PSM, e.g., Mascot

*V. Dorfer and S. Maltsev are equally contributing authors
†S. Winkler serves as corresponding author

GECCO '15, July 11 - 15, 2015, Madrid, Spain
 ISBN 978-1-4503-3488-4/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739482.2768509

[14], SEQUEST [8], Andromeda [5], and, most recently, MS Amanda [6].

In order to estimate the specificity and sensitivity of search engines, sets of target sequences are given to the identification algorithm as well as so-called decoy sequences that are randomly created or scrambled versions of real sequences. As no gold standard data are available for proteomics experiments, these target-decoy searches are used to estimate false identifications among matches to the target database. [13, 7] In practice, a threshold θ is defined and only PSMs with a score higher than this threshold are accepted. θ is set to that certain value leaving only a desired number of decoy matches above the threshold. Applying this false discovery rate (FDR), the number of false identifications can be estimated as being equal to the number of decoy hits and is usually set to values such as, e.g., 1%.

Appropriate peptide identification algorithms should assign low scores to false and decoy sequences whereas target sequences should be assigned high scores. Obviously, these approaches are not always working perfectly - there will always be true PSMs that are scored below θ. To improve the discrimination between correct and wrong identifications we here present a machine learning approach for target-decoy classification. On the basis of features that are calculated for matched sequences and using the information about previously analyzed samples on the original sequence set (target or decoy) we learn mathematical models that calculate updated scores.

This approach is inspired by Percolator [9], a semi-supervised learning method for peptide identification from shotgun proteomics datasets. Percolator uses support vector machines to learn models that discriminate between positive and negative PSMs. Instead of support vector machines, we want to focus on white box modeling, namely symbolic regression by genetic programming for training such discriminators. White box models may further be used to improve score calculation of peptide identifications algorithms.

In Section 2 we define the algorithmic details of the approach pursued to reach these goals. As we show in the empirical section (Section 3) of this paper, this approach leads to scores that increase the number of reliably identified samples that are originally scored using the MS Amanda identification algorithm. In Section 4 we discuss these results and give an outlook to further research in this area.

2. ALGORITHMS

2.1 Overall Workfl w

The overall workflow of the algorithm described in this paper is shown in Figure 1. In an initial training phase we collect information about analyzed PSMs and train models that shall assign improved scores to PSMs; later, these models are used to calculate updated scores that shall help to distinguish more clearly between target and decoy peptide spectrum matches.

2.1.1 Training Phase

- First, in the training phase, standard peptide spectrum matching results are collected. For each given spectrum we get PSMs plus respective scores. Additionally, we also know which PSMs are decoy hits and which target hits are above 1% FDR and therefore considered true hits.

- From this information we calculate a **new score** for each PSM *psm*:

  - If *psm* is a decoy hit, then it is assigned 0:

$$is\_decoy(psm) \Leftrightarrow score_{new}(psm) = 0 \quad (1)$$

  - Otherwise, if *psm* is a true hit, then it is assigned the original score:

$$!is\_decoy(psm)\,\&\,score(psm) > \theta \Leftrightarrow score_{new}(psm) = score(psm) \quad (2)$$

  - All other PSMs matching the target database are not used for training as it is doubtful whether those are true or false hits.

- These new scores are then used in combination with further information on the PSMs, especially on peptide sequences, for training models that assign estimates for the new score to new, unseen PSMs. For all PSMs the following features are calculated:

  - Scores calculated by the peptide identification algorithm.
  - Mass spectrum specific features such as the mass to charge ratio and the charge state of the spectrum.
  - Peptide specific features such as the score difference to the second best matching peptide[1], the peptide length, or the number of missed cleavages.

2.1.2 Application Phase

- In the later application phase, new spectra are presented and, using a peptide search engine, PSMs are calculated. Additionally, using the previously generated mathematical models, an updated score is also calculated for each newly presented PSM.

- As usual, a threshold θ is set such that only a certain ratio of decoy hits are within considered PSMs estimating the number of false identifications among target hits. We can then calculate how many target PSMs can now be confidently identified using either the standard workflow or the new workflow with updated scores presented here.

[1]Usually, only the best matching peptide is considered for a spectrum, still the distance to the second best matching peptide is of high importance.

