Genetic Programming for Measuring Peptide Detectability

Soha Ahmed[1], Mengjie Zhang[1] , Lifeng Peng[2], and Bing Xue[1]

[1] School of Engineering and Computer Science
[2] School of Biological Sciences
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{soha.ahmed,mengjie.zhang,bing.xue}@ecs.vuw.ac.nz, lifeng.peng@vuw.ac.nz

Abstract. The biomarker discovery process usually produces a long list of candidates, which need to be verified. The verification of protein biomarkers from mass spectrometry data can be done through measuring the detection probability from the mass spectrometer (Peptide detection). However, the limited size of the experimental data and lack of a universal quantitative method make the identification of these peptides challenging. In this paper, genetic programming (GP) is proposed to measure the detection of the peptides in the mass spectrometer. This is done through measuring the physicochemical chemicals of the peptides and selecting the high responding peptides. The proposed method performs both feature selection and classification, where feature selection is adopted to determine the important physicochemical properties required for the prediction. The proposed GP method is tested on two different yeast data sets with increasing complexity. It outperforms five other state-of-the-art classification algorithms. The results also show that GP outperforms two conventional feature selection methods, namely, Chi Square and Information Gain Ratio.

1 Introduction

Biomarkers are indicators of a specific biological or disease state [18]. They are important for many clinical applications and classification of different stages of diseases. Biomarker detection methods usually produce many biomarkers [11], and it is necessary to verify those biomarkers before passing them to clinical validation [18]. The peptide detection (helps in verifying candidate biomarkers) is a classification problem where the task is to classify peptides as flyers or non-flyers [12]. The detectable peptides (referred to as quantifiable surrogates) are the peptides that are characterised to be high responding in the body fluids (e.g. blood) [9]. This process of discovering the quantifiable surrogates is called verification, which is a necessary process to bridge the gap between the biomarker discovery and the clinical-validation experiments [24]. The verification process is typically done through the absolute quantification of peptides [1]. The verification of biomarkers is a hard problem due to the high dynamic range of proteins [18], the complexity of the data and the lack of a universal quantitative method.

G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, pp. 593–604, 2014.
© Springer International Publishing Switzerland 2014

Mass spectrometry (MS) is capable of sensitive detection, identification and quantification of proteins. Mass spectrometer measures the molecular weight of the peptides (with respect to a charge ratio) and its corresponding intensity. The product spectrum is composed of the mass to charge ratio (m/z) and the intensity of peptides. Mostly, MS is accompanied with liquid chromatography for separation of the sample which helps decrease the complexity of the sample. The produced LC-MS spectrum contains the m/z, the intensity and the retention time of the peptides. MS-based quantification faces the problem of selecting the best quantifiable peptides that can give detectable MS peak. Therefore, machine learning methods can be useful to automatically predict the high responding peptides.

The physicochemical properties of the peptides can represent the feature vector for predicting the detectability of peptides. Mostly, the peptide detection data sets are composed of a large number of features (properties) some of which can be irrelevant to the classification task. Hence, an effective and powerful method is needed to perform two tasks. Firstly, feature selection is needed to select important physiochemical properties. Secondly, the classification of the data aims to determine the detection probability.

Genetic programming (GP) is an evolutionary technique which has been used successfully for feature selection and classification [5,8]. GP solves a problem by evolving computer programs (functions) [22]. It usually starts with a population of random programs then modifies these programs using its genetic operators [30]. The GP algorithm consists of the following steps [22]:

1. Initialize a random population of programs;
2. Calculate the goodness of each program through the fitness function;
3. If the stopping criteria are not met, do the following:
   - Select some good programs through the selection method;
   - Use the genetic operators to perform the changes on the selected programs;
   - Pass the new programs to the following generation;
   - Calculate the fitness of the programs in the new generation;
4. Return the program with the best fitness as the designed solution.

GP has the potential to perform feature selection and classification at the same time [26], and due to the high dimensionality of peptide data, GP is a good choice for solving peptide detection problem. This paper represents one of the few attempts to use GP for selecting important features required for peptide detection.

