Anal Bioanal Chem (2003) 378 : 2008–2020
DOI 10.1007/s00216-004-2518-0

ORIGINAL PAPER

Simeone Zomer · Christelle Guillo · Richard G. Brereton
Melissa Hanna-Brown

Toxicological classification of urine samples using pattern recognition techniques and capillary electrophoresis

Received: 9 October 2003 / Revised: 21 January 2004 / Accepted: 27 January 2004 / Published online: 9 March 2004

Abstract In toxicology, hazardous substances detected in organisms may often lead to different pathological conditions depending on the type of exposure and level of dosage; hence, further analysis on this can suggest the best cure. Urine profiling may serve the purpose because samples typically contain hundreds of compounds representing an effective metabolic fingerprint. This paper proposes a pattern recognition procedure for determining the type of cadmium dosage, acute or chronic, administrated to laboratory rats, where urinary profiles are detected using capillary electrophoresis. The procedure is based on the composition of a sample data matrix consisting of areas of common peaks, with appropriate pre-processing aimed at reducing the lack of reproducibility and enhancing the potential contribution of low-level metabolites in discrimination. The matrix is then used for pattern recognition including principal components analysis, cluster analysis, discriminant analysis and support vector machines. Attention is particularly focussed on the last of these techniques, because of its novelty and some attractive features such as its suitability to work with datasets that are small and/or have low samples/variable ratios. The type of cadmium administration is detected as a relevant feature that contributes to the structure of the sample matrix, and samples are classified according to the class membership, with discriminant analysis and support vector machines performing complementarily on a training and on a test set.

Keywords Pattern recognition · Capillary electrophoresis · Toxicology · Support vector machines

S. Zomer · R. G. Brereton (✉)
School of Chemistry, University of Bristol,
Cantock's Close, Bristol, BS8 1TS, UK
e-mail: r.g.brereton@bristol.ac.uk

C. Guillo · M. Hanna-Brown
Department of Pharmacy, King's College London,
150 Stamford Street, London, SE1 9NN, UK

Introduction

In toxicology, the exposure of an organism to a specific hazardous substance can determine different pathological conditions depending on factors such as the form of interaction (ingestion, inhalation, contact), the quantity, and the type of dosage that can be either acute (a single large dose) or chronic (several small doses administrated over time). Further investigation of these aspects is therefore a major issue because it may indicate the best cure and it can also provide information on the possible source of contamination, when this is not identified yet.

Urine profiling may serve for this purpose because typically samples contain hundreds of species such as inorganic ions, organic acids, amino acids, purines and pyrimidines, which represent an effective "metabolic fingerprint" of the organism. Because of this, the analysis of urine samples has been the focus of much research especially since the 1980s [1, 2, 3].

Several analytical techniques have been employed for urinary profiling but lately capillary electrophoresis (CE) has emerged as a potential tool [4, 5, 6, 7], because it enables cost-effective, rapid and highly efficient separations with minimal sample volume requirements. Recently, Guillo et al. reported the optimisation and validation of a sulphated β-cyclodextrin-modified micellar electrokinetic capillary electrophoresis (SβCD-MECC) method developed for general profiling of urine, allowing for the separation of over 80 charged and neutral compounds in less than 25 min [8, 9].

Multivariate analysis by means of pattern recognition has an important role in the interpretation of variations in these fingerprints, which would be otherwise difficult by visual inspection. While several works report the use of pattern recognition for urinary profiling coupled with techniques such NMR [10, 11], GC [12, 13], HPLC [14, 15], its use with CE has not been well exploited, with fewer works reported [16].

In this paper we investigate the use of SβCD-MECC methodology combined with pattern recognition as a new tool for urine profiling in a toxicological pilot study examining the influence of cadmium administration on laboratory rats. Two groups of rats are treated using both

chronic and acute dosages with the aim to be able to distinguish the effects from urine profiles.

Data are prepared for pattern recognition by composing a sample matrix with rows referring to the samples and columns to the most frequently encountered peaks in the electropherograms. Instrumental factors that interfere with reproducibility can be reduced by prior pre-processing including baseline correction and profile normalisation, while post-processing based on selective peak normalisation and standardisation enhances the influence of low-level metabolites. Exploratory analysis on the sample data matrix so obtained is performed by means of principal components analysis (PCA) and hierarchical agglomerative cluster analysis (HACA). Visual inspection of the scores plots derived from PCA allows determination of whether the target feature, namely the type of cadmium administration, has a significant influence over the sample matrix, while the dendrogram obtained from HACA provides further insight into the residual lack of reproducibility, by examining the grouping of replicates. Finally, supervised techniques such as discriminant analysis by means of the Mahalanobis distance (DA) and support vectors machines (SVMs) are then used for class modelling aimed at predicting new instances on the basis of the type of cadmium administration. While DA is a widely spread technique, SVMs represent a recent approach, whose potential has yet to be fully exploited in analytical chemistry. Recently, Belousov et al. described its use for classification with particular focus on mid-infrared spectral data [17, 18], while Thiessen et al. [19] applied the approach for time series prediction in process monitoring. Applications in other areas such as genomics [20], optical engineering and econometrics [21] have suggested that this method is a significant improvement on existing approaches, for addressing not only classification tasks but also for calibration. Particularly, the underlying theory shows that the use of SVMs is advisable when dealing with datasets that are limited in size and/or when the ratio samples/variables is rather low, making the method suitable for the case under investigation where both these conditions are met.

Experimental

Sample preparation

Rats were pre-treated with a low methionine diet prior to cadmium dosage. A first group (acute) received one single cadmium dose (to drinking water) at four different concentration levels, while a second group (chronic) was dosed (again via drinking water) over a time of 12 weeks at three different concentration levels (Table 1). For all instances urine was collected over 24 h and subsequently stored at –80°C. Samples were allowed to equilibrate at room temperature before use, and were then vigorously shaken for approximately 1 min and subsequently filtered through a 45-µm filter (Whatman, Clifton, NJ, USA) before analysis.

Table 1 Samples analysed and their dosage level. Each sample was analysed in three replicates

Acute dosage (mg kg^{-1})				Chronic dosage (mg L^{-1})		
0.375	0.75	1.125	1.5	1	3	7
A 313	A 104	A 107[a]	A 110	C 115	C 114	C 101
A 413	A 204[a]	A 207	A 210	C 203	C 202	C 113[a]
A 513	A 304[a]	A 307	A 310[a]	C 215	C 214[a]	C 201
	A 404[a]	A 407[a]	A 410	C 303[a]	C 302	C 213
	A 504			C 315[a]	C 314[a]	C 301
						C 313[a]

[a]Sample was used for the test set

Instrumentation and capillary electrophoresis methodology

CE experiments were carried out on a P/ACE MDQ Capillary Electrophoresis System (Beckman Instruments, Fullerton, CA, USA) fitted with a diode array UV/Vis detector (190–600 nm), a temperature-controlled capillary compartment (liquid cooled) and an autosampler. Electrophoretic data were acquired and analysed with the 32 Karat software. Separations were performed in a 47-cm fused silica capillary (50-µm i.d.) (Composite Metal Services Ltd, Hallow, Worcester, UK). New capillaries were conditioned for 30 min at 25°C with 1 M NaOH, followed by 0.1 M NaOH for 20 min and deionised water for 10 min. The capillary was washed with 0.1 M NaOH and deionised water for 1 min, and then 2 min with the run buffer before each analysis.

Urine samples were analysed using the sulphated β-cyclodextrin-modified micellar electrokinetic capillary chromatography (SβCD-MECC) methodology, developed and validated for urine profiling [8]. The running buffer was composed of 25 mM sodium borate, 75 mM SDS and 6.25 mM sulphated β-cyclodextrin. The pH was adjusted to pH 9.5 with 1 M NaOH, after addition of SDS and cyclodextrin. Buffer solutions were filtered through a 45-µm filter before use. Samples were injected by hydrodynamic injection for 5 s at 0.5 psi to the capillary maintained at 20°C, followed by electrophoretic separation at 18 kV. Electropherograms were recorded at 200, 250 and 290 nm at a sampling frequency of 4 Hz. Thirty-two different samples replicated 3 times were analysed.

