Information Sciences 222 (2013) 229–246

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Comparison of metaheuristic strategies for peakbin selection in proteomic mass spectrometry data

Miguel García-Torres [a,*], Rubén Armañanzas [b], Concha Bielza [b], Pedro Larrañaga [b]

[a] Área de Lenguajes y Sistemas Informáticos. Universidad Pablo de Olavide. Ctra de Utrera, km. 1, 41013 Sevilla, Spain
[b] Computational Intelligence Group, Universidad Politécnica de Madrid. 28660, Boadilla del Monte, Madrid, Spain

ARTICLE INFO

Article history:
Received 22 January 2010
Received in revised form 11 December 2010
Accepted 27 December 2010
Available online 7 January 2011

Keywords:
Metaheuristics
Feature subset selection
Mass spectrometry

ABSTRACT

Mass spectrometry (MS) data provide a promising strategy for biomarker discovery. For this purpose, the detection of relevant peakbins in MS data is currently under intense research. Data from mass spectrometry are challenging to analyze because of their high dimensionality and the generally low number of samples available. To tackle this problem, the scientific community is becoming increasingly interested in applying feature subset selection techniques based on specialized machine learning algorithms. In this paper, we present a performance comparison of some metaheuristics: best first (BF), genetic algorithm (GA), scatter search (SS) and variable neighborhood search (VNS). Up to now, all the algorithms, except for GA, have been first applied to detect relevant peakbins in MS data. All these metaheuristic searches are embedded in two different filter and wrapper schemes coupled with Naive Bayes and SVM classifiers.

1. Introduction

In recent years, mass spectrometry (MS) has become increasingly popular for discovering biomarkers of diseases such as cancer [1,60,62], myocardial infarction [83], etc. Comparing protein expression levels in case samples with control groups may lead to the identification of important biomarkers that can predict the degrees of malignancy in tumors, provide valuable information about the efficacy of specific anti-cancer treatments or help to identify new molecular targets for innovative therapeutic strategies [44].

To address this problem, *matrix-assisted laser desorption and ionization* (MALDI) [49] and *surface-enhanced laser desorption/ionization* (SELDI) [43] ion sources coupled with a *time-of-flight* (TOF) detector are two of the technologies most commonly used to obtain proteomic profiles. These technologies, widely called MALDI-TOF and SELDI-TOF, respectively, measure the relative abundance of ionized peptides with respect to their mass-to-charge (m/z) ratios.

Both technologies produce datasets that are known as MALDI-TOF and SELDI-TOF mass spectra. Such spectra consist of tens of thousands of m/z ratios per patient (spectrum), where each m/z value of the spectrum approximately reflects the abundance of peptides of a set mass [9].

The discovery of biomarkers in mass spectrometry datasets is a recent bioinformatic problem that aims to identify proteins/peptides (biomarkers) that are expressed differently in different disease states. One of the biggest challenges is to identify biomarkers from ideally continuous mass spectrometry data, because, generally, a specific value of m/z cannot be directly mapped to a specific protein, since mass is not sufficient to identify a protein [14]. To determine the exact species

* Corresponding author.
E-mail addresses: mgarciat@upo.es (M. García-Torres), r.armananzas@upm.es (R. Armañanzas), mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (P. Larrañaga).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.12.013

of protein molecule that caused a peak, additional experimentation must be performed. This is beyond the scope of our work. We will use peakbins in the data to identify biomarkers after a binning step as in [27,82].

MS datasets are typically high-dimensional (several thousand features) with a relatively small number of samples (a few hundred). This precludes the use of exhaustive or greedy search strategies for feature selection in favor of stochastic search algorithms, like metaheuristics. Metaheuristics are general strategies which guide and modify other heuristics to search feasible solutions in optimization problems [30].

The scientific community is becoming increasingly interested in applying machine learning techniques to mass spectra classification as shown in recent publications that use random forest [38], probabilistic neural networks [10], compare the results of different classifiers [27,56] or include clustering techniques in the study [17]. To address the problem of biomarker discovery, the following search algorithms have been studied: support vector machine-recursive feature elimination [20,33], ant colony optimization [67,68], genetic algorithms [69] and gradient based leave-one-out gene selection [56].

In this paper we compare several metaheuristics – best first, genetic algorithm, scatter search and variable neighborhood search – to study their performance in the biomarker discovery problem. These strategies have proved to be competitive in feature subset selection problems, although some have not yet been applied to the biomarker discovery problem. To do this, we apply a data analysis pipeline that imitates a real scenario. In this context, we compare the results achieved by each algorithm and the peakbins found in our experiments in order to propose a set of biomarker candidates. Finally, we analyze the occurrence of the peakbins reported by other authors.

This paper is organized as follows. Section 2 describes the feature selection problem we want to solve. Section 3 introduces the metaheuristics and the different strategies used. The experimentation pipeline is explained in Section 4. Then, Section 5 presents the characteristics of each studied MS dataset. Finally, the results are presented in Section 6 and the conclusions in Section 7.

2. Feature selection

Let $\mathbb{X} = \{X_j : j = 1, \ldots, n\}$ be a set of features that characterize a set of input examples $\mathcal{E}$, and $J(S)$ be a quality measure of a subset $S \subseteq \mathbb{X}$ defined as $J : S \subseteq \mathbb{X} \to \mathbb{R}$. The associated optimization problem consists of finding the subset S with the highest quality measure.

Feature subset selection strategies are essentially divided into wrapper, filter and embedded methods [32]. Wrappers use the learner as a black box to score the subsets of features according to their predictive power. Filters select subsets of features as a preprocessing step. Finally embedded methods perform feature selection while building the model. In this paper, we use wrapper and filter approaches, which are explained in further detail in the following.

Filter methods. This approach assesses each subset according to intrinsic properties of the data. The advantages of these methods are that they are computationally simple and fast and they easily scale to high-dimensional datasets. A disadvantage is that they ignore the interaction with the classifier, which may lead to worse classification performance. Since they are independent of the learning algorithm, feature selection needs to be performed only once for a given training dataset. In this study we use the correlation-based feature subset selection [35] (CFS), which evaluates the quality of a subset of features taking into account the correlation of individual features for predicting the class label by means of the level of feature inter-correlation. The goodness of a feature subset S containing k features is given by

$$J(S) = \frac{k\langle r_{cf}\rangle}{\sqrt{k + k(k+1)\langle r_{ff}\rangle}},$$

where $\langle r_{cf}\rangle$ is the mean feature-class correlation ($f \in S$) and $\langle r_{ff}\rangle$ is the average inter-correlation between each pair of features.

Wrapper methods. In this approach, the quality of feature subsets for classification is defined with respect to the induction algorithms. The main advantage is that they include the interaction between feature subset and model selection, and have the ability to take into account feature dependencies. However, they have a higher risk of overfitting than filters and are computationally expensive. As induction algorithms, we used the Naive Bayes and the support vector machine (SVM) with linear decision surface.

- Naive Bayes [46] (NB). This is a probabilistic classifier based on Bayes' theorem. It assumes that the predictor variables are conditionally independent given the value of the class. Although it is the simplest form of Bayesian network, it has been observed that its classification accuracy may be high on datasets where there are strong dependencies among features.
- Support vector machine [78] (SVM). This classifier constructs an n-dimensional hyperplane that optimally separates the data into two categories. To find such hyperplane, SVM solves an optimization problem which finds the separating hyperplane that optimizes a weighted combination of the misclassification rate and the distance of the decision boundary to any sample vector.

2.1. Robustness of feature subsets

The robustness or stability of feature subset selection strategies is a topic of recent interest that aims to measure the sensitivity of a feature selection algorithm to variations in the dataset. This issue is important specially in high dimensional knowledge discovery domains where samples are small [61], e. g. proteomics.

It is worth noting that a stability measure provides no information about the performance of such features but it enhances the confidence in the analysis results. As pointed out in [52], highly correlated features, small sample size, etc., may cause a large variation in the solutions.

To quantify the robustness of a feature selection method, a measure of similarity between two sets of features is needed. Previous works have proposed measures based on the Hamming distance [21], consistency [54,75], Tanimoto's distance [42,47,48], entropy [52,53], the Jaccard index [71] and Pearson's correlation coefficient [41].

In this paper we will use a modification of the consistency index ($\mathcal{I}_C$) [54] proposed in [3] to deal with the difficulty of comparing subsets of different sizes and the Jaccard index ($\mathcal{I}_J$) [71].

Let A and B be subsets of features such that $A,\ B \subseteq \mathbb{X}$. Let $n = |\mathbb{X}|$ denote the cardinality of $\mathbb{X}$ and let $|A| = k_A$, $|B| = k_B$ and $r = |A \cap B|$ be the cardinalities of A, B and the intersection of the two sets, respectively.

The consistency index is originally defined for the case in which $k = k_A = k_B$:

$$\mathcal{I}_C(A,B) = \frac{rn - k^2}{k(n-k)}. \tag{1}$$

By selecting the largest size such that $k = \max\{k_A, k_B\}$, this measure may be adapted to handle different cardinality sets. The Jaccard index is defined as the size of the intersection divided by the size of the union of the sets

$$\mathcal{I}_J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2}$$

Given a set of solutions $\mathcal{S} = \{S_1, \ldots, S_m\}$, the approach for estimating the stability Σ among this set of solutions consists of averaging the pairwise $\mathcal{I}_C$ (Σ_C) or $\mathcal{I}_J$ (Σ_J) similarities

$$\Sigma(\mathcal{S}) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \mathcal{I}(S_i, S_j).$$

In both cases higher values correspond to more stable subsets.

3. Metaheuristics

This section presents metaheuristics and their key characteristics in the feature subset selection context. It then goes onto describe the metaheuristics used in the comparison – best first, genetic algorithm, scatter search, variable neighborhood search.

Metaheuristics are a family of approximate optimization techniques that provide satisfactory solutions in a reasonable time, allowing large and complex problems to be tackled. Unlike exact algorithms, metaheuristics do not guarantee the optimality of the output solutions.

Metaheuristics are general-purpose algorithms that can be applied to solve almost any optimization problem. An optimization problem may be defined by the couple (S,J), where S represents the set of feasible solutions and $J : S \to \mathbb{R}$ the objective function which corresponds to performance measures. This couple defines a relation between any pair of solutions in the search space. To solve the problem, we have to find a solution that optimizes J.