Goals. The main goal of this paper is to develop a new GP method for measuring peptide detectability. The proposed method performs two important tasks. Firstly, feature selection that helps in determining the important physiochemical properties for detection of peptides. Secondly, prediction of flyers (detectable) and non-flyers (non-detectable) peptides which will be useful for both verification of biomarkers candidates and at the same time *absolute quantification* of peptides. Precisely, we will investigate the following:

1. What is the appropriate fitness measure which can make GP reduces the number of selected features with preserving the maximum classification performance?
2. Can GP outperform conventional feature selection and classification methods?
3. What are the important physiochemical properties selected?

Organisation. The rest of the paper is organised as follows. Section 2 discusses the related work on using GP for feature selection and classification and also the previous work done on peptide detection. Section 3 describes the proposed GP approach for peptide detection. Section 4 presents the experiment setup, the data sets description and the feature vector production process. Section 5 reports the full experiment results and discussions. Section 6 concludes the paper and gives some directions for future work.

2 Related Work

2.1 GP for Feature Selection and Classification

GP has been successfully used to select features in either filter or embedded approaches [27, 28]. The advantage of GP for building classification models (without the need to be wrapped to another classifier) makes it a perfect choice for performing both classification and feature selection tasks, especially in high dimensional data such as in [2–4]. GP has been also used to solve the problem of classification of unbalanced data such as in [6–8]. The success of GP in feature selection and classification has encouraged us to use it in prediction of peptide detectability.

2.2 Peptide Detection

Previous studies have been adopted for the use of machine learning techniques for peptide detection [32]. Decision trees and artificial neural networks (ANN) have been used in [15] and [31] to relate the physicochemical properties of proteins to their MS detectability. Evolutionary algorithms were also used in a small number of studies to solve the peptide detection prediction problem in MS data. For example, in [33], genetic algorithms (GAs) have been used to solve this problem where the aim was to reach the optimum experimental conditions for protein detection in MS. GP has been used only in two studies [12, 34] with promising results. Most of these studies were focused on the maximisation of the flyers peptides without taking into account the overall accuracy of prediction both flyers and non-flyers peptides. Moreover, previous studies were mostly focused in determining detectability of peptides based on the whole set of peptides' properties. The advantage of GP to perform both feature selection and classification has not been fully investigated in those studies. In this paper, the determination of the important physicochemical properties for detection prediction is investigated. Moreover, the use of GP system as a prediction system for peptide detectability is also investigated here.

3 The New GP Method for Peptide Detection

3.1 Overall Structure

The proposed GP method is performing two tasks, firstly feature selection, in order to select important physicochemical properties required for accurate prediction, and secondly classification. The method first starts with data set preparation and generation of feature vectors. This is done through search of MS/MS through SEQUEST, which produces a data set containing peptides where the length of each peptide is chosen to be between 5-24 residues. Afterwards the feature vectors are generated, which are composed of the physicochemical properties of each peptide in the data set. For each peptide, 544 properties are extracted from *AAindex* database [19]. The data sets are divided into half for training and half for testing. Only the training set is passed to GP to build a classifier model. The produced model automatically selects features in the terminal nodes of the tree. The selected features are used to form new training and test sets. Finally, the algorithm is applied to the unseen test set to measure the detectability of the peptides.

3.2 Feature Selection

The search space using all of the 544 features (physicochemical property) is extremely large and hence, feature selection should be performed. GP can automatically select features during the evolution process [26]. The terminal nodes of evolved trees contain the selected features for building the classification model. Therefore, GP has the advantage of selecting the features, which have the potential to produce a classifier with better classification performance.

3.3 Peptides Detection (Classification)

Prediction of the detectability of a peptide is a non-trivial classification task which involves complicated relationships between the classification rules and also between the input features [12, 17]. The proposed GP method performs classification by setting a threshold value (as a decision stump) by which the classification decision is taken. If the GP tree output is less than or equal to this threshold, the peptide is classified as detectable (flyer) otherwise, it is classified as non-detectable (non-flyer).