Chemometrics methods

Data preparation

Baseline correction and peak detection

In order to detect peaks and quantify their area, profiles at the three wavelengths are summed and the total profile (an example is shown in Fig. 1) is subjected to preliminary baseline correction using a fit of the baseline regions to a linear model and subtracting this model from the data.

Peaks are then detected using the Chromint software developed by the University of Amsterdam [22], in which detection is based on the analysis of the profile of the first derivative of the signal calculated with a Savitzky–Golay filter. This provides a list of peaks and their position with typically around 25 peaks detected per electropherogram.

Sample matrix

The next step is to produce a sample matrix $\boldsymbol{S}$ in which rows refer to samples and columns refer to peak areas, detected along the total profile, of chemical species that are common to a significant proportion of the samples [13, 23].

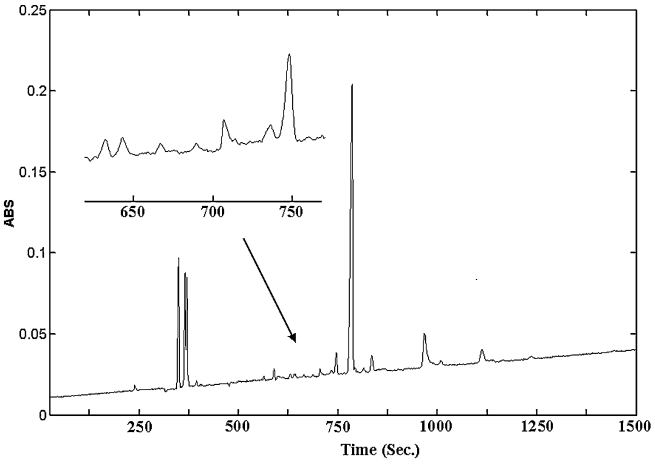

Fig. 1 Total profile of the first replicate of sample A104 (acute dose of 0.75 mg kg^{-1})

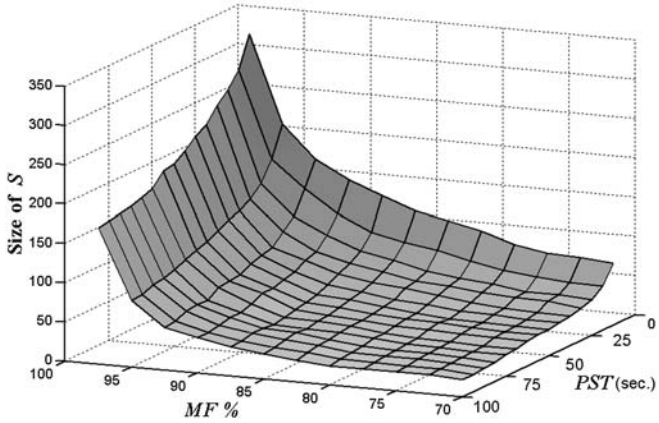

Fig. 2 Size of the sample data matrix S as function of the matching factor $MF\%$ and peak shift tolerance PST

After appropriate post-processing, this matrix can then be used as input to any of the pattern recognition algorithms.

In order to construct the matrix, we use the relative intensity at each of the three wavelengths to determine which peaks are common to the electropherograms. The intensities at the three wavelengths (200, 250 and 290 nm) and the maxima of each peak are measured and their relative proportions calculated according to Eq. (1):

$$^{N}x_{p\lambda} = \frac{x_{p\lambda}}{\sum x_{p\lambda}} \quad (1)$$

where $x_{p\lambda}$ is the original intensity of peak p at its apex in the electropherogram and at wavelength λ and $^{N}x_{p\lambda}$ its normalised value. This step allows the comparison of different sample profiles. A match factor MF for two peaks l and m belonging to two different samples and within a peak shift tolerance in time, is formulated as follows:

$$MF_{lm} = \left(1 - \frac{\left(|x_{l,200} - x_{m,200}| + |x_{l,250} - x_{m,250}| + |x_{l,290} - x_{m,290}|\right)}{2}\right) \times 100 \quad (2)$$

A complete overlap would imply MF=100, while the opposite, theoretically when two peaks absorb uniquely at two different wavelengths, yields MF=0. If MF exceeds a predefined threshold, the two peaks are assumed to arise from the same compound and their peak areas are placed in the same column of $\boldsymbol{S}$. Otherwise a new column is added because a new species has been found. The number of columns in $\boldsymbol{S}$ critically depends on both the threshold selected for MF and peak shift tolerance (PST). Lower PST means looking for matches in a narrower window along the profile, while higher MF means being more restricted in terms of matching; hence, under these conditions a larger sample matrix size is expected (Fig. 2). Some aspects that may help to determine the optimal trade-off for MF and PST are: a) the experimentally observed peak shift; b) the observed range of variation for MF; c) how many species common to all samples are expected; d) how many species are expected in total. The last two points may for example derive from experience in a previous analysis carried out on a similar type of samples.

Notice that the procedure as formulated does not aim at identifying peaks, but is a measure of similarity; therefore, the use of only three wavelengths suffices. Indices comparable to MF such as the correlation coefficient or normed Euclidean distance could be applied if more wavelengths were available; however, in this particular application we are confident that we have correctly matched the peaks in the electropherograms.

Variable and sample selection

Of the all unique peaks detected, many occur in only a small number of samples and therefore are not suitable for discrimination because they are unable to describe or model trends in one of the two classes. Hence, only those peaks occurring in more than a half of the samples are retained for further processing. This results in a major reduction in the number of columns in $\boldsymbol{S}$.

In a small number of samples (less than 10%) less than half the peaks identified are detected, therefore these samples are removed because there is insufficient data for discrimination. This results in a further reduction of the number of rows in the sample matrix.

Normalisation

Finally, the data matrix $\boldsymbol{S}$ consisting of the areas of the selected peaks along the total profile undergoes post-processing. Selective normalisation is applied along rows according to the Eq. (3):

$$^{n}s_{ij} = \frac{s_{ij}}{\sum_{j} s_{ij}} \quad (3)$$

where $s_{i,j}$ is the area of peak referring to species j for sample i. Subsequently standardisation is applied along columns:

$$^{std,n}s_{i,j} = \frac{^{n}s_{i,j} - \overline{^{n}s_j}}{^{n}std_j} \quad (4)$$

where $\overline{^{n}s_j}$ is the mean value of the normalised peak area of the species j over all samples and $^{n}std_j$ the corresponding population standard deviation. This operation is essential because it minimises the risk of the final result being dominated by those peaks occurring in higher quantities, hiding the potential contribution of minor peaks. Figure 3 shows the flow chart of all the steps needed before performing pattern recognition.

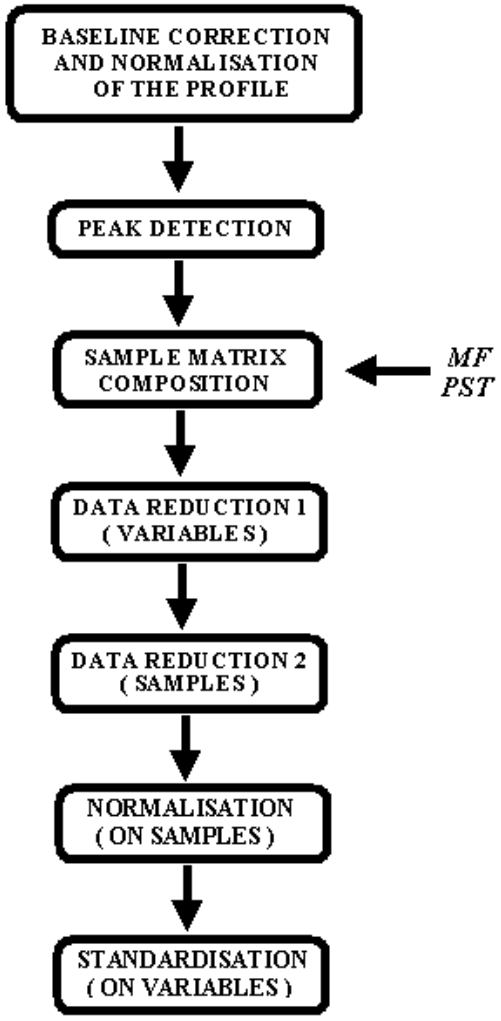

Fig. 3 Flow chart of the pre-processing phase developed for preparing CE data to pattern recognition

