Peptide Detectability following ESI Mass Spectrometry: Prediction using Genetic Programming

David C Wedge
Manchester Interdisciplinary Biocentre
School of Chemistry
The University of Manchester
131 Princess Street
Manchester, M1 7DN
United Kingdom
+44 (0) 161 3065145
david.wedge@manchester.ac.uk

Simon J Gaskell
Michael Barber Centre for Mass Spectrometry
Manchester Interdisciplinary Biocentre
School of Chemistry
The University of Manchester
131 Princess Street
Manchester, M1 7DN
United Kingdom
Simon.Gaskell@manchester.ac.uk

Simon J Hubbard
Michael Smith Building
Faculty of Life Sciences
The University of Manchester
Manchester M13 9PT
United Kingdom
Simon.Hubbard@manchester.ac.uk

Douglas B Kell
Manchester Interdisciplinary Biocentre
School of Chemistry
The University of Manchester
131 Princess Street
Manchester, M1 7DN
United Kingdom
dbk@manchester.ac.uk

King Wai Lau
Faculty of Life Sciences
Michael Smith Building
The University of Manchester
Manchester M13 9PT
United Kingdom
k.lau@manchester.ac.uk

Claire Eyers
Dorothy Hodgkin Fellow
Michael Barber Centre for Mass Spectrometry
Manchester Interdisciplinary Biocentre
School of Chemistry
The University of Manchester
131 Princess Street
Manchester, M1 7DN
United Kingdom
Claire.Eyers@manchester.ac.uk

ABSTRACT

The accurate quantification of proteins is important in several areas of cell biology, biotechnology and medicine. Both relative and absolute quantification of proteins is often determined following mass spectrometric analysis of one or more of their constituent peptides. However, in order for quantification to be successful, it is important that the experimenter knows which peptides are readily detectable under the mass spectrometric conditions used for analysis. In this paper, genetic programming is used to develop a function which predicts the detectability of peptides from their calculated physico-chemical properties. Classification is carried out in two stages: the selection of a good classifier using the AUROC objective function and the setting of an appropriate threshold. This allows the user to select the balance point between conflicting priorities in an intuitive way. The success of this method is found to be highly dependent on the initial selection of input parameters. The use of brood recombination and a modified version of the multi-objective FOCUS method are also investigated. While neither has a significant effect on predictive accuracy, the use of the FOCUS method leads to considerably more compact solutions.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007...$5.00.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: LIFE AND MEDICAL SCIENCES - *Biology and genetics*

General Terms

Algorithms, Design, Theory

Keywords

Genetic Programming, input selection, classification, AUROC, proteomics, mass spectrometry

1. INTRODUCTION

Precise measurement of protein quantities within a biological sample is crucial to the development of various scientific techniques. Applications range from medical diagnosis [16] through systems biology [1] to nutritional analysis [11]. Mass spectrometry (MS) is an increasingly popular method for determining the absolute quantity of defined proteins within a given system.

All MS techniques use the mass/charge (m/z) ratio to distinguish different analytes. The application of MS to protein quantification relies on the use of differentially isotope labeled peptide surrogates, either produced synthetically (AQUA) [11] or via an artificial protein of concatenated peptides (QconCAT) [3]. These reference peptides are designed to mimic the native peptide produced following proteolysis of the protein(s) being quantitated. Known concentrations of the labeled reference peptides are added to the biological sample containing the protein(s) of interest either before (QconCAT) or after (AQUA) digestion with a protease, commonly trypsin.

The protease has the effect of splitting the proteins into various shorter-chain peptides. The resultant complex peptide mixture may then be partially separated using a variety of liquid chromatographic methods before mass determination. The instrument used in this study is an electrospray ionisation (ESI) quadrupole time-of-flight (Q-ToF) mass spectrometer and is particularly suitable for the accurate mass determination and quantification of polypeptides [9].