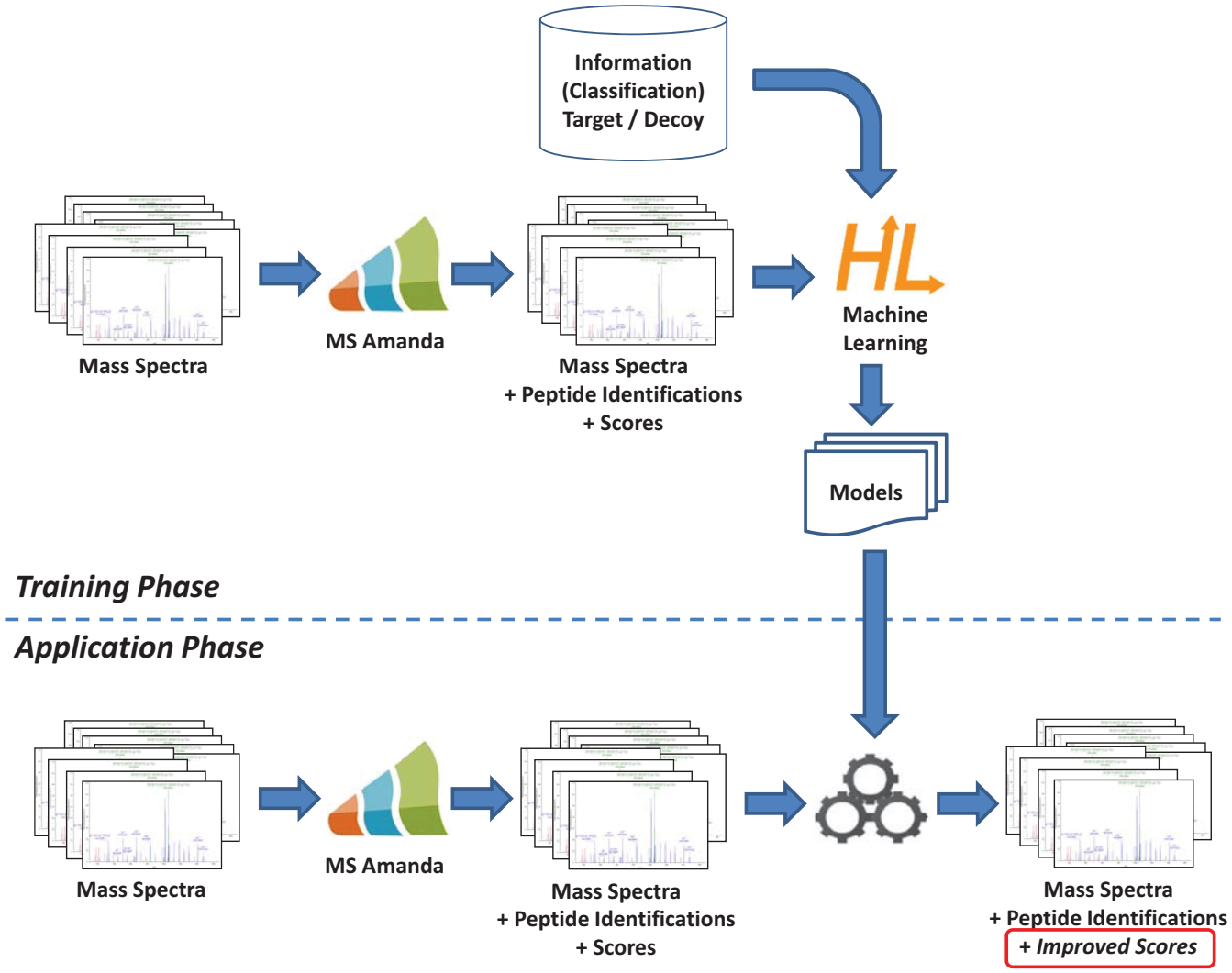

Figure 1: Overview of enhanced peptide identification using MS Amanda and machine learning.

2.2 Methods

In this section we describe the methods we use for identifying peptides and for calculating models that estimate new scoring values:

- As peptide identification algorithm we use MS Amanda.
- For machine learning we use genetic programming as well as random forests.