3.4 Improved Fitness Function

The typical standard classification accuracy of the training set may be inappropriate for the peptide data sets due to the large number of features. We aim to select only the most important features. Hence, the fitness function used is designed to take into account feature selection and classification tasks.

The classification of the data as, true or false has four outcomes: true positive, false positive, true negative, false negative. These outcomes are represented using

Table 1. Binary Confusion Matrix

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

a confusion matrix which is shown in Table 1. The first task is to maximise the classification accuracy, the classification accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN}$$

The second task is to reduce the number of features selected by each genetic program. Therefore, we used the following fitness function which is inspired by [25].

$$Fitness = (1 + a * exp^{\frac{n}{N}}) \times Accuracy \tag{1}$$

where n is the number of features selected by GP and N is the original number of features. The exponential factor decreases with increasing n to give more fitness to the program with less features. a is a factor used to measure the importance between reducing the number of features and increasing the classification accuracy. a is equal to the following:

$$a = (1 - \frac{\text{Current}}{\text{Total}}) \tag{2}$$

Current is the index of the current generation while *Total* is the maximum number of generations. Therefore, the fitness function used in Equation (1) will achieve the two tasks which are reducing the number of features, increasing classification performance.

4 Experiments Setup

This section outlines the data sets acquisition, feature vector production, program representation and evolutionary parameters.

4.1 Peptide Data Sets and Feature Vectors Production

Data Sets. Two tryptic peptide data sets are used to test the new method. Both data sets were analysed using LC-ESI-MS and obtained from [12]. The peptides of first data set were generated from 13 proteins. The proteins were searched against NCBInr database [29] using Mascot server [20](Matrix science) to confirm the identity and elution time. Extracted ion chromatograms were generated for the peptides that did not yield tandem MS data. Each peptide contains at least five amino acids and generated with either 0 or 1 missed cleavage and the m/z values range from 300 and 1800. The class label as a flyer or non-flyer was set by cross referencing the peptides with the generated peptides in the lab. This data set (DS_1) contains 931 peptides (501 in flyer and 430 in non-flyer class)

The second data set (DS_2.) was downloaded from *PeptideAtlas* [10] and originally produced from 24 yeast experiments. The total number of proteins is 2733.The peptides' length (number of amino acids) ranges from 6 to 42 residues with 0-2 missed cleavage. Each peptide was assigned a flyer's class label if it was observed in the 24 experiments otherwise, it was assigned a non-flyer class label. The total number of peptides examples is 21515 in which 2121 peptides are in the flyers' class and 19394 are in the non-flyers' class. More details about the data sets can be found in [12].

Feature Vectors. The data sets were obtained in the form of peptide sequences (amino acids) and the class label. Hence, in order to use those peptides with the machine learning techniques they should transformed to numerical feature vectors. The physicochemical properties of the peptides have shown to be related to their detectability [1]. Therefore, for each peptide, 544 properties were calculated to transform the peptide data into numerical feature vectors. The 544 properties were extracted from *AAindex* database [19] and for each peptide sequence the average of the property value of each individual amino acid is calculated over the whole peptide. The physicochemical properties include, for example, *mass, alpha-helica*l which is the predicted percentage of the secondary structure, *hydrophobicity, gasphase basicity* and *isoelectric point*. Therefore, each peptide is an instance used for training and testing the GP algorithm which modeled by 544 feature values and either flyer or non-flyer class label.

4.2 GP Program Representation

The tree structure is used in the experiments as a representation of the GP program [21]. The features and also a randomly generated constant terminal are used in the terminal set. The function set contains the four standard mathematical operators $+, -, \%, \times$ and a conditional operator if, a max operator and a *Abs* operator. The $+, -, \times$ take two arguments and return the addition, subtraction or multiplication of the two arguments. The % is the usual division , which takes two arguments, but it is protected which returns zero if the division is by zero. max returns the maximum of two arguments while if takes three arguments and returns the second argument if the first is negative otherwise, returns the third one. The *Abs* operator takes only one argument and returns the absolute value of this argument. The classification is performed by taking a threshold value of zero in which if the output of the genetic program is less than or equal to zero the peptide is classified as belonging to the flyer class. Otherwise, it is classified as belonging to the non-flyer class.