Pattern recognition techniques

Methods of pattern recognition can be mainly divided into two categories, namely unsupervised and supervised techniques [24]. The former focuses on investigating the structure in the data, if there are detectable features that give a major contribution, if there are similarities among samples or whether there is presence of outliers. They do not require information about class membership because their aim is not to define a classification rule. Examples are principal components analysis [25] and hierarchical agglomerative cluster analysis [26]. In contrast, supervised methods require information on class membership because their purpose is explicitly to build up a classifier. They are usually constructed using a portion of the samples available (training set), and their performance tested using the remaining ones (test set). Examples are k-nearest neighbour (KNN) [24], discriminant analysis [27] and support vector machines [28, 29, 30, 31]. While KNN and DA are very well known techniques that are widely used because they easy to apply, the use of SVMs is not widespread yet, in spite of its many attractive features. Primarily, the specific theoretical foundation of the method (structural risk minimisation principle) equips SVMs with greater generalisation ability and justifies their employment especially when there are few samples available for classes modelling. This relates to the case under investigation, as only a total of 20 samples replicated 3 times are available. Other attractive features depending to the structure of the embedded SVMs optimisation problem are: a) its robustness when working with datasets having low sample/variables ratios; b) the generation of a completely reproducible solution in reference to the parameters of the classifiers; c) the possibility of drawing class boundaries of various complexity by replacing the kernel, which represents the "core" of the function to optimise. For the case described in this application, feature a) is particularly relevant because only replicates of 10 samples will be selected for modelling each of the classes against a number of variables equal to 11. In fact, DA is known to underperform in such circumstances, hence requiring a prior step of variable reduction. In contrast, features b) and c) may justify the use of SVMs in most complex situations, as an alternative to neural networks (NN), as this latter technique has known drawbacks in terms of reproducibility and generally needs a greater amount of training samples. However, despite these multiple advantages, some studies report how SVMs may not automatically provide the best solution particularly when dealing with well-behaved data distributions [17, 32]. Because in principle prior information on the complexity of the data may not be available, an approach using in parallel DA and SVMs is preferred, and the outcomes of both the techniques tested and compared.

Principal components analysis

PCA is a widespread multivariate technique that decomposes the data matrix $\boldsymbol{S}$ into scores $\boldsymbol{T}$ and loadings $\boldsymbol{P}$, accordingly to the relation:

$$\boldsymbol{S} = \boldsymbol{T}.\boldsymbol{P} \quad (5)$$

While $\boldsymbol{T}$ describes trends among the samples, $\boldsymbol{P}$ describes trends among variables. Scores and loadings are ordered in terms of quantitative significance that is proportional to their eigenvalue, which can also be represented more straightforwardly as %variance or %information. Hence PCA can be employed as a method of variable reduction by retaining the first few principal components. Moreover, visual inspection of the plots generated using scores and loadings provides a quick and highly informative means for investigating relationships between samples and variables respectively, and may be helpful for the identification of clusters related to a certain feature or for detecting potential outliers.

Hierarchical agglomerative cluster analysis

This technique is often used for exploratory purposes because it allows the visualisation of the overall result in the form of a dendrogram that illustrates the relationships between samples. The procedure starts by identifying a number of groups equal to the number of samples that form the dataset, and through an iterative procedure it merges the clusters that are most similar, until reaching a unique cluster collecting all the instances in the final step. If one wants to identify specific clusters, a prior level of similarity or distance can be set as threshold, after which the procedure stops. The adoption of different methods for measuring the similarity amongst samples and for linkage may lead to different results; therefore, it is advisable to compare the outcomes using several approaches. A method to measure the quality of a clustering solution can be by means of the cophenetic correlation coefficient that is the correlation between the original distances between the samples and those that result from the cluster configuration. This coefficient varies in the range (0,1) and values above 0.75 are usually regarded to be good. Details on the algorithm are described elsewhere [24].

Discriminant analysis

Discriminant analysis by means of Mahalanobis distance is a common tool for assigning the class membership to unknown samples. The Mahalanobis distance of sample i to class A is defined by Eq. (6):

$$d_{i,A} = \sqrt{(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_A).\boldsymbol{Q}_A^{-1}.(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_A)'} \tag{6}$$

where $\boldsymbol{x}_i$ is the sample to classify, and $\bar{\boldsymbol{x}}_A$ and $\boldsymbol{Q}_A$ are the centroid and variance–covariance matrix of class A respectively. This equation is computed for each class in the dataset and the sample assigned to that one for which the distance is the lowest. In comparison to the Euclidean distance from the class centroid, which is the simplest form of classification, the Mahalanobis distance offers the advantage of taking into account the dispersion of one class by implementing the variance–covariance matrix in its formulation. However, a major drawback is that the ratio of samples to variables has to be high for this technique to work well, and in any case the number of variables cannot exceed the number of samples because the variance–covariance matrix would not have an inverse. Hence a method of variable reduction is often adopted prior to the use of DA, with a possibility being to use the scores obtained from PCA after estimation of the optimal number of components for modelling the data instead of the raw readings.

Support vector machines

The SVMs algorithm derives the classification rule using only a fraction of the samples in the dataset, which are called support vectors (SVs) and are typically those lying nearby the class boundary. In its simplest form (SVMs dot product), the rule for classifying a new instance $\boldsymbol{x}$ is explicitly expressed as function of the support vectors, by means of Eq. (7):

$$y = \mathrm{sgn}\left(\sum_{i=1}^{N_s} \alpha_i y_i \boldsymbol{s}_i x + b\right) \tag{7}$$

where y (=±1) is the class label assigned that depends on the sign of the function within the parenthesis. In the equation, $\boldsymbol{s}_i$ is a generic SV and the coefficients α_i and b are the solution to an optimisation problem. The absolute value of the function represents a kind of confidence level, since the higher its value, the stronger the SVMs algorithm believes one sample to belong to a specified class. In this

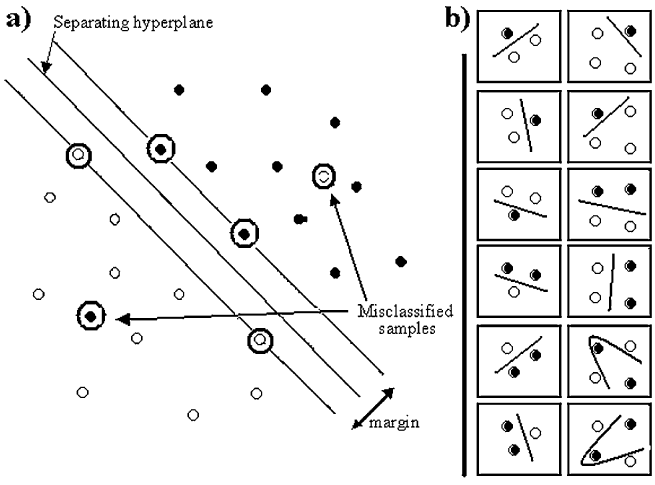

Fig. 4 **a** The SVMs optimisation problem. The optimal separating hyperplane is that one maximising the distance between the class boundary and the support vectors. SVs (*circled*) are those samples both lying on the margin and misclassified as no hyperplane can spot them in the proper class, because data are not completely linearly separable. **b** VC dimension of a classifier: this is defined as the maximum number of training samples that can be spotted in their proper class, given any arbitrary class labelling. In 2 dimensions, given 3 training samples, a line suffices to correctly separate the instances for each of the 6 possible combinations. With four training samples there will be two cases for which a classifier of higher complexity will be required, which will correspond to a higher VC dimension

form, SVMs determine the class boundary as the optimal separating hyperplane between the classes, in the space of the original variables. Theoretically, infinite hyperplanes may possibly be found for separating the classes, but the best is regarded to be the one that maximises the distance (or margin) between the class boundary and the support vectors (Fig. 4). The corresponding optimisation problem for determining the parameters of the optimal separating hyperplane is formulated as follows:

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i} \boldsymbol{x_j}^T \tag{8}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are two generic sample data vectors in the training set, y_i y_j their labels (±1), and the function L_D has to be maximised with respect to the coefficients α. These coefficients will directly determine the disposition of the class boundary, and will be non-zero only for the support vectors, namely those samples lying at the very border between the two classes. The higher the coefficient, the bigger the influence of the sample on the determination of the class boundary. Those samples lying away from the boundary, which may form the majority of cases, will have α=0 and their removal will not influence the final solution. Hence, the number of support vectors is a first relevant diagnostic parameter because it provides a rough idea of the separability between the classes (fewer SVs means fewer samples near the boundary). Besides, their number practically represents an upper bound to the validation error when using a leave-one procedure: in fact, the samples that are not termed as SVs are certainly spotted in the correct class and their removal in validation will not change the structure of the classifier, hence generating the same result. Notice that the optimisation problem as formulated in Eq. (8) incorporates most of the attractive features of the SVMs method as follows:

a) The input vectors $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ appear in the equation only in the form of a scalar product. This reduces the deteriorating effect of data dimensionality and makes this an advisable approach when the ratio of samples to variables is low, where other classic techniques may fail (e.g. DA). As a consequence, no prior step of variable reduction needs to be implemented.

b) The equation to optimise has a convex profile that shows only one minimum, which makes the final solution completely reproducible. A major drawback when using other approaches such as neural networks is that the profile of the function to optimise for estimating the parameters often shows local minimums. The solution is therefore not completely reproducible because the algorithm for optimisation can get stuck in different minimums resulting in different values for the parameters of the classifier [33].

c) It is possible to substitute the simple dot product of the input vectors with a product of other functions (kernel functions) that leave the nature of the optimisation problem unaltered but allow the determination of complex class boundaries. When dealing with composite data distributions this can be a powerful feature to exploit.

In addition to these features, the use of SVMs is however properly justified because the optimisation problem in Eq. (8) relates directly to the structural risk minimisation principle (SRMP), which represents the theoretical foundation of the method. This principle equips SVMs with greater generalisation ability and make this approach attractive especially when the amount of information available is little, namely when there are few samples available. SRMP is formulated as follows:

$$\in \leq \in_N + \sqrt{\frac{d_{VC}\left(\log\left(\frac{2N}{d_{VC}}\right)+1\right) - \log\frac{\eta}{4}}{N}} \tag{9}$$

where $\in$ is the generalisation error (the minimum achievable errors), $\in_N$ is the training error, and d_{VC} is the Vapnik–Chervonenkis dimension that relates to the complexity of the classifier (Fig. 4). This principle defines a theoretical upper bound to the $\in$ error corresponding to a probability of $1-\eta$. According to the formulation, the discrepancy between $\in$ and $\in_N$ will decrease using a higher number of samples, namely with using more information, and will decrease with using a more complex classifier because of a higher risk of over-fitting the training data. SVMs exploit this principle, since the algorithm is focussed on minimising the entire left-hand member of the SRMP formulation, rather than minimising the training error $\in_N$ (empirical risk minimisation principle) as all the other supervised pattern recognition algorithms normally do. Thus, the optimal solution may correspond to a non-minimum for the training error $\in_N$. Equation (8) relates to SRMP because this is reached trough the learning algorithm by maximising the margin which itself constitutes an upper bound to the d_{VC} dimension of the classifier. According to the SRMP formulation, the maximisation of the margin in the SVMs learning procedure generates a classifier with lower d_{VC} (the whole quantity under square root on the right member of Eq. (9) reduces) and greater generalisation ability.

Computation

CE sample profiles were acquired in ANDI format and transferred to Matlab 6.0 (MathWorks Inc.) using The NetCDF Toolbox [34]. Peak detection and quantification was carried out using the chromatographic integration software Chromint [22], and SVMs was performed using the OSU-SVM Classifier Toolbox [35]. All remaining processing were carried out by means of in-house Matlab routines.

Results

The 96 samples were split into a training set and a test set of 60 samples (20 replicates 3 times) and 36 samples (12 replicated 3 times) respectively. Samples of training set were used to identify the peaks of the species to retain in the analysis as columns of the sample data matrix $\boldsymbol{S}$ and

Table 2 Average retention time (RT) and sample frequency of the peaks used as variables to describe the system

Peaks		Training set		Test set	
No.	Average RT (s)	Freq. (no.)	Freq. (%)	Freq. (no.)	Freq. (%)
1	342.22	60	100.00	36	100.00
2	352.81	60	100.00	36	100.00
3	580.35	50	83.33	25	69.44
4	620.01	33	55.00	25	69.44
5	667.50	31	51.67	16	44.44
6	687.52	34	56.67	17	47.22
7	695.91	43	71.67	23	63.89
8	735.31	54	90.00	36	100.00
9	786.74	41	68.33	24	66.67
10	923.51	54	90.00	29	80.56
11	1,073.50	37	61.67	27	75.00

Table 3 Outcomes of PCA extraction and discriminant factors calculated for each PC score both for the training and the test set, in reference to the two classes of interest. *DF%* is not calculated for PCs of highest order because they span a minimal amount of system variance

N. PCs	E-value	Var. %	Cum. Var. %	$DF_{training}$ %	DF_{test} %
1	172.30	28.48	28.48	29.21	90.77
2	136.41	22.55	51.03	1.35	11.81
3	102.34	16.92	67.94	26.51	0.03
4	52.63	8.70	76.64	7.66	9.44
5	46.38	7.67	84.31	7.13	0.44
6	28.81	4.76	89.07	2.75	19.05
7	26.69	4.41	93.48	1.84	7.05
8	21.39	3.54	97.02	–	–
9	11.26	1.86	98.88	–	–
10	6.78	1.12	100.00	–	–
11	0.00	0.00	100.00	–	–

for the construction of the classifiers. According to the procedure outlined in the Section "Sample matrix", visual inspections of the profiles and analysis of the variation range for matching suggested to set *MF*=90% and *PST*=87.5 s as optimal trade-off. Particularly, this value for *PST* roughly corresponded to the maximum delay observed in the release of similar peaks among replicates. This led to the formation of an initial sample data matrix for the training set $\boldsymbol{S_{TRAINING}}$ 60×51 (samples×peak areas). The majority of peaks were removed because they appeared in less than 30 training set samples, leading to a reduction of the columns from 51 down to 11. A minor fraction of samples also had to be removed, because less than 6 out of the 11 peaks used to describe the system had non-zero peak areas, leading to a final size of $\boldsymbol{S_{TRAINING}}$ of 55×11. Finally, in order to prepare the data for pattern recognition, the sample matrix was normalised along rows and standardised along the columns. Subsequently, a second matrix $\boldsymbol{S_{TEST}}$ was built up: the same 11 peaks identified in the training set were retained as descriptors of the system, and the data standardised according to the mean and population standard deviation of the variables in $\boldsymbol{S_{TRAINING}}$. For three samples less than 6 non-zero peak areas were found; hence, these were removed and the matrix $\boldsymbol{S_{TEST}}$ reduced from a size of 36×11 to a size of 33×11. Table 2 shows average retention time and sample frequency of the peaks retained for both the datasets.