2.2.1 MS Amanda

To identify peptides out of mass spectra we used the database search algorithm MS Amanda [6]. MS Amanda is a scoring approach especially designed for mass spectra with high mass accuracy and outperforms gold standard peptide identification algorithms Mascot and SEQUEST at the same false discovery rate. This scoring algorithm is freely available at http://ms.imp.ac.at/?goto=msamanda as a platform independent standalone application as well as integrated in Proteome Discoverer, SearchGUI [18], and PeptideShaker [19].

2.2.2 Genetic Programming

For symbolic regression we use genetic programming (GP) [12] with strict offspring selection (OS) as described in [21] and [1]. The functions set described in [21] (including arithmetic as well as logical ones) was used for building composite function expressions.

Applying offspring selection has the effect that new individuals are compared to their parents; in the strict version, children are passed on to the next generation only if their quality is better than the quality of both parents. Figure 2 shows our GP implementation with OS, Figure 3 schematically shows OS (standard as well as strict).

In addition to splitting the given data into training and test data, we apply GP in such a way that a part of the given training data is not used for training models and serves as

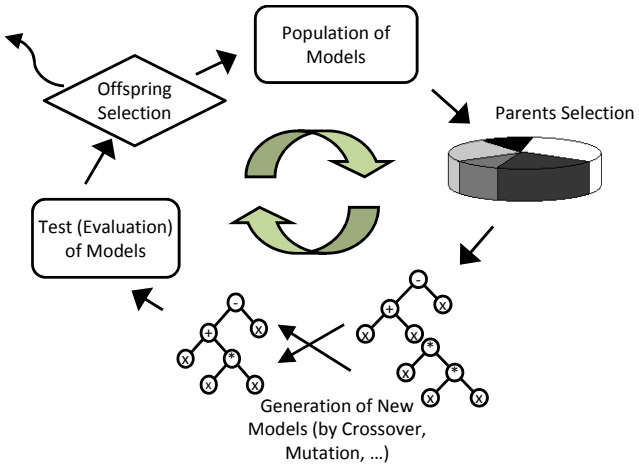

Figure 2: Genetic programming with offspring selection [21].

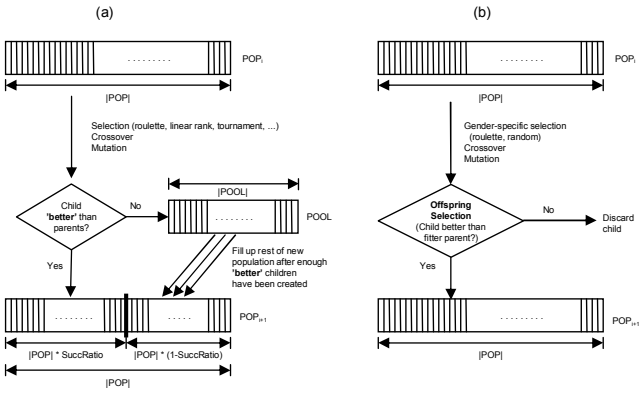

Figure 3: Offspring selection [1].

validation set; in the end, when it comes to returning the eventual results, the algorithm returns those models that perform best on validation data. This approach has been chosen because it helps to cope with over-fitting; it is also applied in other GP based machine learning algorithms as for example described in [3].

We use GP as implemented in HeuristicLab [20, 11] (http://dev.heuristiclab.com), a framework for prototyping and analyzing optimization techniques for which both generic concepts of evolutionary algorithms and many functions to evaluate and analyze them are available. Figure 4 shows GP solving a regression problem in HeuristicLab 3.3.11.

2.2.3 *Random Forest Classificatio*

Random forests (RFs, [4]) are ensembles of decision trees, each depending on randomly chosen samples and features. The best known algorithm for inducing random forests was first described in [4] combining bagging and random feature selection:

- For each tree in the forest, a certain number of input variables is used to determine the decision at a node of the tree.