Since there is no guarantee of the distance of the solution from the optimal solution, the question is when to use metaheuristics. In general, metaheuristics are suitable for solving hard and/or large-size instances of an optimization problem for which there is no efficient exact algorithm available.

In feature subset selection, finding the optimal solution is known to be NP-hard [2] and requires examining all 2^n possible subsets of the feature set, which quickly becomes computationally intractable. Feature selection is a combinatorial optimization problem in which a solution corresponds to a set of features and the objective function to the score associated with the solution according to a measure ($J(S)$). In this context the use of heuristics and metaheuristics seems to be appropriate.

3.1. Best first

Best first [28,70] is a search method that explores a graph by expanding the most promising node according to a heuristic evaluation function. This strategy builds a tree to perform the search. At each level of the tree, it generates all successors of the nodes and sorts them according to the evaluation function.

Fig. 1 shows a search scheme. It keeps a *closed list* ($\mathcal{C}$) of nodes that have been expanded, and an *open list* ($\mathcal{O}$) of nodes that have been generated but not yet expanded. At each iteration of the algorithm, it expands the most promising node on the *open list* (the node n for which the evaluation function $f(n)$ is maximum or minimum). When a node is expanded, it is moved from the *open list* to the *closed list*, and its children are generated and added to the *open list*. The search stops when the stopping criterion is reached.

Best first is a widely studied strategy in many optimization problems like feature subset selection ([51,80]). The version of the best first we use is called beam search, which reduces the memory requirement limiting the number of best partial solutions that are kept as candidates to b (beam width). The larger b is, the closer the search is to exhaustive search and for $b = 1$, the search is identical to a greedy forward search. The beam width is an important parameter and controls the trade-off be-

Procedure *Best first*
begin
1: Initialize $(\mathcal{O}, \mathcal{C})$;
2: **repeat**
3: *SelectNode* $(n, \mathcal{O},)$;
4: *ExpandNode* $(n, \mathcal{O}, \mathcal{C})$;
5: *GenerateChildNodes* $(n, \mathcal{O})$;
6: **until** $(StoppingCriterion)$
end

Fig. 1. Best first pseudocode. $\mathcal{O}$ is the *open list*, $\mathcal{C}$ the *closed list* and n a node.

tween the search speed and exhaustivity. Although there is some work on determining a good beam size [45,65], it is still an open question since it depends on the problem. In order to increase the search space and taking into account the high dimension of the data, we set $b = 4$.

3.2. Genetic algorithm

A genetic algorithm (GA), which was first presented by Holland [39], is an evolutionary population-based strategy that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. This strategy is widely used in many optimization problems [12,50], including feature subset selection.

Each solution represents a chromosome (also called individual) and is composed of genes, which are digits in the solution. The alleles are the possible values a gene can take. Binary encoding, which is one of the most used methods, consists of individuals represented as binary arrays.

As Fig. 2 shows, the strategy starts by generating a population of individuals. Promising individuals are then selected to generate new ones using genetic operators such as crossover and mutation. The purpose of crossover is to combine parents to generate new offsprings. Mutation produces some changes on a single individual and introduces diversity in the population. The process continues across several generations until a stoppping criterion is reached. So as to select the best individuals for reproductive opportunities (apply crossover operator), GAs use a fitness function to measure the relative performance of each individual with respect to the current population.

Due to their popularity GAs have been applied to feature selection in many works ([7,8,40]) as well as to biomarker discovery [69]. In this research we use the simple genetic algorithm [31], which uses a linear transformation of the objective funtion as the fitness method. The efficiency of a GA is greatly dependent on its tuning parameters. We set the crossover probability to 0.6 and the mutation probability to 0.001 following the recommendations of [81]. In order to set the population size and the number of iterations, we have to take into account that in problems with a very large solution space the population size must be large enough to obtain a representative sample of the solution space. Furthermore, a very small population may result in premature convergence, whereas a very large population may result in a slow convergence rate. Since both values are dependent on the problem, we conducted several experiments with different values for the population size and number of iterations. After analysing convergence and computational time, we fixed the population at 500 individuals and the number of iterations was set to 250 in the filter scheme and to 20 in the wrapper case.

3.3. Scatter search

Scatter search [16,55] is an evolutionary population-based metaheuristic that was first introduced in the 1970s as an extension of the formulation for combining decision rules and problem constraints [29].

Procedure *Genetic algorithm*
begin
1: *GeneratePopulation* (P_t);
2: *EvaluatePopulation* (P_t);
3: **repeat**
4: *SelectParents* (P_t, P'_t);
5: *ApplyOperators* (P'_t, P''_t);
6: *EvaluatePopulation* (P''_t);
7: *UpdatePopulation* (P''_t, P_t);
8: **until** $(StoppingCriterion)$
end

Fig. 2. Genetic algorithm pseudocode. P_t is the current population at generation t, P'_t is the set of solutions for combination and P''_t the new solutions generated.

```
Procedure Scatter search
begin
1:  GeneratePopulation (InitPop);
2:  GenerateReferenceSet (RefSet);
3:  repeat
4:      repeat
5:          SelectSubset (Subset);
6:          CombinationMethod (Subset, CurSol);
7:          ImprovementMethod (CurSol, ImpSol);
8:      until (StoppingCriterion_1)
9:      UpdateReferenceSet (RefSet);
10: until (StoppingCriterion_2)
end
```

Fig. 3. Scatter search pseudocode. The *InitPop* is the initial population, *RefSet* the reference set, *Subset* the subset of solutions for combination, *CurSol* the subsets generated after the combination and *ImpSol* the solutions generated as result of applying the improvement method.

The method starts with a population of solutions from which a moderate-sized set, the *reference set* (*RefSet*), is selected to evolve. The evolution is based on intensification and diversification strategies to take advantage of features associated with good solutions and to be able to escape from local optima.

The solutions of the *RefSet* are combined to generate new ones and then a local search is applied to the resulting solutions. The *RefSet* is then updated to incorporate solutions taking into account quality and diversity. These steps are repeated until a stopping criterion is met.

Unlike other evolutionary strategies, such as genetic algorithms, the combination of solutions is guided and not random, and the subset that evolves is smaller in size to usual populations.

Fig. 3 describes the pseudocode of the scatter search. The algorithm starts by generating a population of solutions. This population is composed of a large set of disperse solutions that are improved by the *improvement method*. Then, a representative set of solutions is selected to generate the *reference set*. This set consists of solutions with the best objective function values and the most diverse values from the population. Subsets from the *RefSet* are systematically selected for combination to generate new solutions. The combination method tries to combine the good features of the solutions to get new solutions that are unlike the solutions already in the *RefSet*. Then an improvement method is applied to every solution generated. Finally the *RefSet* is updated taking into account the intensification and diversification criteria.

Scatter search has been succesfully applied to the feature subset selection problem [24,23]. This research used the implementation proposed in [23]. Due to the high dimensionality of the data and following the general recommendations on the strategy, the population size was set to $|InitPop| = 100$[1] and the reference set to $|RefSet| = |InitPop|/2$. The search stops if no improvement of the best solution is found after two iterations. The combination method is a very time-consuming algorithm since it consists of intensive searches around local optima. In general, this leads to convergence after a small number of iterations.

3.4. Variable neighborhood search

Variable neighborhood search (VNS) [36,37] is a metaheuristic based on systematic changes of neighborhood during the search. This strategy is based on three facts:

- A local optimum with respect to one neighborhood structure is not necessarily a local optimum in another one.
- A global optimum is a local optimum with respect to all possible neighborhood structures.
- For many problems local optima with respect to one or several neighborhood structures are relatively close to each other.

The last observation is empirical and implies that a local optimum often provides some information about the global optimum.

Let $\mathcal{N}_k,\ k = 1, \ldots, k_{max}$ be a finite set of neighborhood structures, and $\mathcal{N}_k(S)$ be the set of solutions in the kth neighborhood of a solution S. Usually, these neighborhoods are nested. For this reason to move from the original neighborhood ($k = 1$) to the kth neighborhood, the search may move by repeating a move in the original neighborhood k times. This means that $\mathcal{N}_{k+1}(S) = \mathcal{N}(\mathcal{N}_k(S))$, where $\mathcal{N}_1(S) = \mathcal{N}(S)$ is the original neighborhood.

The pseudocode of the VNS is described in Fig. 4. According to the scheme, a finite number of neighborhood structures are first defined around a solution. Then an initial solution is generated by applying a local search. This solution is shaken to

[1] For HCC, the convergence speed in the wrapper scheme with SVM is low. In this case $|InitPop| = 10$.

generate a random solution within the first neighborhood $\mathcal{N}_1(S)$ of S, and a local search is applied to obtain a local optimum S''. If the local optimum does not improve the current best solution S, then the procedure is iterated using the next neighborhood. If in the last neighborhood $\mathcal{N}_{k_{max}}(S)$ there is no improvemente in S, then the search begins from $\mathcal{N}_1(S)$ until a stopping criterion is reached. If S'' improves S, then the search is refocused around $S \leftarrow S''$, and it begins again with the first neighborhood.

Recently, VNS has been applied to the feature selection problem [25,59]. Based on the algorithm proposed in [25], the implementation is modified as follows to handle high dimensional data in a better way.

The initial solution randomly selects the features based on a probability associated with each attribute. To calculate this probability, we use the symmetrical uncertainty measure between each attribute S_i, $i = 1,\ldots,k$ and the class C:

$$SU(S_i, C) = 2\left[\frac{IG(S_i|C)}{H(S_i) + H(C)}\right],$$

where $IG(S_i|C)$ is called information gain and measures the amount of information gained about C after observing S_i, and $H(S_i)$ and $H(C)$ are the entropy of features S_i and C, respectively. A value of 1 means that the attribute is completely correlated to the class and a value of 0 means that it is not correlated at all. The shaking method changes the state of k features at each iteration, where k takes values ranging from k_{min} to k_{max}. The local search removes the attributes without which the solution improves from the solution. Backward elimination is very time-consuming; therefore only a small number of features, called subset of feature candidates, are explored to improve the efficiency. A feature is a candidate for removal if it is a redundant feature (it is more correlated to the solution than to the class). The stopping criterion is to reach the maximum number of iterations or when the search converges to a local optimum. As the search space grows exponentially with k; we set $k_{min} = 1$, $k_{max} = 10$ and limited the number of iterations to 10.