4.3 GP Evolutionary Parameters

The initial population is generated using the ramped half and half method [21]. The number of individuals in the population is 1024. Crossover, mutation and elitism rates are 70%, 29% and 1% , respectively. The maximum tree depth is set

Table 2. GP evolutionary parameters

Initialization method	Ramped Half-and Half
Tree Depth	8
Number of Generations	100
Mutation rate	29%
Crossover rate	70%
Elitism rate	1%
Population Size	1024
Selection type	Tournament
Tournament Size	4

to 8 in order to avoid bloating. The method of selection used is the tournament method and its size is set to 4. The evolution runs for 100 generations. 50% of the data is randomly selected for training GP and the other 50% is kept as a test set. These parameters are selected based on the literature [3]. Table 2 shows the evolutionary parameters used.

4.4 Methods for Comparison

The proposed GP is compared with several state-of-the-art feature selection and classification algorithms. The Waikato Environment for Knowledge Analysis (WEKA) package [16] is used to run the feature selection and classification algorithms.

Benchmark Classification Methods. Five different classifiers are used (with both the original features and the GP's selected features) and compared to GP classifier. The five classifiers are commonly used in classification tasks.

1. Naive Bayes (NB): NB belongs to the category of Bayesian classifiers which captures the behavior of the data on probability distributions. NB makes an assumption that all the features are conditionally independent [35].
2. Support Vector Machines (SVM): SVM forms a number of hyperplanes and classifies the instances according to the side of the hyperplane to which the instance belongs to [35].
3. Decision Tree (J48): J48 classifies instances through sorting them in a tree which is composed of a hierarchy of nodes. The root node first test the value of the feature and then moves to the child nodes until the label node is reached [35].
4. Conjunctive Rule (CR): CR builds a single conjunctive rule to predict the class labels. It uses the "AND" logical operator to determine correlation of features and classes [35].
5. Voted Perceptron (VP): VP is based on the perceptron algorithm and uses kernel functions to build hyperplanes as decision boundaries [14].

4.4.1 Benchmark Feature Selection Methods

We selected two common feature selection methods to compare the impact of the GP's selected features on the classifiers to the impact of those benchmark methods' features.

1. Chi Square (χ^2) feature evaluation: In statistical analysis methods, χ^2 test is used to measure the in dependency of two events. χ^2 as a feature selection measure the association between the features and classes. A score is given for each feature, according to its χ^2 statistics with respect to the class [13].
2. Information Gain Ratio (IGR) feature evaluation: The features are evaluated by measuring the gain ratio with respect to the class [13]. The gain ratio is the ratio between the total entropy of the features and the intrinsic value.

5 Results and Discussions

Several sets of experiments were performed to test the effectiveness of the proposed GP method. Firstly, GP was run with all the 544 physicochemical properties of the peptides. Secondly, the same GP algorithm was run with the features selected in the terminal nodes of the GP program. The feature selection phase resulted in an average of 5 physicochemical properties for the data set DS_1 and 14 property for data set DS_2. The selected features are fed to the other benchmark classifiers. Moreover, the benchmark feature selection methods (χ^2 and IGR) were used to select features and the top 5 and 14 features from both methods are fed to the same classifiers. In Table 3, the second column gives the performance of the new GP method (annotated as GP). The mean ($\overline{x}$), best and the standard deviation (s) of the 30 runs are reported in the table. The rest of the columns give the results of using the other benchmark classifiers. As these classifiers are deterministic methods only one result is given for each data set. The best performance for each data set is made bold. Table 4 gives the results of using the GP's, χ^2's and IGR 's selected features with the five benchmark classifiers. As the average number of features selected by GP for DS_1 and DS_2 is 5 and 14, respectively, we used the top 5 and 14 features from both χ^2 and IGR to make the comparison. When using the GP's selected features, each of the 30 runs' features are used with each of the classifiers and the average ($\overline{x}$), best and standard deviation are given in Table 4. A statistical T-test (Z-Test) with 0.05 degrees of freedom (95% significance level) is performed to check the significance of the results between the proposed GP method and the methods of comparison. In Tables 3 and 4 the mark $^-$ ($^+$) means the method of comparison is significantly worse (better) than GP, while the mark $^=$ means that there is no significant difference between them. For running GP, the Java-based Evolutionary Computation research system ECJ [23] package was used. All the experiments were executed on a machine with an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, running Ubuntu 4.6 and Java 1.7.0_25 with a total memory of 8GByte .