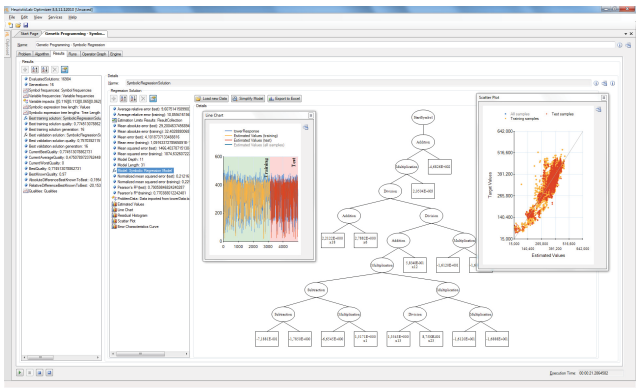

Figure 4: Solving a regression problem with symbolic regression in HeuristicLab 3.3.11.

- A certain number of samples is randomly drawn from the training data base; the rest of the samples is used as internal validation set for estimating the model's prediction error (out-of-bag error).

When it comes to calculating the value predicted for a given sample, this sample is pushed down the trees and is assigned the label (predicted value) of the terminal node it eventually ends up in. This procedure is executed for all trees in the forest and the final prediction for the given sample is the mode vote of all trees.

RFs are a very popular machine learning method as they are known to be one of the most accurate learning algorithms available [15], robust against overfitting, and widely considered a very efficient machine learning method.

Figure 5 schematically shows the aggregation of estimated target values produced by a set of trees as implemented for random forests.

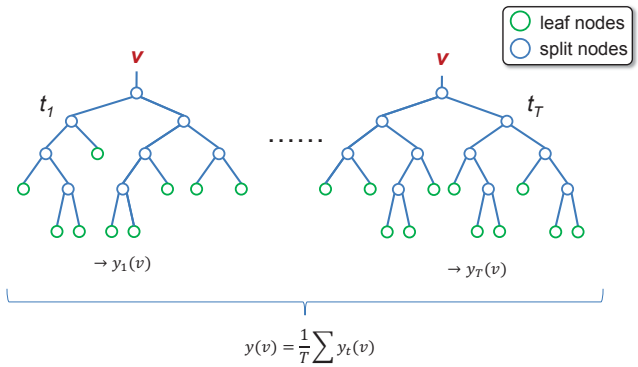

Figure 5: Random forest regression (adapted from [16]).

3. EMPIRICAL TESTS

3.1 Sample Preparation and Data

To test our approach we used two mass spectrometry data sets from a human cancer cell line:

- The first data set *DS1* (1 ug) was measured on a Thermo Fisher QExactive mass spectrometer and acquired along a 3h gradient (high resolution data set for MS2 spectra),
- the second data set *DS2* (1 ug, 1h) was acquired on a Thermo LTQ-Orbitrap Velos and first reported in Koecher et al. [10].

Resulting spectra where analyzed in Proteome Discoverer (version 1.4.0.288) using MS Amanda. Mass spectra were matched to the uniprot human protein database [17] including isoforms and extended for common contaminants and reverted protein sequences accounting for decoy proteins. Database search was conducted using trypsin as digestion enzyme and a 2 missed cleavages constraint. For the high resolution data set (*DS1*) 15 ppm and 0.02 Da were used as precursor mass and as fragment mass tolerance, respectively, while we used 10 ppm and 0.5 Da for the low resolution data set (*DS2*). Carbamidomethylation of cysteine and oxidation of methionine were set as fixed and variable peptide modifications, respectively. For each spectrum MS Amanda reported up to 5 best matching peptides, with an Amanda score ranging between 0.4 and 662 (662 representing a top match).

Each so obtained data set was split into one set of PSMs that are used for training models and one for testing our combined approach:

- For *DS1*, 30,000 samples (PSMs) are used for training and model selection, the remaining 155,271 samples (PSMs) are here used as test samples.
- For *DS2*, 2,000 samples (PSMs) are used for training and model selection, the remaining 35,163 samples (PSMs) are here used as test samples.