4. Mass spectrometry data

Throughout this section we present the characterization of the MS spectra (Section 4.1) and then introduce the preprocessing techniques (Section 4.2) indicating the approach used.

The spectra produced by SELDI and MALDI consist of a vector of counts, where each count corresponds to the number of ions detected during a short fixed interval of time. The count of the number of ions is usually called *intensity* and peaks in the intensity represent the abundance of proteins or peptides that are present in the sample.

In order to characterize the spectra and extract the peakbins, we apply a data analysis pipeline (DAP). The DAP refers to the design of the experiments. Briefly, it starts from the raw MS data, to which it first applies a set of preprocessing techniques. Then the preprocessed data are mined.

Even though the development of preprocesing methods has become an active area of research [6,58,66,77], there is no standard preprocessing pipeline. Some tasks are widely accepted and have become a standard of application. These tasks are (i) baseline removal or correction, (ii) normalization, (iii) signal smoothing, (iv) peak detection and (v) peak assembly and quantification. The preprocessing pipeline and algorithms used in this work are explained in more detail in [3].

4.1. Data analysis pipeline

The structure of the DAP is shown in Fig. 5. First, the DAP workflow applies the correction of the baseline to all spectra. This effect could be integrated into the preprocessing engine but the correction is independent for each spectrum. It can therefore be applied as an independent task.

```
Procedure Variable neighborhood search
begin
1:  Initialize (N_k, S), k = 1, ..., k_max;
2:      repeat
3:          k ← 1;
4:          repeat
5:              ShakeMethod (S, S');
6:              LocalSearch (S', S'');
7:              Move (S'', S);
8:          until (k = k_max)
9:      until (StoppingCriterion)
end
```

Fig. 4. Variable neighborhood search pseudocode. S is a feasible solution and $\mathcal{N}_k$ is the kth neighborhood of S.

Once the data is baseline corrected, the prepocessing engine and data mining task will be applied to each single train/test split generated from the evaluation model. To avoid bias in the results and imitate a real scenario, where new cases would arrive at the end of the analysis, test data will be used for validation purpose only.

As we can see in Fig. 5, we used k-fold cross-validation as the evaluation schema. This randomly splits the data $\mathcal{D}$ into k mutually exclusive subsets (folds) $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$. This estimator generates k train/test split pairs; each time $t \in \{1,2,\ldots,k\}$, $\mathcal{D} \setminus \mathcal{D}_t$ is used for training and $\mathcal{D}_t$ for testing purposes. In order to reduce variability, DAP runs k-fold cross-validation multiple times.

Given a train/test split pair, the training dataset is used to characterize the spectra. Peakbins discovered and assembled in the training dataset are quantified in the test set. After this we will proceed as usual in data mining tasks. For classification, the classifier is induced during training; and the model is applied to classify cases from the test dataset. In this case of feature selection for classification, the most useful features (peakbins) have to be searched before inducing the classifier (peakbin selection). In this case, only selected peakbins will be used for classification.

Distinct subsets of data lead to different peakbin range values for the same peak (assuming a peak corresponds to specific proteins or peptides). To compare peakbins from different training datasets we need a criterion to determine whether or not two or more peakbins refer to the same peak. In our case we will consider that all overlapped peakbins refer to the same peak.

4.2. Preprocessing tasks

These spectra output by a mass spectrometer are affected by errors and noise [11] and thus require low-level preprocessing to correct intensity and m/z values. This step may also be used to reduce the dimensional complexity of the spectra, although this requires care. The use of inadequate methods can introduce additional bias or additional variance into the measurements, making it difficult to reach consistent biological conclusions [4,76].

The true signal (spectrum with neither noise, nor error) can be modeled as a sum of independent, possibly overlapping shapes, each corresponding to a single protein. Because of the unknown characterization of the individual components of a spectrum, the shapes of the peaks should be estimated empirically.

Note that due to the non-uniformization of the original raw data, spectra from the same datasets do not share the same m/z axis. For this reason, before applying the preprocessing pipeline, spectra were binned. The basic idea of binning is to scan the spectra and group adjacent values of the data into ranges called bins.

4.2.1. Baseline removal

Intensity values are always amplified exhibiting a baseline intensity level usually attributed either to clusters of ionized matrix molecules hitting the detector during early portions of the experiment or to detector overload. This noise varies across the m/z axis, and the effect tends to decrease as the intensity value increases.

To remove this chemical noise we used the top-hat morphological operator (THMO) [74], which is a nonlinear positive low pass operator. There has been no comparison of the different techniques proposed, and none of them has reported sig-

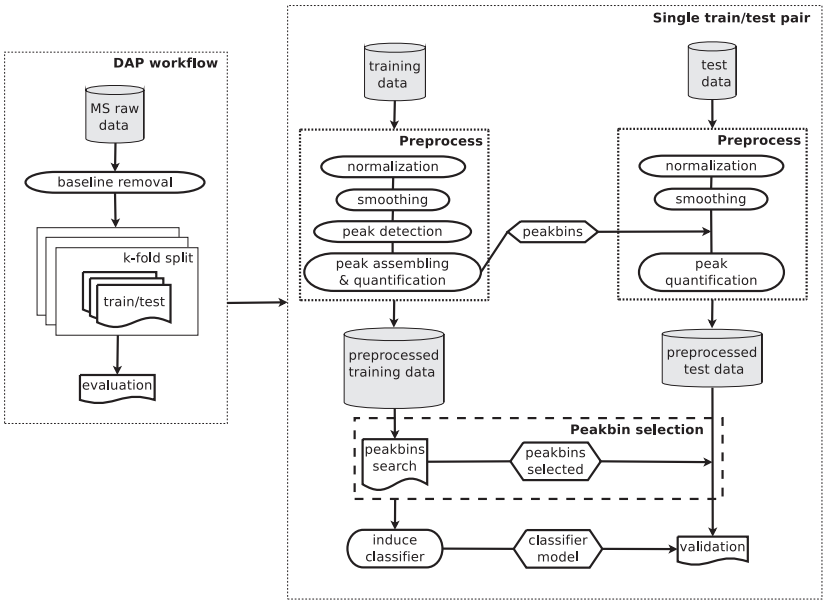

Fig. 5. Data analysis pipeline.

nificant differences. In this context, the main advantages of THMO are that it is a low time-consuming algorithm and is widely used in the image analysis domain.

4.2.2. Normalization

Different samples are usually measured on different scales so that mass spectra are not comparable. The purpose of spectrum normalization is to correct systematic differences in the total amount of protein desorbed and ionized from the sample plate.

Although normalization with respect to the total ion current (TIC) is becoming standard practice, a recent study [57] that compares several normalization methods concludes that the use of local normalization methods achieves better results than global methods like TIC. Furthermore median values have proven to be more robust than averages as scale factors against possible outlying peaks [18]. Taking into account these studies, the normalization used here combines both approaches: it uses local estimators over m/z windows with rescaling to the median value of the TIC. The window width for MALDI/SELDI spectra was set to 200 m/z units.

4.2.3. Signal smoothing

The detection instruments generate electrical noise (also called white noise) that perturbs the original signal changing the intensity randomly. Smoothing reduces the noise level across the whole spectrum by removing low resolution peaks that represent noise perturbations. This was done using wavelet algorithms as proposed in [15].

4.2.4. Peak detection

Peak detection refers to the process of identifying peaks (m/z values) that correspond to specific proteins or peptides striking the detector. In general, it is accepted that a peak candidate must meet the following conditions:

- The peak must have higher intensity than its neighbors.
- The peak must be above a chosen threshold.
- The peak must have an associated signal to noise ratio (SNR) higher than a threshold.

The approach used here is based on the algorithm proposed in [64], except as regards how the algorithm estimates the SNR of a signal window. Unlike the original algorithm, the SNR is estimated as the ratio between the point's height and the median absolute deviation (MAD) in the window under consideration [73].

4.2.5. Peakbin assembly and quantification

The effect of measurement error, called *mass error* effect [72], arises when the m/z value of a peak can differ from one spectrum to another, even if both spectra belong to the same sample. Peak alignment corrects this error by shifting the signal for each spectrum until all peaks correspond to the same biological molecule match.

We used the Pearson linear correlation coefficient to group peaks that are close on the m/z axis across different spectra if their intensity levels are similar. This is done to avoid hiding isotopic formations or very close compounds. Finally the output of the preprocessing task is a list of peakbins.

5. Datasets

Three publicly available datasets were used to compare the performance of the metaheuristic strategies. Two datasets come from a SELDI spectrometer and the last one from a MALDI spectrometer. Spectra were binned to a resolution of 0.025. The mean of the intensities within each bin was used as the protein expression value [79], and a 0 value was assigned if no values were available for an interval. The selected datasets were:

- *Ovarian cancer profiling* (OVA) [61]. This is one of the most analysed SELDI-TOF datasets. The objective is to discriminate between ovarian cancer patients and the control group. We used the *high resolution* MS data, which consist of 200 cases that are made up of 121 ovarian cancer samples and 79 control samples. After binning, each spetrum contains 45,200 values with m/z ranging from 700.116 to 12,000.
- *Detection of drug-induced toxicity* (TOX) [63]. The aim of this study is to be able to detect drug-induced toxicity using a serum proteomic pattern diagnostic device based on SELDI-TOF technology. Only specimens for which the diagnosis is definitely positive or definitely negative were picked from the original sample of 203 specimens. Thus our dataset consisted of 62 samples (34 control group and 28 induced cardiotoxicity samples). A total of 45,200 m/z values, ranging from 799.115 to 12,000, describe each spectrum.
- *Hepatocellular carcinoma* (HCC) [68]. The goal of this research is to distinguish between HCC patients from healthy individuals through MALDI-TOF analysis. The 150 spectra, which consist of 78 patients with HCC and 72 control samples, have 36,802 m/z values across the interval from 700.725 to 9,999.975.