The native peptide and the differentially labeled reference peptide differ only by the presence of different isotopes and they have the same physico-chemical properties. Therefore their behaviour during chromatographic separation and following ionisation and detection in the mass spectrometer is identical. However, as the reference peptide has a defined mass difference from the native peptide (determined by the method of isotope labeling) it can be easily distinguished based on their m/z ratio. Precise determination of the amount of native peptide can then be determined by comparison of the signal intensity with the known quantity of reference peptide. From this, the absolute concentration of the protein from which the native peptide was derived can be inferred.

The most problematic step in this method is selecting the best peptides for quantification, i.e. those that give the best signal-to-noise ratio following MS analysis. Not all peptides that are produced by cleavage of a protein give a detectable MS peak. Unfortunately, all mass spectrometers detect within a limited m/z range, so particles outside this range will not be detectable. In addition, some peptides will not behave favourably on the chromatographic media used for their separation, either not binding in the first instance, or not eluting. Most protein sequences are split at easily identified sites by trypsin digestion. However, some protein sequences are prone to 'missed cleavages', when some peptide bonds fail to cleave at the expected site. Use of the peptides resulting from these cleavages would therefore result in inaccurate quantification. Other peptides are unlikely to be ionised and will therefore be invisible to mass spectrometric detection [18]. Other characteristics, as yet undefined, also determine peptide detectability by ESI-MS.

Predicting whether any particular peptide will be detectable is clearly a complex task. However, the properties described above (ionisability, response to trypsin digestion, elution behaviour, etc.) are all chemical properties which should depend upon the chemical structure of the peptide in question. A variety of quantitative chemical properties of peptides are readily obtained as described in Section 2. These may then be used as independent variables in a statistical or machine learning technique.

Two previous studies have used decision trees [10] and artificial neural networks (ANNs) [18] to map the chemical properties of proteins to their MS detectability, using different ionisation methods. In addition to the overall properties of the peptides, both studies used some information concerning the positions of individual amino acids in the peptide chains and information concerning the environment of the target peptide within the original protein, i.e. the identities of nearby amino acids.

Previous studies have used genetic algorithms to solve a related problem, the optimization of experimental settings for ESI-MS [19]. In this study, we use genetic programming (GP) to create a mathematical function that relates the chemical properties of a peptide to its detectability by ESI-MS. We do not use amino acid positional information or information concerning the environment of the peptide within its parent protein. It is intended that this information will be incorporated in later investigations. Our method has 2 steps-

- In the first step, a population of programs is evolved using the area under the receiver-operator characteristic curve (AUROC) as the objective function. The AUROC is a measure of the extent to which a model can differentiate between positive and negative outputs. It is calculated across the range of input-output space and it therefore summarises the whole decision surface. The AUROC function is described in more detail in section 3.1 and in a number of other studies, for example [6].
- In the second step this decision surface is examined in more detail by considering how the classification accuracy varies as the output threshold is adjusted.

The advantage of this 2-step method is that it allows the user to interact with the system, after the creation of a 'good' model. The threshold is chosen by the user while taking into account the characteristics of a working model. This issue is discussed in full in section 3.3.

The rest of this paper is made up of the following sections. Section 2 describes how the data - physico-chemical parameters and MS peak intensities - were obtained. Section 3 describes the experimental method used for this research. It is comprised of subsections on the AUROC function, data sampling, threshold setting and the selection of input parameters. Section 4 states the design parameters used to create and evolve the GP itself. Section 5 is a results section and Section 6 a conclusion. The paper ends with acknowledgements (Section 7) and references (Section 8).

2. PEPTIDE DATA

For each peptide, 393 properties were calculated by averaging the property values of each individual amino acid over the whole peptide. These parameters cover a wide range of physico-chemical properties. Some example properties are hydrophobicity, isoelectric point, molecular mass and predicted proportion of a particular secondary structure (e.g. alpha-helical). These properties are hereafter referred to as input variables, or simply 'inputs'. On the other hand, the set of input values that refer to a particular peptide sequence are described as 'input vectors'. The values of all input variables for each peptide were normalised to a range of [-1, 1] via a linear transformation.