3.2 Test Results

Both machine learning methods applied here, GP and RFs, were executed with varying parameter settings:

- For GP different model size constraints and population sizes were applied: The allowed model depth was varied between 6 and 10, the allowed model complexity was varied between 50 and 200, and the mutation rate was varied between 10% and 30%. A combination of random and roulette parent selection was applied as well as offspring selection. Offspring was kept strict for all executions, i.e., in each generation only those models were propagated to the next generation that performed better than both parents. The maximum selection pressure was set to 100, and this was used as termination criterion. As fitness function we used Pearson's correlation coefficient (R^2).
- For RFs, different values for the parameters M (the ratio of features used for creating the trees), R (the ratio of samples used for training the trees), and the number of trees were tested: M was varied between 0.3 and 0.7, R was also varied between 0.3 and 0.7, and the number of trees was varied between 50 and 200.

For both methods, the 5 models with best performance on training (in the case of GP: validation) data were selected; the quality of a model is calculated as the correlation (R^2) of estimated and original scores. The test results given in the following are calculated as the average performance of the so selected models, where performance is calculated as PSMs on test data not seen by the identification algorithms.

We here analyze test results (i.e., reported PSMs) for different false discovery rates (FDR) as well as varying size limits for the peptides. Tables 1 and 2 summarize the results where column "ml" gives the minimum peptide length, "A" the results achieved using MS Amanda, "A+GP" the results achieved using the combination of MS Amanda and GP, and "A+RF" the combination of MS Amanda and RFs. Figures 6 and Figures 7 show these results graphically.

Table 1: Test results achieved for data set *DS1*.

FDR	ml	A	A+GP	A+RF
0.1 %	6	7586	8245	9326
			+8.7%	*+22.9%*
	7	7610	9218	9319
			+21.1%	*+22.5%*
	8	8566	9311	10615
			+8.7%	*+23.9%*
0.5 %	6	10350	12452	12956
			+20.3%	*+25.2%*
	7	10857	12953	13246
			+19.3%	*+22.0%*
	8	11094	13144	12641
			+18.5%	*+13.9%*
1 %	6	11976	14451	13837
			+20.7%	*+15.5%*
	7	12154	14407	13912
			+18.5%	*+14.5%*
	8	11892	13493	13390
			+13.5%	*+12.6%*

Table 2: Test results achieved for data set *DS2*.

FDR	ml	A	A+GP	A+RF
0.1 %	6	2708	3103	2788
			+14.6%	*+3.0%*
	7	2781	3191	2804
			+14.8%	*+0.8%*
	8	3467	3704	3612
			+6.8%	*+4.2%*
0.5 %	6	3663	4036	4004
			+10.2%	*+9.3%*
	7	3921	4201	4267
			+7.1%	*+8.8%*
	8	3954	4247	4217
			+7.4%	*+6.7%*
1 %	6	4101	4447	4438
			+8.4%	*+8.2%*
	7	4189	4521	4488
			+7.9%	*+7.1%*
	8	4153	4355	4322
			+4.9%	*+4.1%*

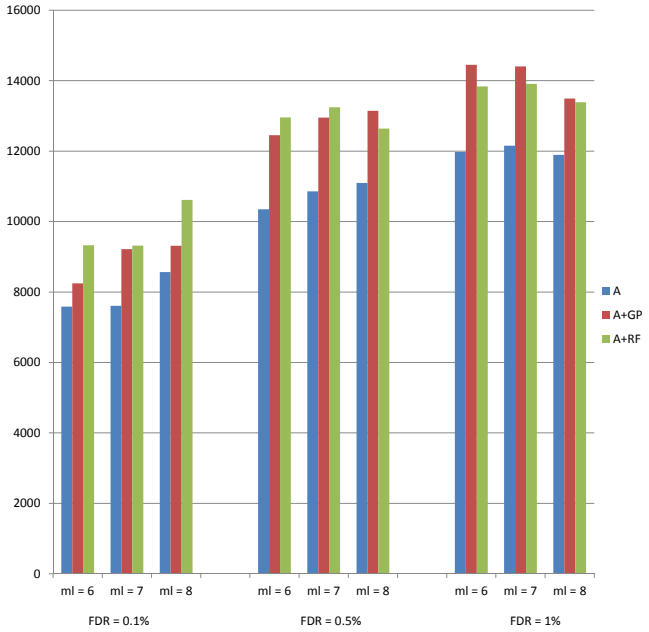

Figure 6: PSMs identified for the first data set (*DS1*) using standard MS Amanda (A, blue) compared to results obtained using new scores calculated with models generated by symbolic regression (A+GP, red) and random forests (A+RF, green).