Explorative analysis

Principal components analysis

Table 3 shows the outcomes of PCA extraction on the training set. The first PCs that span the majority of variance in the system can be used to visualise major trends in the dataset. However, further PCs may still be important because they are possibly associated to the classes of interest. The optimal combination of PC scores for visualisation may be determined manually by trying several arrangements or with the help of some parameters that quantify the separability of the classes along each score. For instance, a discriminant factor (*DF*) based on the analysis of the variance can be calculated as follows:

$$DF^i\% = \frac{SS^i_{TOT} - \left(SS^i_{WAc} + SS^i_{WCh}\right)}{SS^i_{TOT}} \times 100 \qquad (10)$$

where i refers to i-th PC score, SS_{WAc} and SS_{WCh} are the within groups sum of squares for the two classes (Ac= acute, Ch=chronic), and SS_{TOT} is the total sum of squares. Higher DF^i implies a higher variation when passing from a group to another, hence indicating the i-th score to be more useful for distinguishing the classes. When using more scores at the same time, it must also be verified that the selected ones discriminate between complimentary parts of the system, which can be done by setting up a stepwise procedure if the number of possible combinations is relatively high.

For the training set, Table 3 indicates that the scores of PC2 poorly contribute (*DF%*=1.35), while the scores of PC1, PC3 and PC4 that have the higher *DF%* are more appropriate (Fig. 5). Here the two classes can be distinguished with Ac cases forming a major cluster on the centre-left and Ch cases more dispersed on the centre-right. However, a consistent part of the information in the dataset is not represented in the plot, mainly because of the exclusion of PC2 that alone represents roughly a quarter (22.55%) of the whole amount of information available. This means that while the two classes can be spotted, this feature in not clearly dominant in the training set.

A similar analysis can be conducted on the test set. Scores are computed by projecting these samples in the principal component space of the training set through the respective loadings matrix rather than applying PCA directly on the test set, because the training set is employed for constructing the model. This also allows a more consistent comparison between the distributions of the two subsets of data. Table 3 emphasises how various PCs can

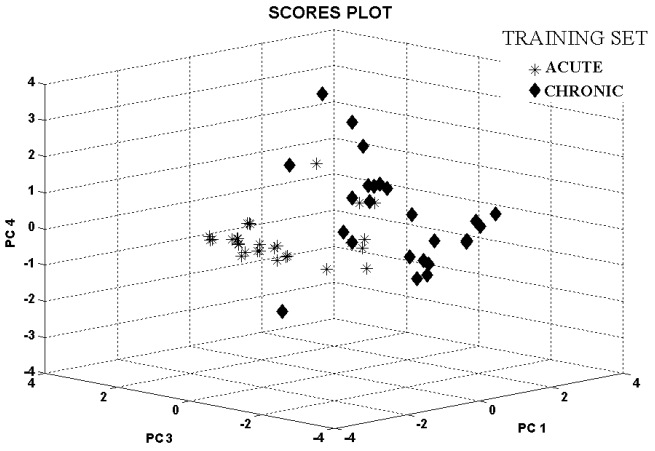

Fig. 5 scores plot of the training set in the space of the components 1, 3 and 4 that together account for the 54.1% of variance in the dataset. Most of the instances of class Ac distribute in a major cluster on the *centre-left* with fewer samples disposed more randomly in the *centre*. Instances of class Ch show higher dispersion

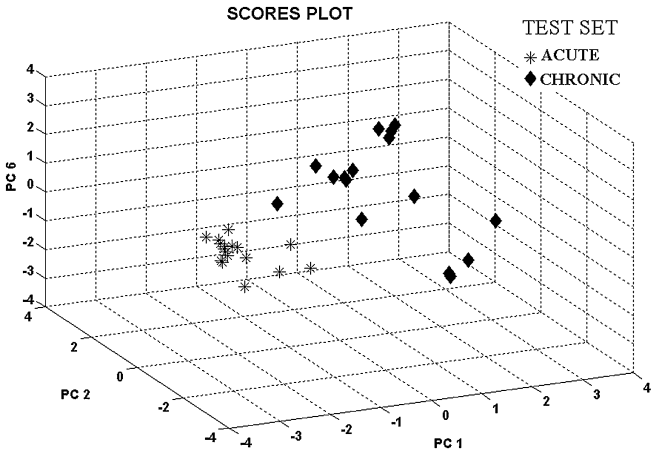

Fig. 6 Scores plot of the test set projected in the space components 1,2 and 6 of the training set. These components together account for 55.79% of variance in the dataset. Here the two classes of interest can clearly be spotted in two different clusters, with class Ch, on the *centre-right*, again characterised by higher dispersion

be considered for visualisation as they display a rather high *DF%*. These values suggest the scores plot generated by using PC1, PC2 and PC6 as optimal combination (Fig. 6). Here the classes can be determined even more clearly than in the previous plot, with class Ch still showing higher dispersion than class Ac. Noticeably, there appears to be some degree of heterogeneity between the samples in the two subsets of data, because the classes in the test set seem more separable. This emerges by comparing the two scores plots as well as by comparing the *DF%* for the same PC score, which in the majority of cases is much higher for the test set.

Hierarchical agglomerative cluster analysis

HACA was performed using both the Euclidean and the Mahalanobis distance for measuring the similarity among couples of samples and with average linkage for merging the clusters. Figure 7 shows the dendrogram for the training set using the Euclidean distance, which resulted in a cophenetic correlation coefficient equal to 0.908 (the Mahalanobis distance generated similar outcomes with a correlation value of 0.886). First, it is possible to observe that various triplets corresponding to replicates. A few replicates cluster in pairs with the third sample appearing in a close neighbouring cluster. However, in a few cases this expected feature does not emerge. This may be due either to the experimental signal of the sample that significantly differs from other replicates, or to a fault within the peak detection routine. The outcome of HACA can provide an important feedback both for the experimental and the data processing phases. Another major feature that emerges is clustering of samples according to the class membership. For example, a cluster collecting 21 instances of class A (from A210 A to A307C) is represented in the top of Fig. 7. Hence, HACA further confirms that the type of dosage is a feature that clearly contributes to the structure of the dataset. However, the level of the dosage does not appear as a clear feature determining the formation of sub-clusters.

Classification

Discriminant analysis

For this technique to work well, the number of samples must be significantly larger than the number of variables. When the number of variables exceeds the number of samples, the variance–covariance matrix has no an inverse and consequently the Mahalanobis distance cannot be computed. There are 30 samples for each class in the training set, but these are reduced down to 10 if replicates are not considered, while the number of variables equals 11. Under such circumstances it is preferable to use a variable reduction by PCA, in which the first few scores are the input to the classifier. The estimation of the optimal number of components to retain in the model is performed by setting up a cross-validation procedure: one sample is removed from the training set, and the model using a defined number of PC scores is constructed using the remaining samples. The prediction ability of the model is tested on the sample left out, and the procedure repeated for all samples in the dataset and for each possible model constructed using a different number of PC scores. Table 4 shows that the minimal number of errors occurs with using either 6 and 7 PCs. Six PCs are preferred because they generate a slightly simpler model. The centroids of the two classes and the variance–covariance matrices are then calculated using the full training set. Figures 8 and 9 show the distance plot for the training and the test set respectively, in which the dotted bar represents the class bound-