Table 1 shows the general basic information related to each dataset. The ID associated with each dataset is shown in the first column. The next columns contain the total number of samples, followed by the number of samples belonging to the

control group (c) and the patient group (a). The number of bins, and m/z value ranges are listed in columns 4 and 5 respectively. Finally the reference to the original work is given.

6. Results and discussion

This section presents the experiments performed and discusses the results. We compare the performance of the metaheuristics and then analyse the set of peakbins found. For the comparison, we present a DAP workflow that imitates a real scenario. In order to clarify the results, we divide the experimentation into three parts:

(a) In the first part (Section 6.1), we study the effects of preprocessing on binned data using the DAP workflow described previously.
(b) Then (Section 6.2), we compare the performance of the metaheuristics under study in the filter and wrapper context.
(c) Finally (Section 6.3), we contrast the peakbins obtained in this study with those reported in the literature.

As Fig. 5 shows, the DAP makes use of cross-validation to assess model quality. The number of folds is set to $k = 5$ because cross-validation consumes a great deal of resources. Furthermore lower values of k produce more pessimistic estimates and higher values more optimistic results. The *true* generalization error is not usually known, and it is not possible to determine whether a given estimate is an overestimate or underestimate. However, cross-validation is suitable for model comparison purposes. In order to reduce variability, 10 cross-validation runs are performed using different partitions. The validation results are averaged over the runs.

As performance measures, we use the following discrimination scores:

- Sensitivity. Sensitivity measures the proportion of actual positives that are correctly identified as such.
- Specificity. Specificity measures the proportion of negatives that are correctly identified.

To measure the quality of the algorithms used for peakbin selection, we report the average number of features selected by each strategy and the robustness of each algorithm. All experiments have been developed using `Weka` [34], and the source code is available upon request.

In order to support the conclusions obtained, statistical tests were applied. For the first part (a), since we compared two different classifiers over multiple datasets, we applied the Wilcoxon signed-rank test following the recommendations of Demšar [19]. This is equivalent to the paired t-test for the case in which values may not fit a normal distribution. In the second part (b), we applied the guidelines proposed by García and Herrera [22,26] because we present the results of several metaheuristics without a control method. They propose using a set or family of hypotheses associated with a set of pairwise comparisons to compare the performance of a set of classifiers over multiple datasets. To adjust the value of the level of significance α, García and Herrera conclude that Bergmann-Hommel's procedure is the most suitable. They also propose an adjustment of the p-value (APV) of a pairwise comparison to take into account the remaining comparisons belonging to the family.

6.1. Study of the effects of data preprocessing

First, we analyze the effect of preprocessing on the classification performance. For this purpose, we compare the performance of NB and SVM on data with and without the preprocessing pipeline. Table 2 shows the average number of peakbins and the associated standard deviation. The new bins are only 3.67%, 5.80% and 0.62% of the original dataset for OVA, TOX and HCC, respectively.

The results of the NB and SVM classification models are shown in Table 3. Column 2 reports the data type; a b for binned data and a p for preprocessed data. The following columns present the sensitivity and specificity for NB and then for SVM. Note that for the OVA and HCC datasets, preprocessing can improve the predictive power of the models when using the NB classifier. For the TOX dataset, however, variability increases causing high values of standard deviation. When using SVM, no improvement is appreciated for the OVA dataset. For HCC, SVM is able to achieve slightly better performance and finally, for TOX, although variability is not so high, the model is unable to detect affected samples. A possible reason for high variability is the low number of cases and therefore the difficulty in cleaning the spectra.

Table 1
Characteristics of the datasets.

ID	#Inst.	#(c,a)	#Bins	m/z Value ranges	Type	Ref.
OVA	200	(79,121)	45,200	700.116–12,000	SELDI-TOF	[61]
TOX	62	(34,28)	45,200	799.115–12,000	SELDI-TOF	[63]
HCC	150	(72,78)	36,802	700.725–9,999.975	MALDI-TOF	[68]

To check whether or not the differences between the generated models are statistically significant, we performed the following comparisons: (*a*) all results achieved by both classifiers on each dataset, (*b*) results output by each classifier on each dataset (two comparisons, one per classifier), and (*c*) sensitivity and specificity values per classifier on each dataset (four comparisons in all). For comparison (*a*), the test detects differences for a level of significance of $\alpha = 0.1$ (p-value = 0.052) whereas, only differences for NB results are statistically significant at $\alpha = 0.1$ (p-value = 0.094) in comparison (*b*). In both cases, differences are in favour of the models built on preprocessed data. No other statistical differences are found.

Since any preprocessing can degrade the quality of the input spectra, this analysis may help us to determine whether or not the output spectra are or not suitable for our purpose. The results suggest that experiments performed on the TOX dataset may lead to less robust results than for OVA and HCC because of its high variability.

6.2. Peakbin selection analysis

The results of peakbin selection are presented in Tables 4–6. Table 4 refers to the sensitivity and specificity reached by NB and SVM when using the set of peakbins found by BF, GA, SS and VNS. The first column shows the ID of the dataset, followed by the algorithm ($\mathcal{A}$) used. Columns 3 to 6 represent the values achieved when using the filter approach and columns 7 to 10 list wrapper values.

Table 5 shows the average number of peakbins selected by each algorithm on each dataset. As in Table 4, columns 1 and 2 refer to the ID and $\mathcal{A}$ respectively. From columns 3 to 5, the values for the average number of peakbins selected with the standard deviation, and the consistency measures computed using the consistency index $\mathcal{I}_C$, Eq. (1), and the Jaccard index $\mathcal{I}_J$, Eq. (2). These values refer to the filter approach. Columns 6 to 11 present the same values using NB and SVM in the wrapper scheme.

Finally, Table 6 reports the computational time of each algorithm. Column *Filter* represents the average time required by the search process, in the filter scheme, on a single training dataset and its associated standard deviation. The following columns correspond to the wrapper scheme with NB and SVM, respectively.

6.2.1. Filter approach

In order to determine the confidence in the results, the statistical tests were applied to (a) the performance, (b) the subset size and (c) the computational time.

6.2.1.1. Performance. The performance of the four algorithms seems to be similar in OVA and HCC and different in TOX; however, the high variability in TOX suggests that no clear conclusion can be drawn for this dataset. Statistical tests find significant differences between the SS and the BF and VNS algorithms at $\alpha = 0.05$. In both cases, SS is the one with the lowest values for all heuristics.

6.2.1.2. Feature selection analysis. SS is the algorithm that most reduces the number of features as well as the one with the lowest standard deviation. GA is the strategy that finds larger sets of features in all cases; however, it achieves low values in stability measures except for HCC. The dimensionality reduction performed by SS leads to less stable sets of peakbins than for BF and VNS.

In HCC, results are very similar in terms of both dimensionality reduction and stability measures. We find that the stochastic methods used – GA, SS and VNS – are, in most cases, less stable since randomness allows the search to escape from local minima and so reach different solutions. Finally, the differences between SS and GA were found to be statistically significant at $\alpha = 0.05$.

6.2.1.3. Computational time. In all cases, BF is the fastest algorithm; however these differences are only significant with respect to VNS with a confidence level of $\alpha = 0.1$.

6.2.2. Wrapper approach

Taking into account what we said above about cross-validation (see Section 6), we set the internal k-fold cross-validation to $k = 5$ in our experiments.

Table 2
Number of bins of the original datasets after binning and average number of peakbins with the associated standard deviation of preprocessed data.

ID	#Bins	#Peakbins
OVA	45,200	1,660.98 ± 4.36
TOX	45,200	2,620.00 ± 21.65
HCC	36,802	227.30 ± 2.44

Table 3
Performance of the baseline classifiers on binned and preprocessed data. Column 1 refers to the ID associated with each dataset, then the type of data (*b* for binned and *p* for preprocessed). The following columns show the sensitive and specificity results with the standard deviation for the NB and SVM models, respectively.

ID	t	NB		SVM	
		Sensitivity	Specificity	Sensitivity	Specificity
OVA	b	81.40 ± 0.59	81.52 ± 0.93	98.34 ± 1.57	99.88 ± 0.40
	p	94.42 ± 1.71	96.06 ± 1.33	98.35 ± 0.86	99.70 ± 0.64
TOX	b	75.62 ± 2.64	71.07 ± 0.56	96.81 ± 2.50	84.76 ± 4.46
	p	74.71 ± 18.82	83.07 ± 18.71	93.27 ± 3.66	88.27 ± 6.99
HCC	b	47.92 ± 1.43	87.01 ± 0.77	87.37 ± 2.15	88.50 ± 0.91
	p	86.94 ± 3.75	87.42 ± 2.83	88.92 ± 2.97	90.75 ± 2.68

Table 4
Sensitivity and specificity with their respective standard deviation obtained by the algorithms ($\mathcal{A}$) best first (BF), genetic algorithm (GA), scatter search (SS) and variable neighborhood search (VNS). The best method for each dataset, performance measure and classifier is marked in bold.