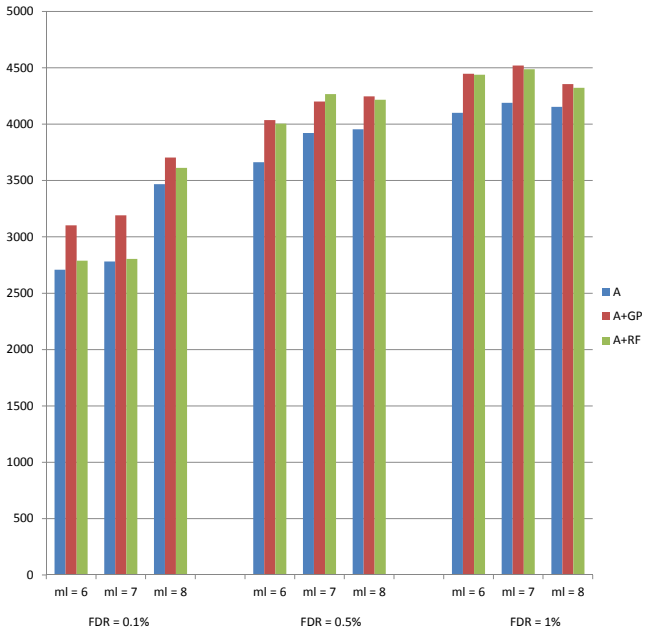

Figure 7: PSMs identified for the second data set (*DS2*) using standard MS Amanda (A, blue) compared to results obtained using new scores calculated with models generated by symbolic regression (A+GP, red) and random forests (A+RF, green).

3.3 Test Results Discussion

The results summarized in Tables 1 and 2 show that in all cases, i.e., for both data sets and for all choices of minimum peptide length and false discovery rate, the scores calculated by models identified by nonlinear modeling lead to better peptide identification rates.

- For the high resolution data set (*DS1*) we see that the number of identified PSMs can be increased by 10% – 20%:
  - The highest relative increase (up to +25%) can be seen for 0.5% FDR, and also for 0.1% FDR and also 1% FDR the number of PSMs can be increased by up to 20% and more.
  - For FDR 0.1% RFs show a better performance than models identified using GP,
  - whereas models learned using GP perform better when setting the false discovery rate to 1%.
- For the low resolution data set (*DS2*) we see that the performance increase is not as high as for the high resolution data, but still the numbers of identified PSMs can be increased significantly. We here see that the models identified by genetic programming perform better than RFs:
  - For 0.1% FDR, using scores calculated by models identified by GP the performance can be increased by up to 14%, whereas using RFs increases the performance by up to 4%.
  - For 0.5% FDR, both machine learning approaches tested here lead to performance increases of 7% – 10%.
  - For 1% FDR, both machine learning approaches tested here lead to performance increases of 4% – 8%, where results achieved using GP are slightly better than those achieved using RFs.

4. CONCLUSION

We have tested various machine learning approaches for calculating new scores for peptide spectrum matches of high accuracy mass spectra. Results show that not only black box modeling (using RFs or SVMs, as used in Percolator), but also white box modeling (using symbolic regression) is perfectly well suited for improving the separation of correct and false peptide identifications of mass spectra. White box approaches generate models that can be analyzed regarding their structure and variable impacts, and they can also be compared for different data sets as those models are transparent. Components of these models can provide further insight into characteristics of target versus decoy identifications which may additionally be integrated in peptide identification algorithms in advanced scoring models. Future plans include detailed analysis of the generated models to extract significant differentiation properties that shall further be integrated in the peptide identification algorithm MS Amanda. The comparison of models generated for different data sets will help us to gain further insight in peptide identification.

5. ACKNOWLEDGMENTS

This work was supported by the Austrian Science Fund (FWF, TRP 308-N15) as well as the Austrian Research Promotion Agency (FFG, K-project HOPL).