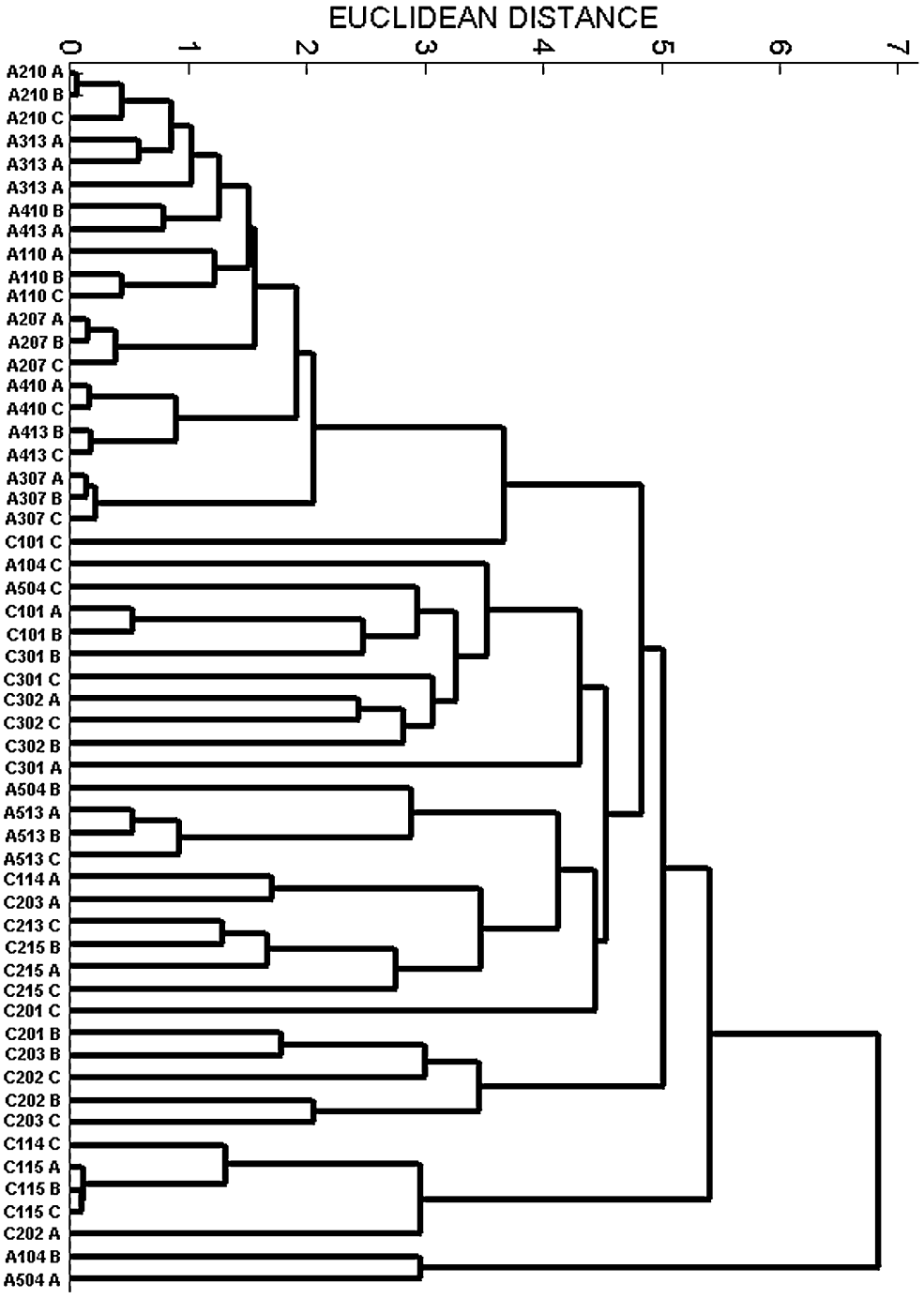

Fig. 7 Dendrogram on the instances of training set, generated using Euclidean distance and average linkage. Many replicates cluster at the earliest stages forming triplets and duets, indicating an acceptable level of reproducibility. Major clusters are formed according to the class membership, while the dosage level within each class does not seem to emerge as a feature that contributes to clustering

ary. This type of plot can be theoretically divided in four sub-regions: top left and bottom right, in which samples are assigned to one class with high degree of confidence as the difference between the two distances is significant; bottom left, in which assignment is ambiguous as samples may fit both classes well; and finally top-right is a region where potential outliers may occur, as they badly fit both the classes [24].

For the training set, the classifier results in 3 errors (94.55% correctly classified). Quantitatively the performance is rather satisfactory because especially many of the instances of class Ch lie rather far from the boundary, on the bottom-right. The performance on the test is slightly

Table 4 Number of errors and % of correct classifications for a DA model implementing an increasing number of PC scores as input, by using a leave-one-out method as procedure for validation. This suggests using a model with 6 PC scores

PCs no.	No. Err.	% C.C.
1	18	70.00
2	9	85.00
3	11	81.67
4	9	85.00
5	8	86.67
6	5	91.67
7	5	91.67
8	7	88.33
9	7	88.33
10	7	88.33
11	7	88.33

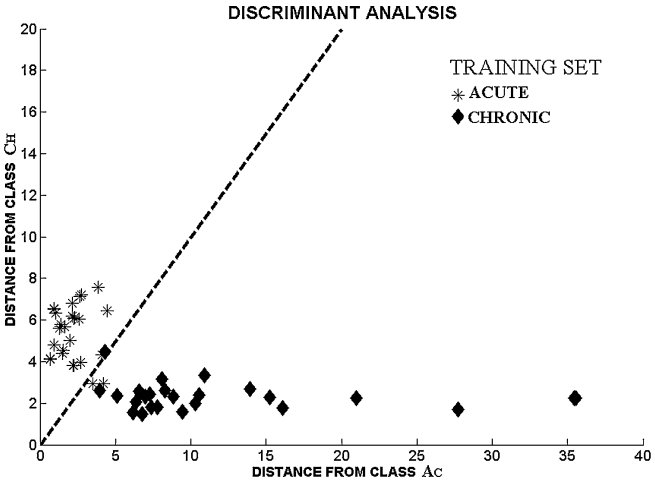

Fig. 8 Plot of the Mahalanobis distance from class centroids on the training set. Samples are well separated on the basis of the class membership with good level of confidence especially for the instances of chronic Ch that lie further away from the boundary. Samples misclassified for class Ac correspond to those lying away from the major cluster on the *centre-left* of Fig. 5

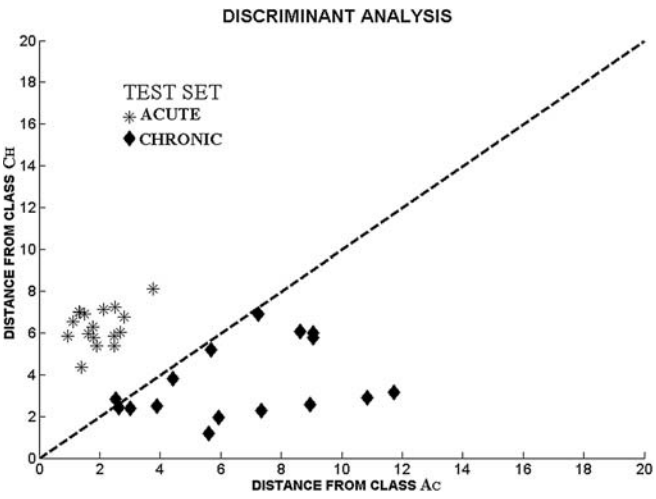

Fig. 9 Plot of the Mahalanobis distance from class centroids on the test set. Qualitatively the performance improves with only one sample misclassified because classes show simpler distribution (Fig. 6). However the overall level of confidence decreases in comparison to the training set of Fig. 8, as samples lie closer to the boundary

improved. Only a sample corresponding to C315 is misclassified (96.97% correctly classified). In contrast, the overall level of confidence worsens, as in the plot of Figure 9 samples tend to be closer to the class boundary. The improved performance on the test set can be interpreted in light of the scores plot in Figs. 5 and 6: DA is a classifier that works well when the overall data distribution is fairly simple (e.g. two partially overlapped classes where data are normally distributed). When applied to the test set, the model improves its performance because the simpler data distribution (Fig. 6) matches with simple class boundaries that DA draws between the classes. However the overall worsening in confidence depends on the fact that the model is trained with the instances as in Fig. 5, which is partly heterogeneous to the test set.