ID	$\mathcal{A}$	Filter				Wrapper			
		NB		SVM		NB		SVM	
		Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
OVA	BF	**97.63 ± 1.02**	97.50 ± 1.68	**99.12 ± 0.88**	**98.68 ± 1.52**	92.52 ± 3.49	91.93 ± 2.92	93.62 ± 2.11	91.03 ± 4.39
	GA	95.69 ± 1.03	96.92 ± 0.87	98.63 ± 0.78	97.00 ± 2.12	94.44 ± 0.76	95.00 ± 1.45	96.39 ± 2.13	96.04 ± 3.47
	SS	96.31 ± 1.31	96.09 ± 1.10	97.34 ± 1.63	95.70 ± 1.51	94.71 ± 1.93	92.03 ± 4.21	96.21 ± 2.61	91.83 ± 2.58
	VNS	96.11 ± 1.04	**97.79 ± 1.36**	98.71 ± 0.88	**98.68 ± 1.24**	**95.48 ± 0.73**	**95.42 ± 1.49**	**97.18 ± 1.86**	**96.74 ± 1.81**
TOX	BF	61.18 ± 27.41	78.63 ± 29.32	**90.69 ± 5.78**	**87.73 ± 6.55**	62.10 ± 12.14	74.06 ± 14.25	72.54 ± 10.59	71.37 ± 9.11
	GA	**73.65 ± 14.44**	**86.83 ± 16.61**	80.84 ± 12.78	85.59 ± 6.36	74.91 ± 19.43	**84.02 ± 17.54**	**91.29 ± 6.09**	83.62 ± 8.67
	SS	67.08 ± 26.01	81.27 ± 25.50	81.16 ± 15.54	85.32 ± 7.25	68.54 ± 14.88	75.39 ± 18.52	70.74 ± 11.18	80.60 ± 5.00
	VNS	71.71 ± 20.31	83.06 ± 25.39	87.28 ± 5.13	86.25 ± 6.85	**76.01 ± 16.18**	83.98 ± 21.10	86.51 ± 8.91	**85.99 ± 6.13**
HCC	BF	89.02 ± 4.31	86.80 ± 3.29	87.02 ± 5.33	87.24 ± 3.52	81.93 ± 6.52	84.68 ± 4.33	81.61 ± 4.11	84.62 ± 4.21
	GA	**89.67 ± 4.61**	86.47 ± 4.37	84.92 ± 5.30	**87.54 ± 4.87**	**86.32 ± 4.49**	85.09 ± 6.03	**86.70 ± 3.70**	**89.80 ± 2.50**
	SS	88.50 ± 4.26	**87.21 ± 3.66**	86.58 ± 3.49	86.60 ± 3.08	85.27 ± 3.29	86.32 ± 3.93	78.25 ± 12.00[a]	76.29 ± 8.72[a]
	VNS	88.77 ± 4.54	87.18 ± 3.60	**87.31 ± 5.35**	87.40 ± 4.94	86.13 ± 5.00	**87.99 ± 4.66**	83.47 ± 4.29	86.27 ± 5.23

[a] Denotes SS with reduced search space due to the low convergence speed.

Table 5
Mean number of peakbins (with its associated standard deviation) and stability measures Σ_C and Σ_J obtained by BF, GA, SS and VNS.

ID	$\mathcal{A}$	Filter			Wrapper					
					NB			SVM		
		#Peakbins	Σ_C	Σ_J	#Peakbins	Σ_C	Σ_J	#Peakbins	Σ_C	Σ_J
OVA	BF	61.42 ± 1.36	0.4953	0.3786	6.10 ± 1.31	0.1270	0.0823	5.38 ± 0.57	0.1433	0.0979
	GA	170.92 ± 19.32	0.1767	0.1779	347.40 ± 54.47	0.0637	0.1914	199.98 ± 22.39	0.0413	0.1083
	SS	27.86 ± 1.74	0.3744	0.2657	7.04 ± 0.43	0.1306	0.0822	6.24 ± 0.55	0.1302	0.0846
	VNS	92.14 ± 3.89	0.4280	0.3211	64.10 ± 4.38	0.2938	0.2167	65.34 ± 4.37	0.3087	0.2220
TOX	BF	40.52 ± 3.68	0.2422	0.1660	4.82 ± 0.63	0.0401	0.0272	3.82 ± 0.51	0.0537	0.0382
	GA	185.04 ± 81.65	0.0063	0.0636	401.46 ± 32.70	0.0071	0.1116	352.38 ± 78.20	0.0638	0.0789
	SS	17.38 ± 1.30	0.1776	0.1165	5.04 ± 0.49	0.1448	0.0979	4.12 ± 0.41	0.0982	0.0662
	VNS	156.72 ± 8.67	0.2511	0.1935	117.62 ± 8.52	0.1998	0.1493	129.12 ± 10.56	0.2004	0.1591
HCC	BF	32.68 ± 1.63	0.4577	0.4113	33.24 ± 1.62	0.0535	0.0670	13.60 ± 1.19	0.0352	0.0602
	GA	32.86 ± 1.38	0.4329	0.3837	10.52 ± 2.37	0.0321	0.2867	90.74 ± 4.35	0.0169	0.3868
	SS	31.26 ± 1.78	0.4331	0.3865	11.28 ± 1.08	0.1532	0.1281	3.00 ± 2.63[a]	0.4137[a]	0.4218[a]
	VNS	31.36 ± 1.46	0.4348	0.3876	19.50 ± 1.49	0.1520	0.1498	20.66 ± 2.35	0.1421	0.1485

[a] Denotes SS with reduced search space due to the low convergence speed.

6.2.2.1. Performance. For the NB classifier, BF is the least discriminative strategy; it achieves the lowest sensitivity and specificity values across all datasets. SS achieves slightly lower results than for GA and VNS in OVA and HCC. However, the performance in TOX is poor. VNS is the algorithm that achieves better performance on average. Statistically significant differences were found for BF and VNS with a confidence level of 95% ($\alpha = 0.05$) in favour of VNS.

Results achieved with SVM depend on the dataset. On average, VNS performance is better than BF and SS with a confidence level of $\alpha = 0.1$. For HCC, the reduction of the search space in SS negatively affects its performance scores.

Table 6
Computational time (in seconds) of each feature selection algorithm ($\mathcal{A}$). [a]

ID	$\mathcal{A}$	t(s)		
		Filter	Wrapper NB	SVM
OVA	BF	40.1 ± 9.3	169.3 ± 41.0	606.9 ± 117.0
	GA	213.8 ± 40.2	6810.7 ± 770.4	49042.3 ± 5410.7
	SS	401.0 ± 230.1	6480.8 ± 2170.0	29260.5 ± 9980.4
	VNS	1134.6 ± 347.1	841.5 ± 442.5	2382.7 ± 896.0
TOX	BF	56.4 ± 19.7	105.0 ± 28.1	437.5 ± 245.8
	GA	588.2 ± 159.3	3130.1 ± 450.8	21966.8 ± 3863.6
	SS	220.4 ± 135.3	922.1 ± 447.4	3701.5 ± 1693.1
	VNS	1341.6 ± 800.6	660.0 ± 352.3	1906.1 ± 899.6
HCC	BF	0.3 ± 0.1	35.3 ± 16.8	13337.8 ± 3560.6
	GA	8.3 ± 1.0	664.2 ± 55.9	1040.4 ± 81.6
	SS	56.2 ± 22.9	7769.7 ± 3187.5	4633.2 ± 2070.2[a]
	VNS	46.3 ± 13.3	103.8 ± 30.6	10590.5 ± 4190.1

[a] denotes SS with reduced search space due to the low convergence speed. The algorithms run on a SUN x4600 M2 with 24 cores.

6.2.2.2. Feature selection analysis. In OVA and TOX, BF is the algorithm that most reduces the number of features. However, in HCC, BF retains more features than the rest of the algorithms. VNS stands out as the most stable method for OVA and TOX, and SS seems to strike a balance between reduction and stability. No statistically significant differences were found in any case.

6.2.2.3. Computational time. The wrapper scheme is very time-consuming, specially when using the SVM classifier. As in the filter case, BF is again the fastest algorithm. The test detects statistically significant differences at $\alpha = 0.1$ using NB when comparing BF with the GA and SS algorithms.

6.2.3. Filter vs wrapper

6.2.3.1. Performance. For OVA and HCC, the filter approach outperforms wrapper results in most cases. For TOX, no conclusion can be reached because of the high data variability. These differences are statistically significant at $\alpha = 0.05$. For NB, the wrapper approach achieves better mean values in performance measures than CFS in training data, suggesting that this scheme might be suffering from overfitting.

6.2.3.2. Feature selection analysis. All the algorithms except the GA find larger and more stable subsets when using the filter than the wrapper. In the case of GA, there is no clear pattern, despite the fact that the subsets are more stable for OVA and HCC.

6.2.3.3. Computational time. As expected, the filter was found to be faster than the wrapper approach with a significance level of $\alpha = 0.05$.

6.2.4. Summary

In general, experimental results conducted with the filter approach provide better performance and computational time at the expense of larger feature subsets. The performance of BF, GA and VNS is similar. Of the four algorithms studied, however, BF with CFS seems to be the best one since it finds more stable feature subsets in OVA and HCC, and its values are close to the best ones (found by VNS) for TOX. BF is also less time-consuming than any other algorithm studied.

6.3. Peakbin analysis

The results of the experiments performed provide a set of relevant peakbins in MS data. Naturally, not all the selected peakbins have the same discriminative power. For this reason, a posterior analysis of such results is necessary in order to discover candidate biomarkers. The aim of the analysis that we present is to draw up a list of the most important peakbins selected by the algorithms and compare them with the peakbins discovered by other authors. The reported peakbins are anonymous so that the only thing that is known about them is their m/z. Ideally this work should continue with the characterization of the protein or peptide that caused the peakbins and the validation of the peakbins as biomarkers.

In this study, we analyze, for each dataset, the occurrence of the peakbins selected by the algorithms under study to provide the list of peakbin candidates. Then we study the occurrence, in our experiments, of the peakbins proposed in previous works.

Table 7 shows the frequency of occurrence of peakbins selected by BF, GA, SS and VNS for the peakbins reported in [13,63,68,38,67]. The first and second columns show the dataset ID and the reference of the work. The next two columns present m/z ranges reported in the works mentioned above and the matched m/z range values. Single values correspond to a single peakbin. Finally the frequency of ocurrence in the algorithm under study is shown.

Table 7
Frequency of occurrence of peakbins selected by BF, GA, SS and VNS and reported for OVA [13,38], TOX [63] and HCC [67,68] datasets.