The behaviour of trypsin as a proteolytic enzyme is well understood and the peptides produced may be predicted using an '*in silico* digest' of each protein. This involves the application of a simple rule – the enzyme cleaves each amino acid after every lysine or arginine residue, unless followed by a proline. This rule was applied to 13 protein sequences, resulting in 931 unique peptides (including peptides with up to one missed cleavage).

These peptides were then cross-referenced with those generated in the laboratory using in-solution tryptic digestion on the same 13 proteins, which are commercially available as purified proteins. The signal intensity of each of the 931 unique peptides generated from these proteins was determined following reversed-phase chromatographic separation in-line with electrospray ionization and detection using a Q-ToF mass spectrometer (from Waters Corporation). A little over half of the peptides resulted in measurable signals. The target outputs were Boolean, i.e. each peptide is/is not detectable. However, in future we intend to apply GP to the prediction of signal intensities.

3. EXPERIMENTAL METHOD

3.1 Objective Function

The objective function used by the GP was the AUROC. This measure is widely used in medical decision-making. We briefly describe this function here. More complete analyses are provided in various texts, for example [6], [20].

When classifying data as true or false, there are 4 possible outcomes: true positive, false positive, true negative and false negative. These classification types are commonly represented in a confusion matrix, as in Table 1. The number of true positives may be expressed as a fraction of the actual number of positives in the sample and the other quantities may similarly be expressed as fractions.

Table 1. Binary confusion matrix showing classification types

	Positive	**Negative**
Positive prediction	True positive (TP)	False positive (FP)
Negative prediction	False negative (FN)	True negative (TN)

Many tests result in a numeric output, which will then result in a 'true' or 'false' prediction, depending whether the output is greater than or less than some threshold. By using a low threshold, the number of positive predictions will be increased. Some of these predictions may be true positives. On the other hand some of them may be false positives. Receiver-Operator Characteristic (ROC) curves focus on the 'true' predictions, plotting the true positives against the false positives, as illustrated in Figure 2. Following the curves from left to right corresponds to lowering the threshold, resulting in a greater proportion of both true and false positives.

For a particular predictor the ROC curve may be estimated by obtaining the output from each input vector. The predictor's threshold is then successively set to each of these output values, each threshold giving rise to a point on the ROC curve.

The ideal situation occurs when a particular threshold completely separates the positive and negative data, resulting in a true positive rate of 1.0 and a false positive rate of 0.0. This ideal is indicated by the upper dashed line in Figure 1. The area under the ROC curve, or AUROC, is a measure of the extent to which the predictive model approaches this ideal. It is easily estimated using the trapezium rule. Models with an AUROC value close to 1.0 (the upper dashed line in Figure 1) allow a fairly accurate separation of positive and negative samples. On the other hand models with an AUROC close to 0.5 (the lower dashed line) have little or no discriminatory power.

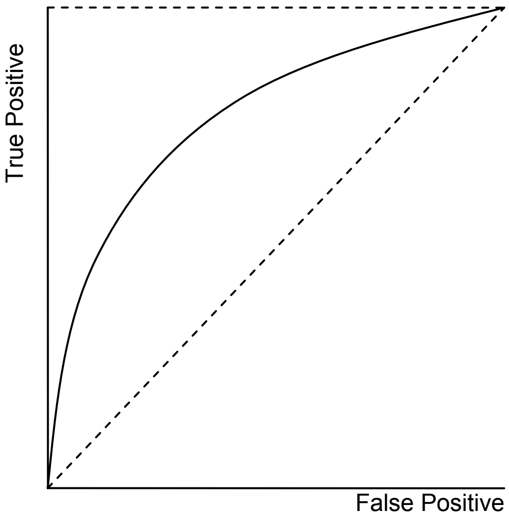

Figure 1. A Receiver-Operator Characteristic Curve

3.2 Data Sampling

The data were split randomly into 10 partitions and 10-fold cross validation was performed 10 times. Thus, each partition was 'held back' for testing in turn. The remaining data were split randomly in the ratio 2:1 into training and validation sets, 10 times. Overall, the data were therefore split 100 times into training, validation and test sets in the ratio 6:3:1. For each split, a population of solutions (programs) was created and evolved using the training data.