Support vector machines

If from exploratory analysis (HACA and PCA), the distribution of the data is thought to be not too complex, it is preferable to rely on the simplest form of SVMs (dot product), without employing more sophisticated SVMs classifiers based on the replacement of the kernel in Eq. (7). The SVMs dot product will be able to operate in the space of the original variables of the sample matrix $\boldsymbol{S}$ possibly picking up structure in the data that DA would be unable to model so well, resulting also in a comparably easily understood classification rule [Eq. (6)]. Moreover, this form of SVMs classifier is rather easy to manage, as it requires the adjustment of only one parameter, which is the penalty error C. This bounds the coefficients of the optimisation problem according to:

$$0 \le \alpha_i \le C \tag{11}$$

where C is inversely proportional to the error tolerance for the training samples: including no error (no upper bound to the α multipliers) corresponds to find a hyperplane that minimises the training error. However this may result in a very narrow margin that indicates poor generalisation ability. Generally, we want to trade-off between these two aspects, and therefore it is sensible to include some error tolerance. A possible way of tuning this parameter is to compare different classifiers by implementing a validation procedure similar to the one described for the selection of the scores in Section "Discriminant analysis" under "Classification". The classifier with best prediction ability would then be selected. A second possible approach is to consider the profile of the number of support vectors as function of C: in practical applications it is observed

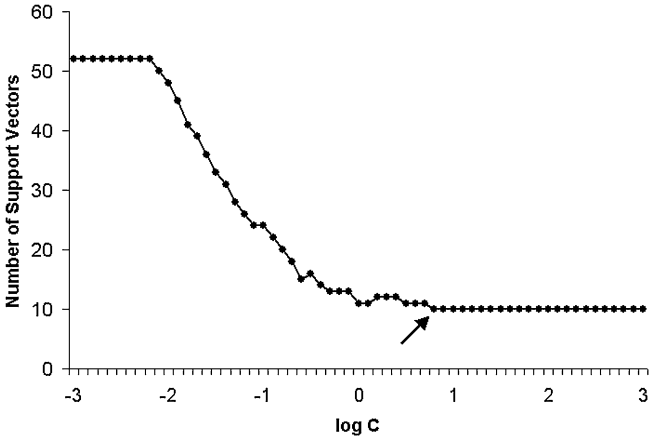

Fig. 10 Profile of the number of SVs as function of the penalty error C. Smaller C implies lower number of SVs because the margin gets narrower, involving less samples in the determination of the optimal separating hyperplane. The *arrow* indicates the most advisable value

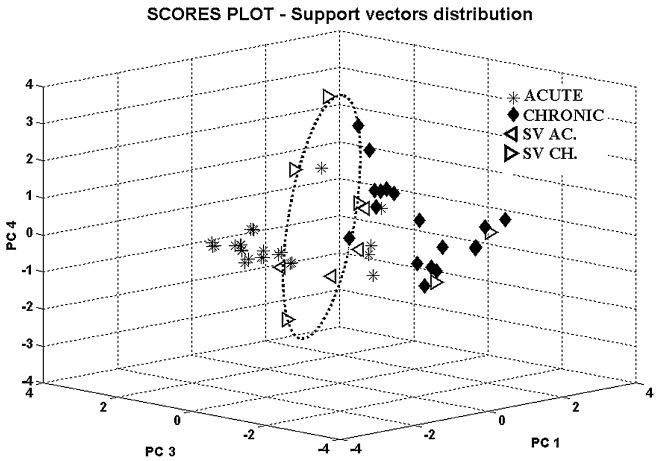

Fig. 11 SVs selected in the training set, which will determine the structure of the SVMs dot product classifier. Notice that they dispose mainly along the inter-class space, qualitatively indicated by the *dotted oval*. For class Ac, only one SV falls in the cluster on the *centre-left* where the majority of samples of this class lie. The remaining three are involved in modelling the smaller portion of samples on the *centre-right*

that the number of SVs decreases as C increases, until a minimum is maintained when a certain threshold for C is overcome. A good choice may be this threshold because it corresponds to the lowest penalty error value (that means wider margin and better generalisation ability according to SRMP) given the lowest number of SVs, which indicates a good separability between the classes because fewer samples lie nearby the boundary. Figure 10 shows this relation, suggesting an optimal value of $\log C$=0.8 corresponding to 10 SVs out of 55 samples. The number of SVs indicates a good separability of the classes as they compose only 2/11 of the training set. Even though the class boundary is calculated in the space of the original variables, the projection of the samples in the scores plot of Fig. 11 illustrates how the support vectors mainly span along the interclass space, consistently with the theoretical formulation of the method.

This is also consistent on the experimental side, because these samples correspond to the instances at higher dosage for class Ch and at lower dosage for class Ac that are likely to be the most similar, thus lying nearby the boundary (Table 5).

Also it is interesting to notice how for class Ac, only one of the four SVs falls in the major cluster on the left. This emphasises how SVMs strives in modelling the remaining samples that spread on the centre-right of the scores plot. As a result, when checking the performance on the training set, perfect classification is obtained. Figure 12 reports the value of the decision function of Eq. (6): this takes a sign according to class membership, expressing also a relatively good level of confidence for the vast majority of the instances. Figure 13 reports the results on the test set. The performance is much poorer and all the misclassifications (8 for 75.76% in overall performance) are concentrated in class Ch. This can be qualitatively explained by the partial lack of homogeneity between the

Table 5 Distribution of SVs within each class in relation to the dosage level. According to theory, SVs are those samples lying nearby the boundary that, practically, correspond to those with higher dosage levels for class Ch and lower level for class Ac (in fact, levels of 1.125 mg kg^{-1} and 1.5 mg kg^{-1} for Ac and 1 mg L^{-1} for Ch are not represented). The corresponding α coefficients of the separating hyperplane also indicate the most influential samples. This further confirms this trend as they correspond to C101 C and A413 A

SV	α	Dosage level
A104 C	2.65	0.75 mg kg^{-1}
A104 B	2.23	0.75 mg kg^{-1}
A504 C	1.68	0.75 mg kg^{-1}
A413 A	6.31	0.375 mg kg^{-1}
C101 C	5.67	7 mg L^{-1}
C213 C	2.76	7 mg L^{-1}
C301 B	1.97	7 mg L^{-1}
C202 B	1.79	3 mg L^{-1}
C302 A	0.59	3 mg L^{-1}
C114 A	0.087	3 mg L^{-1}

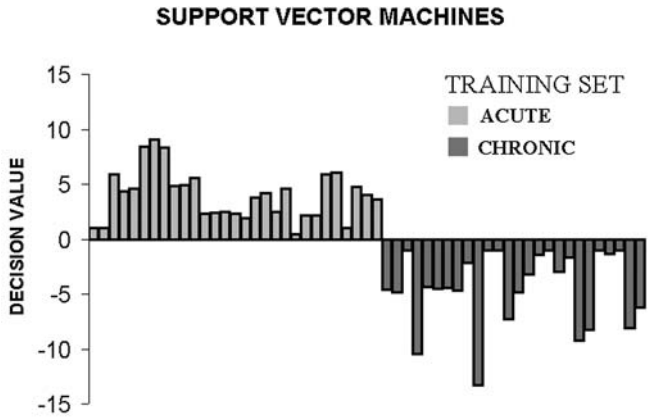

Fig. 12 Values of the SVMs decision function as in Eq. (6) for the training set. All samples are correctly classified (sign of the function) because SVMs strive to model this set of data nearby the boundary, where DA fails. Lower confidence is shown for instances lying in this region (support vectors)

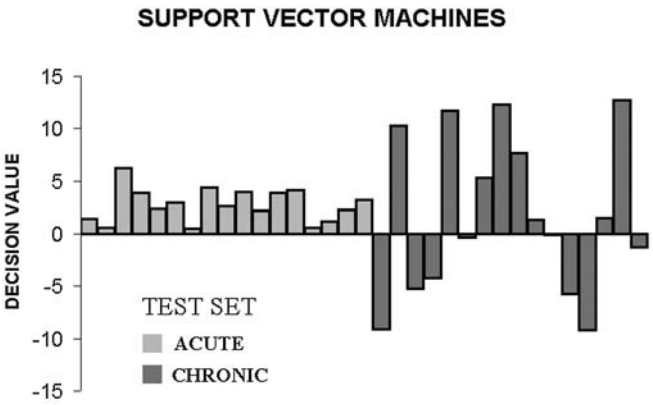

Fig. 13 Values of the SVMs decision function as in Eq. (6) for the test set. The SVMs classifier fails to correctly assess the class membership of class Ch where all the errors are concentrated

two subsets of data, as emphasised in the Section "Principal components analysis" under "Explorative analysis", and by the fact that class Ch shows higher dispersion in both the subsets. Its instances in the test set seem to be affected more by the particular distribution of the support vectors of class Ac.