ID	Ref.	m/z range	Peakbin (m/z)	Occurrence (%)											
				Filter				Wrapper							
								NB				SVM			
				BF	GA	SS	VNS	BF	GA	SS	VNS	BF	GA	SS	VNS
OVA	[13]	845.089	845.116–845. 366	100	68	100	96	18	46	12	58	20	12	8	72
		1151.684	1152. 616	96	46	60	76	2[a]	68	4	62	6	20	–	48
		8602.237	8582.116–8626. 116	64	54	58	74	4[a]	20	6	52	4	1 8	10	50
		8709.548	8705.116–8714. 366	18	42	30	56	6	20	4[a]	28	2	6	4	26
	[38]	1046.546–1055.644	1050.116–1051. 116	84	92	84	100	–	74	8	92	2	44	4[a]	92
		3955.309–3972.978	3961.866–3963. 616	100	18	10	60	16	44[a]	–	4	8	22	6	6
		7049.480–7073.061	7052.116–7063. 116	98	34	48	86	4	58	4	44	–	24[a]	22	52
		7295.049–7319.037	7305.366–7310.866	2	22	2	10	–	28	–	8	–	28	4	6
		8319.365–8344.980	8319.866–8323. 366	42	34[a]	14[a]	48	2[a]	42[a]	–	28	–	48[a]	2	24
		8508.124–8534.028	8516.116–8529.116	50	28	22	60	–	20[a]	2	30	2[a]	12	4	30
		8590.289–8616.318	8582.116–8626. 116	64	54	58	74	5[a]	20	6	52	6	1 8	10	50
TOX	[63]	810.337	810.615	86	28	68	74	4	22	–	58	22	44	46	56
		981.824	978.365–983.365	–	6	–	–	–	42	–	–	–	46	–	–
		1987.972	1968.865–2106.115	–	58	–	68	–	–	–	–	–	42	–	–
		2013.577	1968.865–2106.115	–	58	–	68	–	–	–	–	–	42	–	–
		10645.952	–	–	–	–	–	–	–	–	–	–	–	–	–
HCC	[68]	933.6–938.2	933.225–938.475	68	66	64	66	12	80	36	50	18	30	–[a]	8
		1378.9–1381.2	1379.725–1380.475	76	78	76	68	40	32	42	18	12	38	–[a]	14
		1737.1–1744.6	1739.225–1743. 975	88	82	86	90	10	70[a]	16	48	20	52	–[a]	44
		1863.4–1871.3	1863.975–1870. 225	98	98	98	94	56	78	70	72	38	66	80[a]	70
		2528.7–2535.5	2530.225–2532. 225	64	60	56	58	8[a]	52	20	44	14	64	20[a]	46
		4085.6–4097.9	4092.225–4092.975	46	46	46	46	24	24	32	28	30	36	20[a]	28
	[67]	1777.0–1784.8	1776.975–1782.975	84	88	84	88	8	66	20	50	16	34	–[a]	50
		1864.0–1870.2	1863.975–1870. 225	98	94	98	94	56	78	70	72	38	66	80[a]	70
		2303.7–2309.9	2306.975–2308.725	36	34	34	38	–	44	4[a]	20	8	38	20[a]	30
		2377.6–2382.6	2377.475–2379.225	2	–	2	2	12[a]	68[a]	6	8	2[a]	68	–[a]	–

[a] Denotes the frequency percentage given does not correspond to the peakbin value shown in column 4 but to a close one that belongs to the range of m/z values reported in the corresponding work, -denotes not matched.

6.3.1. OVA

Fig. 6 shows the peak frequency plot for the OVA dataset. The top subplot shows the differences among the average peakbins for each phenotype. The bottom subplots show, for each algorithm and approach, the peakbins with occurrence values greater than 90%. In the case of frequency values lower than 90%, the top five most often selected peakbins are considered instead.

For CFS, peakbins [845.116–845.366] and [1034.116–1036.116] are always selected for BF and SS. They also have very high frequency values in VNS (96% and 98% respectively). Another interesting peakbin is [8996.855–9043.118], which has an occurrence of 100% in VNS and GA. For the wrapper scheme, GA is the only algorithm that has peakbins with a frequency value greater than 90% when using NB. However, peakbins [1034.116–1036.116] and [8996.855–9043.118] are highly represented in BF, SS and VNS, and GA and VNS, respectively.

Petricoin [61] proposed five biomarkers with a very high performance. However, these results are said to contain artifacts from an unfit denoising [4,5], making a comparison with other results impossible. All the peakbins proposed in [13] are found by all the proposed algorithms. However, the occurrence levels we found differ from those reported in some cases. With CFS, the proposed peakbin [845.089] belongs to the top five most selected peakbins for BF, SS and VNS, whereas this applies to [1151.684] in the case of BF only. In the wrapper approach, none of the peakbins belong to those with higher frequency values.

The second paper [38] proposes seven different peakbins to those reported above. Using CFS, the m/z range [3955.309–3972.978] reaches 100% occurrence for BF. The range ([1046.546–1055.644] is the only peakbin with high frequency value in all strategies, especially in VNS with 100%. Wrapper results show low occurrence values, except for the peakbin [1046.546–1055.644] in the case of GA and VNS using NB and VNS using SVM.

6.3.2. TOX

As shown in Fig. 7, the analysis of the most selected peakbins in CFS reveals that peakbin [1705.865–1786.865] is the one with highest values in BF (98%) and VNS (96%). In GA and SS, it achieves an occurrence of 86% and 68% respectively. In wrapper with NB, only GA and VNS find the peakbins [2268.365–2335.115] and [1788.865–1942.865] with a frequency value higher than 90%, scoring 92% and 94%, respectively. With SVM, VNS selects [1705.61–1943.865] with an occurrence of

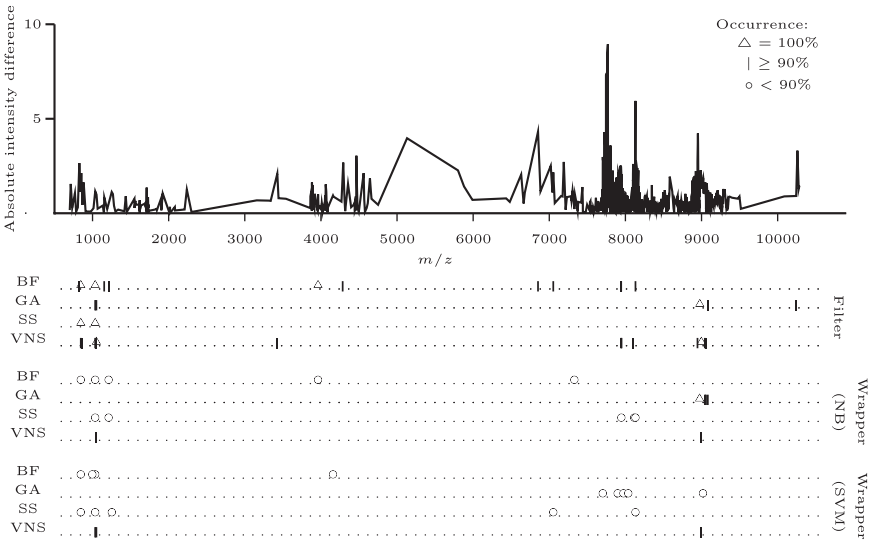

Fig. 6. Peak frequency plot for the OVA dataset.

100%. Finally peakbin [7002.115–7004.365] seems another interesting m/z range since it has been selected by BF and SS in filter and wrapper schemes.

Comparing our results with those reported in [63] using CFS, the m/z range [810.337] is the only one that is found by all algorithms. Furthermore, the occurrence values are in the top two for BF and SS. Reported peakbins [1987.972] and [2013.577] match our peakbin [1968.865–2106.115]. Such differences may be due to the preprocessing engine used. Note that, in spite of the difficulties, VNS finds three and GA four of the five peakbins reported with high frequency values. For the wrapper scheme occurrence levels are, in most cases, quite low (<60%). GA is the only strategy able to find two peakbins – [810.337] and [981.824] – using NB, but finds all of them except [10645.952] using SVM. Peakbin [810.337] seems to be a good candidate because it appears in most cases.

6.3.3. HCC

In this dataset, Fig. 8 shows that the peakbins [1906.725–1911.225] and [1863.975–1870.225] are the top two peakbins selected by all algorithms. The first peakbin has an occurrence of 100% and the second scores values greater than 90% in all cases. In wrapper with NB, both peakbins belong to the top three peakbins of BF, SS and VNS. Finally, when using SVM, VNS also finds these peakbins that are in the top three.

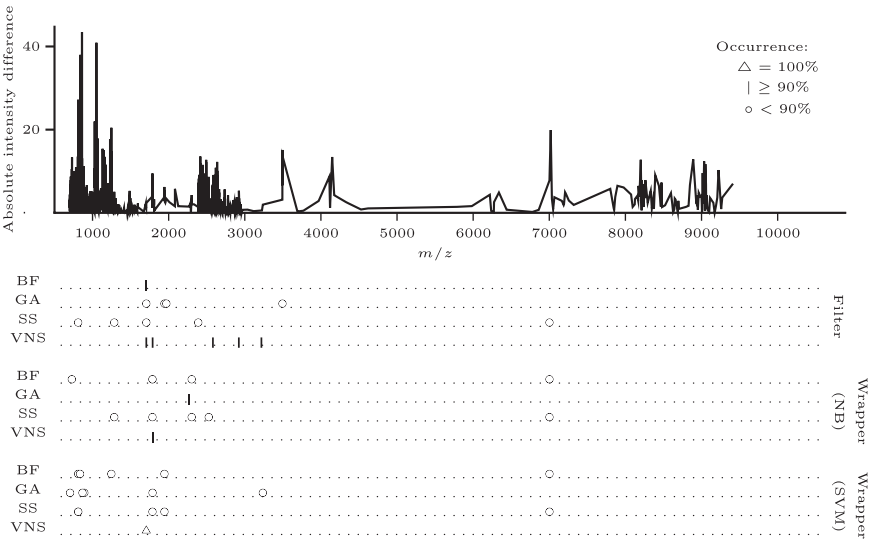

Fig. 7. Peak frequency plot for the TOX dataset.