The validation data was used for 2 purposes, identifying an early stopping point and program selection. Evolution ran for a maximum of 500 generations. However, evolution was stopped if there had been no improvement in the best-performing program on the *validation* data for 25 generations, in order to avoid overfitting. The 'patience' of the algorithm was set at 25 generations following an initial investigation which showed that validation accuracy frequently improved after 10 generations of stagnation but very rarely improved after 25 generations of stagnation. It was found that very few evolutionary runs completed all 500 generations, suggesting that the algorithm ran to convergence in the great majority of cases. Once evolution had been halted, the program that gave the highest AUROC on the validation data was selected as the 'best' program.

3.3 Threshold Setting

The next task was to choose a threshold. Each program gives a real-valued output for each input vector. The output can be calculated for each point within the training set. A threshold can then be set which results in the required number of positive or negative outputs. The dataset used here contains 501 detectable peptides and 430 peptides that were not detected. However, only

60% of the data is used for training purposes. One would therefore expect to find about 300 (around 60% of 501) detectable peptides within the training data.

A 'balanced' prediction may therefore be made by setting the threshold such that 300 peptides are predicted to be detectable. It is however possible to increase the threshold so that the number of peptides predicted to be detectable is reduced, but the confidence with which they are predicted will, hopefully, be increased. The confidence in the prediction may be expressed as the positive predictive value (PPV), which is the proportion of predicted positive values that are actually positive, i.e. TP/(TP+FP). 6 different thresholds were set for each solution, corresponding to the prediction of 300, 270, 240, 210, 180 and 150 detectable peptides within the training data. These correspond to 100%, 90%, 80%, 70%, 60% and 50%, respectively, of the expected number of detectable peptides. Using a number of different thresholds allows the user to get a picture of the decision surface. In particular it is possible to analyse the effect of increasing thresholds in reducing the number of positive predictions but increasing the PPV.

Note that the thresholds are set using the *training* data as a guide. The test data are not used in the selection of the best program or in threshold-setting. This means that the test data may be used to give an unbiased assessment of the performance of each program. All results quoted in Section 5 are obtained using these data.

3.4 Input Selection

The search space using all 393 input variables is clearly extremely large. In order to investigate whether the large search space made optimisation difficult, we performed two further series of runs using subsets of the available inputs. After carrying out evolution using all inputs, the inputs were placed in order of their frequency within the final populations, i.e. the number of occurrences of a node corresponding to each input. These values were normalised by dividing by the total number of programs produced, giving the average number of times each input was used per program.

The GP algorithm was then repeated using subsets of the original inputs.

- Firstly, only the inputs that had an average usage in excess of 0.05 per program were included. This criterion reduced the number of inputs to 34.
- Secondly, just those inputs that had an average usage in excess of 1.00 per program were included. This criterion reduced the number of inputs to 6.

Results using all 393 inputs and those from using the two reduced input sets are compared in Section 5.

4. GENETIC PROGRAM DESIGN

The 'programs' used in this study were binary trees, whose internal nodes were simple mathematical functions, i.e. +, -, ×, ÷. (The division function was made a closed function by defining the result of dividing by zero as 0.) Terminal nodes were either inputs or constant values.

The population size was 100, with initial populations generated using the ramped half-and-half method with starting tree depths between 5 and 10 inclusive. Evolution took place in steady-state mode using tournaments of size 2 to select both parents and programs to be removed. Crossover was performed by swapping randomly chosen subtrees from 2 parents, with a probability of 0.7. Six different mutation operators were used: subtree replacement, point replacement, shrink, swap, constant value mutation and expand. These operators are described in [14] and [4]. The overall mutation rate was 0.2, with the operator being chosen at random.