6. REFERENCES

[1] Michael Affenzeller, Stephan Winkler, Stefan Wagner, and Andreas Beham. *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*. Chapman & Hall / CRC, 2009.

[2] Thomas E. Angel, Uma K. Aryal, Shawna M. Hengel, Erin S. Baker, Ryan T. Kelly, Errol W. Robinson, and Richard D. Smith. Mass spectrometry-based proteomics: existing capabilities and future directions. *Chemical Society reviews*, 41(10):3912–3928, May 2012.

[3] Wolfgang Banzhaf and Christian W.G. Lasarczyk. Genetic programming of an algorithmic chemistry. In U. O'Reilly, T. Yu, R. Riolo, and B. Worzel, editors, *Genetic Programming Theory and Practice II*, pages 175–190. Ann Arbor, 2004.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Jürgen Cox, Nadin Neuhauser, Annette Michalski, Richard A. Scheltema, Jesper V. Olsen, and Matthias Mann. Andromeda: A peptide search engine integrated into the maxquant environment. *Journal of Proteome Research*, 10:1794–1805, 2011.

[6] Viktoria Dorfer, Peter Pichler, Thomas Stranzl, Johannes Stadlmann, Thomas Taus, Stephan M. Winkler, and Karl Mechtler. MS Amanda, a universal identification algorithm optimized for high accuracy tandem mass spectra. *Journal of Proteome Research*, 13:3679–3684, 2014.

[7] Joshua E. Elias and Steven P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods*, 4(3):207–14, March 2007.

[8] Jimmy K. Eng, Ashley L. McCormack, and John R. Yates III. 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*, 5(11):976–989, 1994.

[9] Lukas Käll, Jesse D. Canterbury, Jason Weston, William Stafford Noble, and Michael J. MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature methods*, 4(11):923–925, 2007.

[10] Thomas Köcher, Peter Pichler, Remco Swart, and Karl Mechtler. Analysis of protein mixtures from whole-cell extracts by single-run nanoLC-MS/MS using ultralong gradients. *Nature protocols*, 7(5):882–90, May 2012.

[11] Michael Kommenda, Gabriel Kronberger, Stefan Wagner, Stephan Winkler, and Michael Affenzeller. On the architecture and implementation of tree-based genetic programming in heuristiclab. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, pages 101–108, New York, NY, USA, 2012. ACM.

[12] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.

[13] Roger E Moore, Mary K Young, and Terry D Lee. Qscore: an algorithm for evaluating SEQUEST database search results. *Journal of the American Society for Mass Spectrometry*, 13(4):378–86, April 2002.

[14] David N. Perkins, Darryl J. C. Pappin, David M. Creasy, and John S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551–3567, 1999.

[15] Mark R. Segal. *Machine Learning Benchmarks and Random Forest Regression*. Center for Bioinformatics & Molecular Biostatistics, 2004.

[16] Jamie Shotton, Tae-Kyun Kim, and Bjorn Stenger. Boosting & randomized forests for visual recognition. In *ICCV 2009*, 2009.

[17] The UniProt Consortium. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Research*, 41:D43–D47, 2013.

[18] Marc Vaudel, Harald Barsnes, Frode S. Berven, Albert Sickmann, and Lennart Martens. SearchGUI: An open-source graphical user interface for simultaneous OMSSA and X!Tandem searches. *Proteomics*, 11(5):996–9, March 2011.

[19] Marc Vaudel, Julia M. Burkhart, René P. Zahedi, Eystein Oveland, Frode S. Berven, Albert Sickmann, Lennart Martens, and Harald Barsnes. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nature Biotechnology*, 33(1):22–24, January 2015.

[20] Stefan Wagner, Gabriel Kronberger, Andreas Beham, Michael Kommenda, Andreas Scheibenpflug, Erik Pitzer, Stefan Vonolfen, Monika Kofler, Stephan M. Winkler, Viktoria Dorfer, and Michael Affenzeller. Architecture and design of the heuristiclab optimization environment. *Advanced Methods and Applications in Computational Intelligence, Topics in Intelligent Engineering and Informatics*, 6:197–261, 2013.

[21] Stephan M. Winkler. *Evolutionary System Identification - Modern Concepts and Practical Applications*. PhD thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz, 2008.