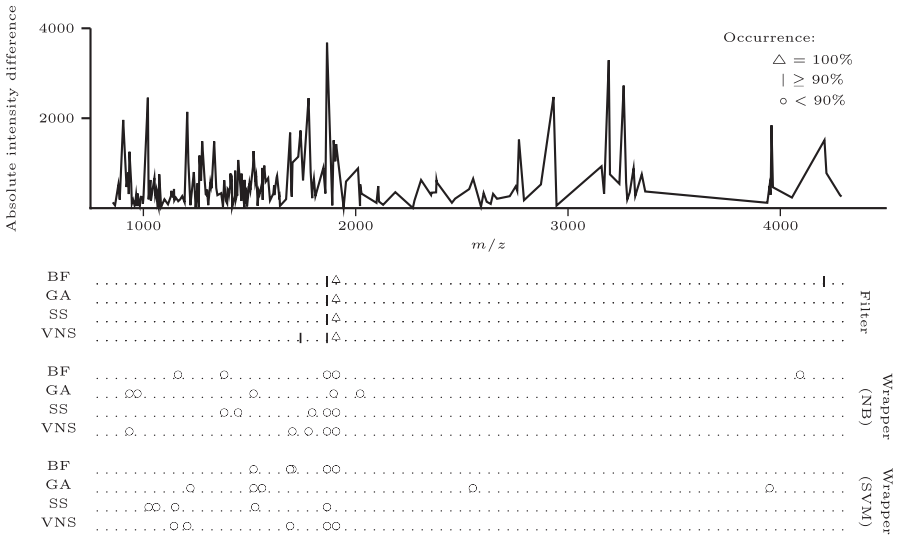

Fig. 8. Peak frequency plot for the HCC dataset.

The original work [68] reported six peakbins. In the filter approach, all algorithms find these peakbins with a similar occurrence. The peakbins [1737.1–1744.6] and [1863.4–1871.3] have high occurrence values, especially the first one with values greater than 90%. Using NB in the wrapper approach, occurrence decreases in almost all cases. With SVM, VNS finds the peakbins with a similar occurrence level as with NB, except for [933.6–938.2]. In the second work [67], the results are similar.

6.3.4. Summary

For each dataset, a small number of peakbins have been found in most results. These peakbins are:

- *OVA:* [845.116–845.366], [1034.116–1036.116], [1050.116–1051.116] and [8996.855–9043.118].
- *TOX:* [1705.865–1786.865].
- *HCC:* [1739.225–1743.975] and [1863.4–1871.3].

Other peakbins, which have lower frequency values than the above, but are interesting because of the results achieved by some of the algorithms and reported in the literature are:

- *OVA:* [1152.616], [3961.866–3963.616] and [3961.866–3963.616].
- *TOX:* [810.615].
- *HCC:* [933.225–938.475], [1379.725–1380.475] and [1776.975–1782.975].

Except for TOX, most peakbins proposed in other works are found by the algorithms used. However, the experiments run show up some discrepancies about the occurrence level of such peakbins. A possible reason is that serum contains a huge number of discriminatory molecules and the chance of different algorithms using different search strategies finding the same peakbins is very small. Other reasons could be the differences in the DAP or in the preprocessing algorithms.

7. Conclusions

In this paper we have applied the best first, genetic algorithm, scatter search and variable neighborhood search metaheuristics to the problem of peakbin discovery in MS proteomic data.

The original raw data is affected by different noises and biases that should be removed before tackling any analysis. Following the recommendation of several authors, our preprocessing engine consists of baseline removal, normalization, signal smoothing, peak detection and peakbin assembly and quantification. Preprocessing not only corrects the data but also reduces data dimensionality without losing discriminatory capability.

After preprocessing, data mining techniques can be applied. Following a rigorous validation scheme, we apply a pipeline that simulates real situations where we have a set of labeled cases on which we perform the peakbin discovery. The predictive model is built based on such peakbins. Finally, new unlabeled cases are classified by applying this model.

Even though the filter approach is usually reported to perform worse than the wrapper scheme, we found that the evaluation of CFS performance shows higher average results than the wrapper method in the MS domain. Overfitting could be the reason for these results since it is known that wrappers suffer from this effect in small samples.

Except for TOX, the BF, GA, SS and VNS strategies presented achieved competitive results and found most peakbins reported in other papers for OVA and HCC. For TOX, it is difficult to extract knowledge because of the huge variance, and, consequently, stability is very low. Of the four algorithms used, BF using CFS seems to be the one that achieves better results in terms of dimensionality reduction and computational time, while performing similarly to VNS and GA.

Despite the enormous number of selected peakbins, mean values achieved by the different models are very similar to each other. This could suggest that different proteins or peptides have the same information for prediction purposes. Consequently, future research might undertake the analysis of the effect of redundancy in MS data.

Acknowledgements

This work has been partially supported by the projects TIN-68084-C02-00, TIN2010-20900-C04-04 and TIN2007-62626, Cajal Blue Brain and Consolider CSD2007-00018. Rubén Armañanzas is supported by a Juan de la Cierva grant (Spanish Ministry of Science and Innovation). Part of the computer time was provided by the Centro Informático Científico de Andalucía (CIC).

References

[1] B.L. Adam, Y. Qu, J.W. Davis, M.D. Ward, M.A. Clements, L.H. Cazares, O.J. Semmes, P.F. Schellhammer, Y. Yasui, Z. Feng, G.L. Wright, Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men, Cancer Research 62 (2002) 3609–3614.
[2] E. Amaldi, V. Kann, On the approximation of minimizing non zero variables or unsatisfied relations in linear systems, Theoritical Computer Science 209 (1998) 237–260.
[3] R. Armañanzas, Y. Saeys, I. Inza, M. García-Torres, C. Bielza, Y. van de Peer, P. Larrañaga, Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms, in: IEEE/ACM Transactions on Computational Biology and Bioinformatics. accepted for publication.
[4] K.A. Baggerly, J.S. Morris, K.R. Coombes, Reproducibility of SELDI-TOF protein patterns in serum: Comparing datasets from different experiments, Bioinformatics 20 (5) (2004) 777–785.
[5] K.A. Baggerly, J.S. Morris, S.R. Edmonson, K.R. Coombes, Signal in noise: evaluating reported reproducibility of serum proteomic tests for ovarian cancer, Journal of the National Cancer Institute 97 (4) (2005) 307–309.
[6] K.A. Baggerly, J.S. Morris, J. Wang, D. Gold, L.C. Xiao, K.R. Coombes, A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples, Proteomics 3 (9) (2003) 1667–1672.
[7] J. Bala, K. Dejong, J. Huang, H. Vafaie, H. Wechsler, Using learning to facilitate the evolution of features for recognizing visual concepts, Evolutionary Computation 4 (3) (1996) 297–311.
[8] R. Blanco, I. Inza, M. Merino, J. Quiroga, P. Larrañaga, Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with tips, Journal of Biomedical Informatics 38 (5) (2005) 376–388.
[9] M.S. Boguski, M.W. McIntosh, Biomedical informatics for proteomics, Nature 422 (6928) (2003) 233–237.
[10] P. Bougioukos, D. Cavouras, A. Daskalakis, S. Kostopoulos, I. Kalatzis, G. Nikiforidis, A. Bezerianos, Proteomic mass spectra classification for biomarker discovery in prostate cancer, employing pattern recognition techniques, in: Proceedings of the second International Conference on Experiments/Process/System Modelling/Simulation & Optimization, 2007.
[11] M. Cannataro, P.H. Guzzi, T. Mazza, P. Veltri, Preprocessing, management, and analysis of mass spectrometry proteomics data, in: Proceedings of Network Tools and Applications in Biology, 2005.
[12] C.H. Cheng, T.L. Chen, L.Y. Wei, A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting, Information Sciences 180 (9) (2010) 1610–1629.
[13] T.P. Conrads, V.A. Fusaro, S. Ross, D. Johann, V. Rajapakse, B.A. Hitt, S.M. Steinberg, E.C. Kohn, D.A. Fishman, G. Whiteley, J.C. Barrett, L.A. Liotta, E.F. Petricoin, T.D. Veenstra, High resolution serum proteomic features for ovarian cancer detection, Endocrine-Related Cancer 11 (2004) 163–178.
[14] K.R. Coombes, K.A. Baggerly, J.S. Morris, Pre-Processing Mass Spectrometry Data, Springer, Boston, MA, 2007. Chapter. 4, pp. 79–102.
[15] K.R. Coombes, S. Tsavachidis, J.S. Morris, K.A. Baggerly, M. Hung, H.M. Kuerer, Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform, Proteomics 5 (16) (2005) 4107–4117.
[16] C.G. da Silva, Time series forecasting with a non-linear model and the scatter search meta-heuristic, Information Sciences 178 (16) (2008) 3288–3299. including Special Issue: Recent advances in granular computing, Fifth Internation Conference on Machine Learning and Cybernetics.
[17] S. Datta, L.M. DePadilla, Feature selection and machine learning with mass spectrometry data for distinguishing cancer and non-cancer samples, Statistical Methodology 3 (2006) 79–92.
[18] M.E. de Noo, R.A. Tollenaar, A. Ozalp, P.J. Kuppen, M.R. Bladergroen, P.H. Eilers, A.M. Deelder, Reliability of human serum protein profiles generated with c8 magnetic beads assisted MALDI-TOF mass spectrometry, Analytical Chemistry 77 (22) (2005) 7232–7241.
[19] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[20] K. Duan, J.C. Rajapakse, SVM-RFE peak selection for cancer classification with mass spectrometry data, in: Proceedings of the third Asia-Pacific Bioinformatics Conference, 2004, pp. 191–200.
[21] K. Dunne, P. Cunningham, F. Azuaje, Solutions to instability problems with sequential wrapper-based approaches to feature selection. Tech. rep., Department of Computer Science, Trinity College, Dublin, 2002.
[22] S. García, F. Herrera, An extension on ‘‘statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[23] F.C. García-López, M. García-Torres, B. Melián-Batista, J.A. Moreno-Pérez, J.M. Moreno-Vega, Solving the feature selection problem by a parallel scatter search, European Journal of Operations Research 169 (2) (2006) 477–489.
[24] F.C. García-López, M. García-Torres, J.A. Moreno-Pérez, J.M. Moreno-Vega, Scatter search for the feature selection problem, Lecture Notes in Artificial Intelligence 3040 (2004) 517–525.
[25] M. García-Torres, F.C. García-López, B. Melián-Batista, J.A. Moreno-Pérez, J.M. Moreno-Vega, Solving feature subset selection problem by a hybrid metaheuristic. In: First International Workshop in Hybrid Metaheuristics at ECAI 2004 (HM 2004), 2004 pp. 59–69.
[26] S. García, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (2010) 2044–2064.
[27] P. Geurts, M. Fillet, D. de Seny, M.A. Malaise, M.P. Merville, L. Wehenkel, Proteomic mass spectra classification using decision tree based ensemble methods, Bioinformatics 21 (14) (2005) 3138–3145.
[28] M.L. Ginsberg, Essentials of Artificial Intelligence, Morgan Kaufmann, 1993.
[29] F. Glover, Heuristics for integer programming using surrogate constraints, Decision Sciences 8 (1977) 156–166.