5.1 GP as a Classifier

As shown in Table 4, the best of GP managed to outperform NB, SVM, J48, CR and VP for data set DS_1. The best of GP is better than these classifiers by 2.37-9.48%, while the mean of the 30 runs is better than SVM, J48, CR and VP

Table 3. Comparison of the performances of GP to benchmark classifiers.

Dataset	GP		NB	SVM	J48	CRt	VP
	Best	$\overline{x} \pm s$					
DS_1	**59.48**	56.62±1.00	57.11^{+}	$56.03^{=}$	$56.03^{=}$	50.00^{-}	53.23^{-}
DS_2	**90.15**	**90.14**±0.01	67.07^{-}	90.13^{-}	87.91^{-}	89.14^{-}	90.13^{-}

by 0.61-6.62%. For data set DS_2, the average and the best of GP outperformed all other classifiers. The results of T-test also show that GP is significantly better than the five classifiers in the data set DS_2. Furthermore, GP is significantly better than CR and VP for data set DS_2. However, there is no significant difference between GP and SVM and J48 in DS_1. The only exception is NB in DS_1, where the performance of NB is slightly better than that of GP, although the best of GP outperforms NB.

5.2 GP as a Feature Selection Method

For each GP run, we used the selected features in the terminal nodes of the GP evolved tree with the other classifiers. The purpose is to test the capability of GP to select the important features in addition to its capability for classification. GP selected an average number of features of 5 for DS_1 and 14 for DS_2 and hence, for both χ^2 and IGR the top 5 and 14 features were used. It can be seen from Table 4 that GP managed to select the features, which achieve better performance with most the classifiers than both χ^2 and IGR on both DS_1 and DS_2. The significance test shows that for DS_1, GP selected features that have a significant better performance than those of IGR when used with all the classifiers. Moreover, it significantly outperformed χ^2 when used with NB, SVM and VP and they were similar when used with J48 and CR. For DS_2, GP features made a significantly better performance when used with NB and equal performance when used with most of the rest of the classifiers. The only exception

Table 4. Comparison of the Performances of *GP*, χ^2 and *IGR* Selected Features

Data set	Classifier	GP		χ^2	T-test	*IGR*	T-test
		Best	$\overline{x} \pm s$	Best		Best	
DS_1	NB	**57.11**	**53.87**±2.34	50.96	−	52.50	−
	SVM	**60.56**	**54.71**±2.19	52.90	−	52.04	−
	J48	**57.11**	**54.60**±1.61	54.19	=	52.04	−
	CR	**57.32**	**53.00**±2.14	52.68	=	50.04	−
	VP	**60.56**	**54.35**±2.32	52.04	−	52.04	−
DS_2	NB	**85.77**	**84.59**±0.55	71.24	−	71.05	−
	SVM	90.15	90.15±0.0	90.14	=	**90.22**	=
	J48	**90.34**	89.95±0.29	90.06	+	90.13	+
	CR	90.15	90.15±0.0	**90.22**	=	**90.22**	=
	VP	**90.22**	90.15±0.20	**90.22**	=	**90.22**	=

here is with J48 which has a slightly better accuracy with χ^2 and IGR more than the average of GP. This is perhaps because J48 also uses IGR to further select features, and therefore might be biased to IGR.