The training procedure was repeated using a crossover operator that included brood recombination [17], with a brood size of 8. This may give a 'push' to the evolutionary process in situations where the crossover operator is unable to explore the search-space adequately. Brood recombination may be viewed as an attempt to introduce a more homologous crossover operator, by 'weeding out' the products of destructive crossover [4]. An alternative way of viewing brood recombination is that it improves the 'evolvability' of programs [2].

Further factors influencing the performance of a GP are the maintenance of diversity and avoidance of bloat within the population. To test whether this is an important factor with the peptide dataset, experiments were also performed using a multi-objective function related to Edwin de Jong's 'Find Only and Complete Undominated Sets' (FOCUS) method [13]. This method uses a 3-objective function composed of the basic fitness function, a measurement of diversity based upon the average 'distance' of a program from all other programs within the population and the length of a program. The aim is to maximise the diversity and minimise the length of a program while optimising the basic fitness function.

In the original FOCUS method, only non-dominated programs were maintained in the population. Other programs were discarded, leading to small population sizes. Our findings on a number of datasets, to be published in future work, indicate that this procedure can lead to a collapse in the population to very small numbers of individuals and hence premature convergence. In this study we therefore use a modified form of the FOCUS method. We keep a constant population size and select programs for breeding and removal using tournament selection based on a ranking of programs according to the number of other programs by which each program is dominated.

Overall, we report the results of 12 different experiments: using 6, 34 or 393 input parameters; with and without brood re-combination; and with and without a multi-objective fitness function.

All computer code has been written by DCW in the Java programming language. It is intended that the source code and documentation will be made available for download from the webpage http://dbkgroup.org/dcw/.

5. RESULTS

5.1 Positive Predictive Values and Sensitivities

Table 2 shows the positive predictive value (PPV) and sensitivity for balanced prediction, i.e. the number of positive predictions on the training sample is approximately equal to the expected number of positive predictions. The positive predictive value is the fraction TP / (TP+FP), i.e. the fraction of peptides that are predicted to be detectable that are actually detectable. The sensitivity is defined as TP / (TP+FN), i.e. the fraction of the samples that are actually detectable that are predicted to be detectable. The last line in Table 2 may be interpreted thus:

'Of the peptides that are actually detectable, 70.3% of them are predicted to be detectable (along with some false positive predictions). Positive predictions are made with 69.9% confidence.'

When the scientific context of the data is considered it may be seen that optimisation of the PPV is of greater importance than optimisation of the sensitivity. Each protein gives rise to a number of peptides upon trypsin digestion. However, not all possible peptides are needed for protein identification (and quantification). It has been shown that identification of as little as 20% of the possible peptides is in some cases sufficient to allow protein identification [8]. A sensitivity considerably lower than 100% may therefore be acceptable. On the other hand, it is important to have a high PPV. Creating isotope labeled peptides for quantification is time-consuming and expensive. Identifying detectable peptides with high confidence will reduce the frequency with which effort is devoted to producing peptides that are not detectable.

A comparison of the first four and second four rows shows that using just 34 inputs improves both the fraction of detectable peptides that are identified and the confidence with which these predictions are made. The last four rows show further small improvements in PPV and sensitivity under most algorithms upon reducing the number of inputs to 6.

Introducing multi-objective evaluation into the algorithm is seen to improve the performance when all 393 inputs are used. On the other hand, the introduction of brood re-combination does not have a beneficial effect on PPVs and sensitivities and may even be detrimental.

Table 2. Results, showing average cross-validated positive predictive value (PPV) and sensitivity

Number of inputs	Brood re-combination ?	Multi-objective ?	PPV	Sensi-tivity
393	No	No	0.634	0.646
393	Yes	No	0.606	0.606
393	No	Yes	0.667	0.668
393	Yes	Yes	0.667	0.667
34	No	No	0.703	0.701
34	Yes	No	0.700	0.704
34	No	Yes	0.700	0.692
34	Yes	Yes	0.704	0.699
6	No	No	**0.707**	**0.705**
6	Yes	No	0.701	0.697
6	No	Yes	0.705	0.704
6	Yes	Yes	0.699	0.703

When applied to the 34-input and 6-input models neither brood re-combination nor a multi-objective evaluation function had substantial effect. The implication of this is that reducing the size of the search space enables a basic GP algorithm to perform well, by focusing on the models that are likely to be more informative. On the other hand, when the search space is very large, the evolutionary process has more difficulty finding a satisfactory model and is therefore more sensitive to the algorithm used.