Conclusions

Unsupervised pattern recognition techniques showed that the type of cadmium administration, either acute or chronic, has an important influence on the data structure that can clearly be identified in urine profiles obtained by capillary electrophoresis. The scores plots (Figs. 5 and 6) suggest the presence of a major cluster related to instances of class Ac but with instances of class Ch showing higher variability. HACA (Fig. 7) also suggests major clusters in relation to class membership. However, it does not result in the formation of sub-groups according to dosage level. This sub-feature within each of the two classes is much harder to spot, but this may be partly explained with the limited size of the dataset, where only a maximum of 4 samples were implemented for each dosage level (Table 1). Moreover, HACA suggests an acceptable level of reproducibility for the experimental methodology and the computational approach: replicate samples often cluster showing high similarity. Major differences in a few cases may be either due to an actual difference in the profile or to a fault in the peak detection routine. In this way HACA provides useful feedback because, if considering a higher number of replicates in further analysis (a number of three is not sufficient for defining a sample reference profile), those clustering afar from the majority might be removed.

As far as supervised pattern recognition is concerned, both DA and SVMs show that it is possible to discriminate samples on the basis of the class membership. A substantial difference is that DA is applied on sample matrix $\boldsymbol{S}$ after variable reduction by means of PCA (the first 6 PC scores are retained to feed the classifier); SVMs are applied on $\boldsymbol{S}$ in the space of the original variables. The two techniques lead however to complementary results because while SVMs outperform DA on the training set ($C.C._{DA}$=94.55% versus $C.C._{SVM}$=100%), DA performs better on the test set ($C.C._{DA}$=96.97% versus $C.C._{SVM}$= 75.76%). This can be explained by the partial lack of homogeneity between the two sets of data, on which the two scores plots supply an insight: SVMs perform better on the training set because this approach actively models the small fraction of samples belonging to class Ac that are far from the major cluster on the centre-left in Fig. 5. According to SVMs theory, the support vectors that define the classification rule are found along the interclass space and in fact three out of four of the SVs for class Ac are taken from the centre-right region of the scores plot, with the remaining one located in the cluster on the centre-left (Fig. 11) where the majority of samples for this class lie. DA is unable to model this distribution so well. Significantly, some of the misclassified instances of DA in the training set correspond in fact to support vectors. In contrast, the test set shows a simpler distribution (Fig. 6), and DA performs better because the classifier in the training phase is equally influenced by all training samples, the majority of which do not lie along the class boundary. Note also that all the misclassified instances for SVMs on the test set belong to class Ch. This can qualitatively be explained by the higher dispersion of this class, whose instances are affected by the particular distribution of the support vectors for class Ac on the centre-right.

Other studies [17, 32] suggest that SVMs may not automatically provide the best solution when dealing with simple data distributions. However, prior information is often not available because exploratory analysis (e.g. with PCA or HACA) is not performed, or factors that are out of control (e.g. the appearance of other analytes in urine profiles) may lead to a sudden increasing in complexity of the data. Hence, especially when a limited number of samples are available for constructing the classifier, the use of SVMs is justified because of their greater theoretical generalisation ability due to the embodied SRMP and their robustness to low sample-to-variable ratios that allows working in the space of the original variables. Prior steps of variable reduction may result in omitting relevant information on the system, and the determination of these features (e.g. number of PCs to model the data) can sometimes be ambiguous. In case the complexity of the data sensitively increases, other kernel functions can be implemented in the SVMs optimisation problem that will be able to draw more appropriate class boundaries.

Acknowledgments S.Z. acknowledges financial support from GlaxoSmithKline and the University of Bristol. C.G. and M.H.-B. also gratefully acknowledge Prof J. Timbrell (King's College London, Department of Pharmacy) for supply of urinary test samples for CE analyses.

References

1. McBurney A, Gibson T (1980) Clin Chim Acta 102:19–28
2. Christiensen E, Brock Jacobsen B, Gregersen N, Hjeds H, Pederson JB, Brandt NJ, Baekmark UB (1981) Clin Chim Acta 116:331–341
3. Frattini P, Santagostino G, Schinelli S, Cucchi ML, Corona GL (1983) J Pharm Meth 10:193–198
4. Clark EA, Fanguy JC, Henry CS (2001) J Pharm Biom Anal 25: 795–801
5. Adam T, Lochman P, Friedecky D (2002) J Chrom B 767:333–340
6. Barbas C, Garcia A, De Miguel L, Sima C (2002) J Chrom B 780:73–82
7. Vuorensola K, Siren H, Kostiainen R, Kotiaho T (2002) J Chrom A 979:179–189
8. Guillo C, Perrett D, Hanna-Brown M (2004) Chromatographia (in press)
9. Guillo C, Barlow D, Perrett D, Hanna-Brown M (2004) J Chromatogr A 1027:203–212
10. Holmes E, Nicholson JK, Nicholls AW, Lindon JC, Connor SC, Polley S, Connelly J (1998) Chem Intell Lab Syst 44:245–255
11. Hart BA, Vogels JTWE, Spijksma G, Brok HPM, Polman C, Van Der Greef J (2003) J Neurol Sci 212:21–30
12. Kim KR, Park HG, Paik MJ, Ryu HS, Oh KS, Myung SW, Liebich HM (1998) J Chrom B 712:11–22

13. Service KM, Brereton RG, Harris S (2001) Analyst 126:615–623
14. Xu G, Di Stefano C, Liebich HM, Zhang Y, Lu P (1999) J Chromatogr B 732:307–313
15. Yang J, Xu G, Kong H, Zheng Y, Pang T, Yang Q (2002) J Chromatogr B 780:27–33
16. La S, Cho JH, Kim JH, Kim KR (2003) Anal Chim Acta 486: 171–182
17. Belousov AI, Verzakov SA, Von Frese J (2002) Chem Intell Lab Syst 64:15–25
18. Belousov AI, Verzakov SA, Von Frese J (2002) J Chemometrics 16:482–489
19. Thissen U, Van Brakel R, De Weijer AP, Melssen WJ, Buydens LMC (2003) Chem Intell Lab Syst (in press)
20. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D (2000) Bioinformatics 16:906–914
21. Seong-Whan L, Verri A (eds) (2002) Pattern recognition with support vector machines: proceedings of the first international workshop, SVM 2002, Niagara Falls, Canada, 10 Aug 2002
22. Chromatographic Integration Matlab Toolbox. University of Amsterdam, The Netherlands. http://www-its.chem.uva.nl/research/pac/. Cited 8 Oct 2003
23. Shen H, Carter JF, Brereton RG, Eckers C (2003) Analyst 128:287–292
24. Brereton RG (2003) Chemometrics data analysis for the laboratory and chemical plant. Wiley, Chichester
25. Jackson JE (1991) A user's guide to principal components. Wiley, New York
26. Crosta GF, Zomer S, Pan YL, Holler S (2003) Opt Eng 42: 2689–2701
27. De Maesschalck R, Jouan-Rimbaud D, Massart DL (2000) Chem Intell Lab Syst 50:1-18.
28. Vapnik VN (2000) The nature of statistical learning theory, 2nd edn. Springer, Berlin Heidelberg New York
29. Scholkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimisation and beyond. MIT Press, London
30. Gunn SR (1997) Technical report. University of Southampton, England. http://www.ecs.soton.ac.uk/~srg/publications/pdf/SVM.pdf. Cited 8 Oct 2003
31. Burges CJC (1998) Data Min Knowl Disc 2:121–167
32. S. Raudys (2000) Neural Net 13:17–19.
33. Sánchez MS, Swierenga H, Sarabia LA, Derks E, Buydens L (1996) Chem Intell Lab Syst 33:101–119.
34. Denham RC (2000) The NetCDF toolbox for data conversion. Geological Survey, Woods Hole. http://woodshole.er.usgs.gov/staffpages/cdenham/public_html/. Cited 8 Oct 2003
35. Ma J, Ahalt S (2001) OSU SVM classifier Matlab Toolbox (Version 3.00). Ohio State University, Columbus. http://eewww.eng.ohio-state.edu/~maj/osu_svm/. Cited 8 Oct 2003