6 Conclusions and Future Work

The objective of this paper was to investigate the performance of GP capability to reduce the number of redundant properties with preserving the maximum accuracy for measuring peptide detectability. This goal was successfully achieved by developing a GP system which selects features and at the same time performs detection. The proposed method works by maximising the classification accuracy and minimising the number selected features in the terminal nodes of the GP tree, and therefore, the system is a multi-objective system. The new method is tested against five other classifiers namely, NB, SVM, J48, CR, VP. Moreover, in order to compare the feature selection capability of the proposed method, it is tested against two well known feature selection methods, namely, χ^2 and IGR. The results show that GP outperformed most of these state of art feature selection and classification algorithms.

There are many other investigations that need to be done in the future. Firstly, the peptide data sets are mostly characterized by being unbalanced data which means that peptides in one class (mostly flyer's class) is much less than the peptides in the other class. This makes the classifiers biased towards the majority class, and hence, the specificity rate will be much higher than the sensitivity rate. This means that the overall classification accuracy is not the only evaluation criteria that should be used for measuring the peptide detectability and the imbalance problem should be taken into account. The use of GP to solve the imbalance problem will be the first future direction. Another future direction is the verification of the candidate biomarkers detection in MS data through the linkage of the detectability of the biomarkers in the mass spectrometer. Finally, the absolute quantification of proteins using GP through peptide detection will be performed in the future.

References

1. Abbatiello, S., Mani, D., Keshishian, H., Carr, S.: Automated Detection of Inaccurate and Imprecise Transitions in Peptide Quantification by Multiple Reaction Monitoring Mass Spectrometry. Clinical Chemistry 56, 291–305 (2010)
2. Ahmed, S., Zhang, M., Peng, L.: Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach. In: Vanneschi, L., Bush, W.S., Giacobini, M. (eds.) EvoBIO 2013. LNCS, vol. 7833, pp. 43–55. Springer, Heidelberg (2013)
3. Ahmed, S., Zhang, M., Peng, L.: Genetic programming for biomarker detection in mass spectrometry data. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS, vol. 7691, pp. 266–278. Springer, Heidelberg (2012)
4. Ahmed, S., Zhang, M., Peng, L.: Enhanced feature selection for biomarker discovery in LC-MS data using GP. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation, pp. 584–591 (2013)

5. Augusto, D.A., Barbosa, H.J.C., Ebecken, N.F.F.: Coevolutionary multi-population genetic programming for data classification. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010, pp. 933–940. ACM, New York (2010)
6. Bhowan, U., Johnston, M., Zhang, M.: Developing new fitness functions in genetic programming for classification with unbalanced data. IEEE Transactions on Systems, Man, and Cybernetics, Part B 42(2), 406–421 (2012)
7. Bhowan, U., Johnston, M., Zhang, M., Yao, X.: Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Trans. Evolutionary Computation 17(3), 368–386 (2013)
8. Bhowan, U., Zhang, M., Johnston, M.: Genetic programming for classification with unbalanced data. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 1–13. Springer, Heidelberg (2010)
9. Cho, C.-K.J., Drabovich, A.P., Batruch, I., Diamandis, E.P.: Verification of a biomarker discovery approach for detection of Down syndrome in amniotic fluid via multiplex selected reaction monitoring (SRM) assay. J Proteomics, 2052–2059 (2011)
10. Desiere, F., Deutsch, E.W., King, N.L., Nesvizhskii, A.I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S.N., Aebersold, R.: The PeptideAtlas project. Nucleic Acids Research 34(suppl 1), D655–D658 (2006)
11. Domon, B., Aebersold, R.: Options and considerations when selecting a quantitative proteomics strategy. Nat. Biotechnology 28, 710–721 (2010)
12. Eyers, C.E., Lawless, C., Wedge, D.C., Lau, K.W., Gaskell, S.J., Hubbard, S.J.: CONSeQuence: prediction of reference peptides for absolute quantitative proteomics using consensus machine learning approaches. Molecular & Cellular Proteomics 10(11) (2011)
13. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3, 1289–1305 (2003)
14. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Mach. Learn. 37(3), 277–296 (1999)
15. Gay, S., Binz, P.-A., Hochstrasser, D.F., Appel, R.D.: Peptide mass fingerprinting peak intensity prediction: Extracting knowledge from spectra. PROTEOMICS 2(10), 1374–1391 (2002)
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. In: SIGKDD Explorer Newsletter, pp. 10–18 (2009)
17. He, H., Garcia, E.A.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
18. Huttenhain, R., Malmstrom, J., Picotti, P., Aebersold, R.: Perspectives of targeted mass spectrometry for protein biomarker verification. Curr. Opin. Chem. Biol. 13, 518–525 (2009)
19. Kawashima, S., Kanehisa, M.: AAindex: Amino Acid index database. Nucleic Acids Research 28(1), 374 (2000)
20. Koenig, T., Menze, B.H., Kirchner, M., Monigatti, F., Parker, K.C., Patterson, T., Steen, J.J., Hamprecht, F.A., Steen, H.: Robust Prediction of the MASCOT Score for an Improved Quality Assessment in Mass Spectrometric Proteomics. Journal of Proteome Research 7(9), 3708–3717 (2008)
21. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
22. Koza, J.R.: Introduction to genetic programming: tutorial. In: GECCO (Companion), pp. 2299–2338 (2008)