5.2 Program Sizes

Table 3 shows the average size (number of nodes) of the best programs produced for each run. It is clear from this table that the modified FOCUS method produces considerably more compact solutions than the basic method. The incorporation of the multi-objective function into the GP therefore serves a purpose in producing input-output relationships that are much more readily interpretable. The use of brood re-combination on the other hand has a smaller and variable effect on program size.

Table 3. Average cross-validated program sizes (number of nodes) for the best-performing programs under a variety of GP algorithms

Number of inputs	Brood re-combination ?	Multi-objective ?	Program size
393	No	No	72.5
393	Yes	No	74.3
393	No	Yes	21.5
393	Yes	Yes	15.3
34	No	No	75.9
34	Yes	No	76.4
34	No	Yes	18.4
34	Yes	Yes	18.2
6	No	No	70.2
6	Yes	No	77.6
6	No	Yes	21.4
6	Yes	Yes	21.2

5.3 Threshold Setting

Figure 2 shows the variation in average PPV and sensitivity values as the threshold is adjusted, for programs trained with 34 input parameters and a basic GP algorithm, i.e. without brood

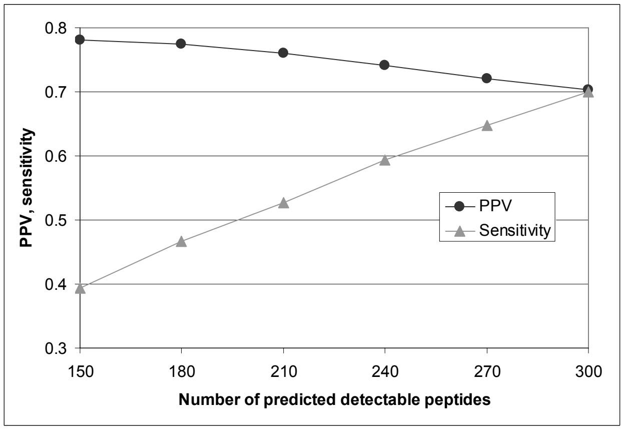

Figure 2. Average cross-validated positive predictive value (PPV) and sensitivity as a function of the number of peptides predicted to be detectable, using 34 input parameters

recombination or FOCUS. As expected, increasing the threshold (moving from right-to-left) reduces the sensitivity but increases the PPV, or confidence level, with which predictions are made.

Figure 3 shows the average PPVs of GPs trained with 393, 34 and 6 inputs. Also included is the performance of the best performing solution, i.e. the program with the highest AUROC on validation data. This figure confirms the improvement in performance upon reducing the number of input parameters. The results for the 'best' solution show the advantage of using the AUROC as an objective function. High AUROC values imply that it is possible to increase the proportion of true positives without a large increase in the number of false positives, i.e. there is a point in Figure 1 close to (0,1). We see this for the best solution (the top line in Figure 3). As the number of predicted detectable peptides is increased (moving from left to right) the PPV falls off for most predictors. However, for the best predictor the PPV remains high as the number of predicted detectable peptides is increased. For this predictor a fairly low threshold may be set without introducing large numbers of false positives.