[30] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 5 (1986) 533–549.
[31] D.E. Goldberg, Genetics Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.
[32] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[33] I. Guyon, J. Weston, S. Barnhill, V.N. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1-3) (2002) 389–422.
[34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (1) (2009) 10–18.
[35] M.A. Hall, Correlation-based feature subset selection for machine learning. Ph.d. thesis, University of Waikato, Hamilton, New Zealand, 1998.
[36] P. Hansen, N. Mladenovic̀, Variable neighborhood search, Computers & Operations Research 24 (1997) 1097–1100.
[37] P. Hansen, N. Mladenovic̀, Variable neighborhood search: principles and applications, European Journal of Operational Research 130 (2001) 449–467.
[38] S. He, X. Li, Profiling of high-throughput mass spectrometry data for ovarian cancer detection, in: Proceedings of the Second International Conference on Experiments/Process/System Modelling/Simulation & Optimization, vol. 4881 of Lecture Notes in Computer Science, 2007, pp. 860–869.
[39] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, University of Michigan Press, 1975.
[40] W.H. Hsu, Genetic wrappers for feature selection in decision tree induction and variable ordering in bayesian network structure learning, Information Sciences 163 (1-3) (2004) 103–122.
[41] Q. Hu, S. An, D. Yu, Soft fuzzy rough sets for robust feature evaluation and selection, Information Sciences 180 (22) (2010) 4384–4400.
[42] Q. Hu, J. Liu, D. Yu, Stability analysis on rough set based feature evaluation, in: Rough Sets and Knowledge Technology, vol. 5009 of Lecture Notes in Computer Science, 2008, pp. 88–96.
[43] T.W. Hutchens, T. Yip, New desorption strategies for the mass spectrometric analysis of macromolecules, Rapid Communications in Mass Spectrometry 7 (7) (1993) 576–580.
[44] L. Jacotot, P. Vaglio, Y. Toiron, H. Sobol, J.P. Borg, X. Saunier, E. Russo, Automated, high throughput preparation of proteinchip®arrays for SELDI-TOF MS profiling, International Biotechnology Laboratory 24 (2) (2006) 20–21.
[45] F. Janssen, J. Fürnkranz, A re-evaluation of the over-searching phenomenon in inductive rule learning, in: Proceedings of the SIAM International Conference on Data Mining, , 2009, pp. 329–340.
[46] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, 1995, pp. 338–345.
[47] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms, in: Proceedings of the Fifth IEEE International Conference on Data Mining, 2005, pp. 218–225.
[48] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowledge and Information Systems 12 (1) (2007) 95–116.
[49] M. Karas, D. Bachmann, U. Bahr, F. Hillenkamp, Matrix-assisted ultraviolet laser absorption of non-volatile compounds, International Journal of Mass Spectrometry and Ion Processes 78 (1987) 53–68.
[50] I. Kaya, A genetic algorithm approach to determine the sample size for attribute control charts, Information Sciences 179 (10) (2009) 1552–1566. including Special Issue on Artificial Imune Systems.
[51] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.
[52] P. Krízek, Feature selection: stability, algorithms, and evaluation, Ph.d. thesis, Czech Technical University in Prague, 2008.
[53] P. Krízek, J. Kittler, V. Hlavác, Improving stability of feature selection methods, in: Computer Analysis of Images and Patterns, vol. 4673 of Lecture Notes in Computer Science, 2007, pp. 929–936.
[54] L.I. Kuncheva, A stability index for feature selection, in: Proceedings of the 25th IASTED International Multi-Conference, 2007, pp. 390–395.
[55] M. Laguna, R. Martí, Scatter Search: Methodology and Implementations in C, Kluwer Academic Press, 2003.
[56] Q. Liu, A.H. Sung, M. Qiao, Z. Chen, J.Y.Y. ang, M.Q. Yang, X. Huang, Y. Deng, Comparison of feature selection and classification for MALDI-MS data, BMC Genomics 10 (Suppl 1) (2009) S3.
[57] W. Meuleman, J.Y.M.N. Engwegen, M.W. Gast, J.H. Beijnen, M.J.T. Reinders, L.F.A. Wessels, Comparison of normalisation methods for surface-enhanced laser desorption and ionisation (SELDI) time-of-flight (TOF) mass spectrometry data, BMC Bioinformatics 9 (88) (2008).
[58] J.S. Morris, K.R. Coombes, J. Koomen, K.A. Baggerly, R. Kobayashi, Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum, Bioinformatics 21 (9) (2005) 1764–1775.
[59] J. Pacheco, S. Casado, L. Núñez, Use of VNS and TS in classification: variable selection and determination of the linear discrimination function coefficients, IMA Journal of Management Mathematics 18 (2) (2007) 191–206.
[60] C.P. Paweletz, B. Trock, M. Pennanen, T. Tsangaris, C. Magnant, L.A. Liotta, E.F. Petricoin, Proteomic patterns of nipple aspirate fluids obtained by SELDI-TOF: Potential for new biomarkers to aid in the diagnosis of breast cancer, Disease Markers 17 (4) (2001) 301–307.
[61] E.F. Petricoin, A.M. Ardenaki, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, L.A. Liotta, Use of proteomic patterns in serum to identify ovarian cancer, The Lancet 359 (2002) 572–577.
[62] E.F. Petricoin, D.K. Ornstein, C. Paweletz, A. Ardekani, P. Hackett, B. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. Simone, P. Levine, W. Linehan, M. Emniert-Buck, W. Steinberg, E. Khon, L. Liotta, Serum proteomic patterns for detection of prostate cancer, Journal of the National Cancer Institute 94 (20) (2002) 1576–1578.
[63] E.F. Petricoin, V. Rajapaske, E.H. Herman, A.M. Arekani, S. Ross, D. Johann, A. Knapton, J. Zhang, B.A. Hitt, T.P.D.T. Conrads, L.A. Liotta, F.D. Sistare, Toxicoproteomics: serum proteomic pattern diagnostics for early detection of drug induced cardiac toxicities and cardioprotection, Toxicologic Pathology 32 (2004) 122–130.
[64] J. Prados, A. Kalousis, M. Hilario, On preprocessing of SELDI-MS data and its evaluation, in: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, 2006, pp. 953–958.
[65] J.R. Quinlan, R.M. Cameron-Jones, Oversearching and layered search in empirical learning, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1995, pp. 1019–1024
[66] A.J. Rai, Z. Zhang, J. Rosenzweig, I. Shih, T. Pham, E.T. Fung, L.J. Sokoll, D.W. Chan, Proteomic approaches to tumor marker discovery, Archives of Pathology and Laboratory Medicine 126 (12) (2002) 1518–1526.
[67] H.W. Ressom, R.S. Varghese, S.K. Drake, G.L. Hortin, M. Abdel-Hamid, C.A. Loffredo, R. Goldman, Peak selection from MALDI-TOF mass spectra using ant colony optimization, Bioinformatics 23 (5) (2007) 619–626.
[68] H.W. Ressom, R.S. Varghese, E. Orvisky, S.K. Drake, G.L. Hortin, M. Abdel-Hamid, C.A. Loffredo, R. Goldman, Ant colony optimization for biomarker identification from MALDI-TOF mass spectra, in: Proceedings of the 28th Annual International Conference of the IEEE, 2006, pp. 4560–4563
[69] C. Reynés, R. Sabatier, N. Molinari, S. Lehmann, A new genetic algorithm in proteomics: Feature selection for SELDI-TOF data, Computational Statistics & Data Analysis 52 (2008) 4380–4394.
[70] S.J. Russel, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.
[71] Y. Saeys, T. Abeel, Y. van de Peer, Robust feature selection using ensemble feature selection techniques, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, vol. 5212 of Lecture Notes In Artificial Intelligence, 2008, pp. 313–325.
[72] H. Shin, M.K. Markey, A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples, Journal of Biomedical Informatics 39 (2) (2006) 227–248.
[73] H. Shina, B. Sheub, M. Josephc, M.K. Markey, Guilt-by-association feature selection: Identifying biomarkers from proteomic profiles, Journal of Biomedical Informatics 41 (1) (2008) 124–136.

[74] P. Soille, Morphological image analysis, Springer, 1999.
[75] P. Somol, J. Novovičová, Evaluating the stability of feature selectors that optimize feature subset cardinality, in: Structural, Syntactic, and Statistical Pattern Recognition, vol. 5342 of Lecture Notes in Computer Science, 2008, pp. 956–966.
[76] J. Sorace, M. Zhan, A data review and re-assessment of ovarian cancer serum proteomic profiling, BMC Bioinformatics 4 (24) (2003).
[77] M. Sturm, A. Bertsch, C. Gröpl, A. Hildebrandt, R. Hussong, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, A. Zerck, K. Reinert, O. Kohlbacher, OpenMS-An open-source software framework for mass spectrometry, BMC Bioinformatics 9 (163) (2008).
[78] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.
[79] J. Villanueva, J. Phillip, D. Entenberg, C.A. Chaparro, M.K. Tanwar, E.C. Holland, P. Tempst, Serum peptide profiling by magnetic particle-assisted, automated sample processing and MALDI-TOF mass spectrometry, Analytical Chemistry 76 (6) (2004) 1560–1570.
[80] L. Xu, P. Yan, T. Chang, Best first strategy for feature selection, in: Proceedings of the Nineth International Conference on Pattern Recognition, vol. II, 1988, pp. 706–708.
[81] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems 13 (2) (1998) 44–49.
[82] J.S. Yu, S. Ongarello, R. Fiedler, X.W. Chen, G. Toffolo, C. Cobelli, Z. Trajanoski, Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data, Bioinformatics 21 (10) (2005) 2200–2209.
[83] C.H. Zu, S. Ragg, S. Rahmann, Discovering biomarkers for myocardial infarction from SELDI-TOF spectra, in: Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation, 2006, pp. 569–576.