23. Luke, S.: Essentials of Metaheuristics. In: Lulu, 2nd edn. (2013), http://cs.gmu.edu/$\sim$sean/book/metaheuristics/
24. Mallick, P., Schirle, M., Chen, S., Flory, M., Lee, H., Martin, D., Ranish, J., Raught, B., Schmitt, R., Werner, T., Kuster, B., Aebersold, R.: Computational Prediction of Proteotypic Peptides for Quantitative Proteomics. Nat. Biotechnol. 25(1), 125–131 (2007)
25. Muni, D., Pal, N., Das, J.: Genetic programming for simultaneous feature selection and classifier design. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(1), 106–117 (2006)
26. Neshatian, K., Zhang, M.: Unsupervised Elimination of Redundant Features Using Genetic Programming. In: Proceedings of 22nd Australasian Conference on Artificial Intelligence, pp. 432–442 (2009)
27. Neshatian, K., Zhang, M.: Using genetic programming for context-sensitive feature scoring in classification problems. Connect. Sci. 23(3), 183–207 (2011)
28. Neshatian, K., Zhang, M.: Improving relevance measures using genetic programming. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds.) EuroGP 2012. LNCS, vol. 7244, pp. 97–108. Springer, Heidelberg (2012)
29. Pruitt, K.D., Tatusova, T., Maglott, D.R.: NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research 33(suppl 1), D501–D504 (2005)
30. Smart, W., Zhang, M.: Using Genetic Programming for Multiclass Classification by Simultaneously Solving Component Binary Classification Problems. In: Keijzer, M., Tettamanzi, A.G.B., Collet, P., van Hemert, J., Tomassini, M. (eds.) EuroGP 2005. LNCS, vol. 3447, pp. 227–239. Springer, Heidelberg (2005)
31. Tang, H., Arnold, R.J., Alves, P., Xun, Z., Clemmer, D.E., Novotny, M.V., Reilly, J.P., Radivojac, P.: A computational approach toward label-free protein quantification using predicted peptide detectability. Bioinformatics 22(14), e481–e488 (2006)
32. Timm, W., Scherbart, A., Bocker, S., Kohlbacher, O., Nattkemper, T.: Peak intensity prediction in MALDI-TOF mass spectrometry: A machine learning study to support quantitative proteomics. BMC Bioinformatics 9(1), 443 (2008)
33. Vaidyanathan, S., Broadhurst, D.I., Kell, D.B., Goodacre, R.: Explanatory Optimization of Protein Mass Spectrometry via Genetic Search. Analytical Chemistry 75(23), 6679–6686 (2003)
34. Wedge, D.C., Gaskell, S.J., Hubbard, S.J., Kell, D.B., Lau, K.W., Eyers, C.: Peptide detectability following ESI mass spectrometry: prediction using genetic programming. In: GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, vol. 2, pp. 2219–2225 (2007)
35. : In: Witten, I.H., Frank, E. (eds.) Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers Inc., San Francisco (2005)