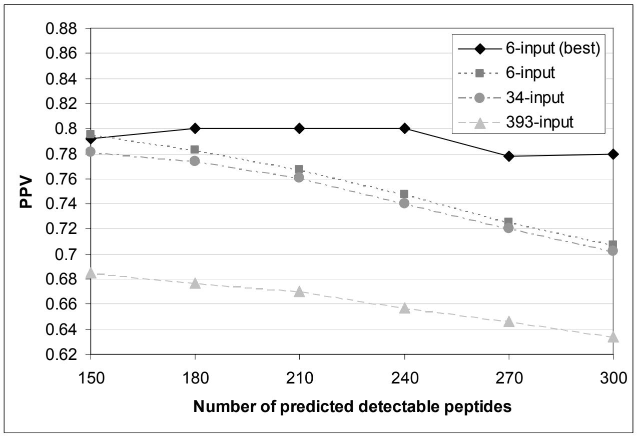

Figure 3. Positive predictive values (PPV) as a function of the number of peptides predicted to be detectable, including average results using 393, 34 and 6 inputs and the result using the best 6-input predictor.

It can be seen from Figure 3 that peptides may be predicted to be detectable with a confidence level of 80% using this predictor. This result is a considerable improvement on results obtained using decision trees [10] and is comparable to those obtained using neural networks [18]. In addition, both earlier studies used further information relating to the internal structure (amino acid sequence) and environment (location within the parent protein) of the peptides. In future studies we intend to include this information and anticipate improvements in classification accuracy.

6. CONCLUSIONS AND FURTHER WORK

We have shown that genetic programming is an effective tool in the prediction of mass spectrometric peptide detectability, giving results that compare favorably with those obtained using neural networks or decision trees. An additional advantage of the method described here is that the model and the threshold to be used are set independently. This allows a user to assess the balance between sensitivity and confidence in predictions made on a working model.

The identification of the most significant inputs is a simple task for the GP technique, unlike approaches such as neural networks. This allows the user to evaluate those physico-chemical properties that may affect peptide detectability, thus ultimately improving our understanding of the mechanisms involved in peptide analysis by MS. Our understanding of these mechanisms is further enhanced by the use of a multi-objective function which reduces the incidence of program bloat.

It was found that re-running the GP on a subset of the original input parameters, selected by their relative frequency in successful trees, enabled the search algorithm to perform better while using the same GP algorithm. This was seen to be more effective than improving the GP algorithm through the introduction of batch recombination or a multi-objective fitness function.

We cast some light on the efficacy of different objective functions and GP methods when applied to a 'real' dataset. We have specifically looked at the use of AUROC, PPV and sensitivity statistics as measures of a classifier's performance and at the FOCUS method and brood re-combination as attempts to improve the performance of the GP itself.

We intend to incorporate additional information concerning individual amino acid residues into future models and anticipate improvements in classification accuracy. Genetic Programming will also be applied to the problem of predicting actual MS peak intensities, rather than just a binary classification (is/is not detectable). This should allow the generation of expected spectrograms, which may be compared with the observed spectrograms.

In parallel with the development of genetic programming methods one of the authors (KW) has been developing two further algorithms for the prediction of peptide detectability. These use decision trees [5] and a nearest-neighbour algorithm [7]. As a part of the QconCat project the authors are in the process of combining all three methods, so creating a classification method based on a consensual approach.

We believe that the development model used here, in which the identification of an effective classifier and the choice of a threshold are carried out in separate steps, could be profitably applied to other classification problems. Investigations are ongoing into the effectiveness of the FOCUS method and modifications thereof.

7. ACKNOWLEDGMENTS

This research is partially supported by the United Kingdom Home Office (DCW) and the BBSRC (DBK). This research is also supported by the BBSRC via grants EGM17685 (CE, KW, SJG, SJH), BBSB17204 (SJH), and the EPSRC via grant EP/D013615/1 (SJG, SJH, KW).

8. REFERENCES

[1] Aebersold, R. and Mann, M. Mass spectrometry-based proteomics. In *Nature, 422* (Mar. 2003), 198-207.

[2] Altenberg, L. The evolution of Evolvability in Genetic Programming. In *Advances in Genetic Programming*, K.E. Kinnear, K.E. (ed.), 47-74. MIT Press, Cambridge, MA, 1994.

[3] Beynon, R.J., Doherty, M.K., Pratt, J.M and Gaskell, S.J. Multiplexed absolute quantification in proteomics using artificial QCAT proteins of concatenated signature peptides. In *Nature Methods, 2, 8* (Aug. 2005), 587-589. Published online at http://www.nature.com/nmeth/journal/v2/n8

[4] Banzhaf, W., Nordin, P., Keller, R.E. and Francone, F.D. *Genetic Programming – An Introduction*, Morgan Kaufmann, San Fransisco, CA, 1998.

[5] Breiman, L., Friedman, J., Olshen, R. and Stone, C. *Classification and Regression Trees*. Chapman & Hall / CRC, 1984

[6] Broadhurst, D.I. and Kell, D.B. Statistical Strategies for Avoiding False Discoveries in Metabolomics and Related Experiments. *Metabolomics, 2, 4* (Dec. 2006), 171-197.

[7] Cover, T. and Hart, P. Nearest neighbor pattern classification. In *IEEE Transactions on Information Theory, 13, 1* (Jan. 1967), 21-27.

[8] Eriksson, J., Chait, B.T. and Fenyo, D. A Statistical Basis for Testing the Significance of Mass Spectrometric Protein Identification Results. *Analytical Chemistry 72, 5* (Mar. 2000), 999-1005.

[9] Fenn, J.B., Mann, M., Meng, C.K., Wong, S.F. and Whitehouse, C.M. Electrospray ionization for mass spectrometry of large biomolecules. *Science, 246, 4926* (Oct. 1989), 64-71.

[10] Gay, S., Binz, P.-A., Hochstrasser, D.F. and Appel, R.D. Peptide mass fingerprinting peak intensity prediction: Extracting knowledge from spectra. In *Proteomics, 2, 10* (Nov. 2002), 1374-1391.

[11] Gerber, S.A., Rush, J., Stemman, O., Kirshner, M.W. and Gygi, S.P. Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. In *PNAS, 100, 12 (Jun' 2003)*, 6940-6945

[12] Gianazza,E., Eberini, I., Arnoldi, A., Wait, R. and Sirtori, C.R. A Proteomic Investigation of Isolated Soy Proteins with Variable Effects in Experimental and Clinical Studies. In *The Journal of Nutrition, 133, 1* (Jan. 2003), 9-14.

[13] de Jong, E.D., Watson, R.A. and Pollack, J.B. Reducing Bloat and Promoting Diversity using Multi-Objective Methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, (Jul' 2001), 11-18

[14] Langdon, W.B. *Genetic Programming and Data Structures*. Kluwer, Massachusetts, MS, 1998.

[15] Pratt, J.M., Simpson, D.M., Doherty, M.K., Rivers, J., Gaskell, S.J. and Beynon, R.J. Multiplexed absolute quantification for proteomics using concatenated signature peptides encoded by QconCAT genes. In *Nature Protocols, 1, 2* (2006), 1029-1043.

[16] Rifai, N, Gillette, M.A. and Carr, S.A. Protein biomarker discovery and validation: the long and uncertain path to clinical utility. In *Nature Biotechnology, 24, 8* (Aug. 2006), 971-983.

[17] Tackett, W.A. *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, 1994.

[18] Tang H., Arnold, R.J., Alves, P., Xun, Z., Clemmer, D., Novotny, M.V., Reilly, J.P. and Radivojac, P. A computational approach toward label-free protein quantification using predicted peptide detectability. In *Bioinformatics, 22, 14* (Jul' 2006), e481-e488.

[19] Vaidyanathan, S., Broadhurst, D.I., Kell, D.B. and Goodacre, R. Explanatory Optimization of Protein Mass Spectrometry via Genetic Search*. Anal. Chem. 75, 23* (Dec. 2003), 6679-6686.

[20] Westin, L.K., *Receiver operating characteristic (ROC) analysis*. Technical paper, UNINF-01.18, 2001, Umea University, http://www.cs.umu.se/research/report