Journal of Computer-Aided Molecular Design **17:** 335–346, 2003.

ANVAS: Artificial Neural Variables Adaptation System for descriptor selection

Paolo Mazzatorta[1,2,*], Marjan Vračko[2] & Emilio Benfenati[1]
[1]*Istituto Mario Negri, via Eritrea 62, 20157 Milan, Italy;* [2]*National Institute of Chemistry, Hajdrihova 19, 1001 Ljubljana, P.O. Box 3430, Slovenia*

Key words: counterpropagation, genetic algorithm, neural networks, QSAR, variable selection

Summary

A new algorithm model-oriented for variable selection is presented in this study. It is based on the combination of genetic algorithms (GA) for hyperspace exploration, and counterpropagation artificial neural network (CP ANN) for deriving the fitness score. The proposed method performed very well on both well defined synthetic data sets and real academic data sets.

Introduction

> "I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection."
>
> Charles Darwin from *The Origin of Species.*

In computer chemistry, mathematical models such as quantitative structure-activity relationships (QSARs) are used to represent, explain, and most important predict or estimate a wide variety of physical, chemical, biological (including biomedical, toxicological, ecotoxicological), and technological properties [1–3]. Study on QSARs means that property relates to the corresponding structures. Consequently, provided a method to describe the structures, good predictive model can be constructed. Hence, it is beneficial to generate high numbers of descriptors containing topological, geometric, electronic and quantum-chemical features that maximize the amount of information in the input space. As the information content spreads over a very large number of potential molecular descriptors, it remains difficult to exploit. It is well known that increasing the number of variables will often cause a reduction in the generalization ability of the model (the curse of dimensionality) and some QSARs based descriptors do not add information but increase noise. Therefore it is necessary to select a subset of descriptors that retains most of the intrinsic information content. A number of mathematical and statistical methods [4–12], even if they proved to be efficient in several applications, rapidly exhibit limitations in large datasets [13, 14]. Therefore the use of these techniques in QSAR is not suitable.

This work proposes a general methodology for searching a solution space, result of a combination of genetic algorithms (GAs) for hyperspace exploration, and counterpropagation artificial neural network (CP ANN) for deriving the fitness score: ANVAS (Artificial Neural Variable Adaptation System). These two approaches are complementary and are presented as a preliminary test bed for their future application in QSAR studies. GA has already been considered to solve general optimization problems [13, 15–17] and CP ANN has been successfully applied in a number of computational chemistry problems [18–21].

Materials and methods

The GA is a stochastic global search method that mimics the natural biological evolution [15, 17, 22, 23]. GAs operate on a population of potential solutions applying the principle of survival of the fittest to produce approximations to a solution. At each generation, a new set of approximation is created by the process of selecting individuals according to their level

*Corresponding author. E-mail: mazzatorta@marionegri.it

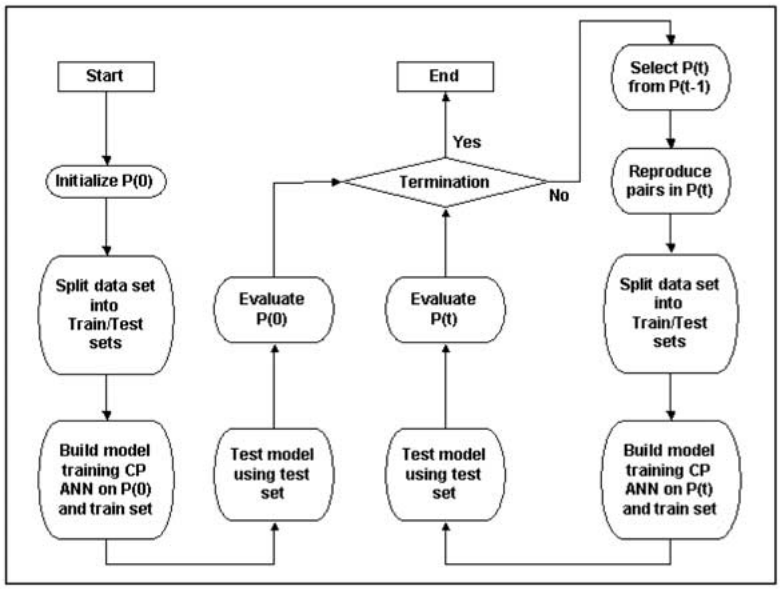


Figure 1. Flow-chart of ANVAS.

of fitness in the problem domain and breeding them together using operators from natural genetics. This process leads to the evolution of populations of individuals from a seed population, just as in natural adaptation. Individuals, or current approximations, are encoded as strings, *chromosomes*, composed over some alphabet(s), so that the *genotypes* (chromosome values) are uniquely mapped into the decision variable (*phenotypic*). In the context of variable selection the representation is the binary alphabet {0, 1}, where 0 defines the absence of the descriptor, and 1 defines its presence.

A flow-chart of the procedure is shown in Figure 1. The population at time t is represented by the time-dependent variable P, with initial population of random estimates being $P(0)$. Using this outline, the remainder of this section describes the major elements of the procedure.

Population representation and initialization

The first step is to create an initial population. This is achieved by generating the required number of individuals using a random number generator that uniformly distributes numbers in the desired range. The structure of each chromosome is composed of n variables, representing its dimension, each coded by a bit, 0 or 1; where 0 means the variable is not active, while 1 indicates its activity.

Objective function

The objective function is used to provide a measure of how individuals perform in the problem domain. In a minimization problem, the fittest individuals will have the lowest numerical value of the associated objective function. The index proposed to evaluate the performances of the individuals is the complement to 1 of the determination coefficient (R^2) for the test set as follows:

$$ObjF = 1 - R^2_{test} \tag{1}$$

where R^2 is a statistic used to determine how well a regression fits. For a general function $y = f(x)$, R^2 represents the fraction of variability in y that can be explained by the variability in x. In other words, R^2 explains how much of the variability in the y's can be explained by the fact that they are related to x. Then, R^2 can be expressed as:

$$R^2 = \frac{SS_{Total} - SS_{\mathrm{Re}\,s}}{SS_{Total}} = 1 - \frac{SS_{\mathrm{Re}\,s}}{SS_{Total}} \tag{2}$$

where SS_{Total} is the total sum of squares and SS_{Res} is the residual sum of squares of the data samples.

The data set is randomly split into training and test sets at each generation to prevent poor generalization related to unrepresentative sets. Active variables are present as inputs to a CP ANN. Hecht-Nielsen [24] and Dayhof [25] give a detailed description of CP ANN architecture and learning strategy. Architecture

of the CP ANN is shown in Figure 2. CP ANN consists of two layers of neurons, input or Kohonen layer and output layer. The input or Kohonen layer gets input variables related to considered objects. During the learning, the target values are given to output layer, which has the same topological arrangement of neurons as Kohonen layer. Learning in Kohonen layer is the same as in Kohonen networks. This means a vector of input variables is presented to all neurons. Program selects the neuron, which weights are closest to the input values. The chosen neuron is called the winning neuron. The position of objects is projected to the output layer. In the next step, the weights in output layer are corrected so that they fit the output values of corresponding objects. Then, the prediction for a new object runs in two steps. It will be first situated in Kohonen layer on the neuron with the most similar weights. Finally, the position is projected to the output layer, which provides the output value.

Selection and reproduction procedure

The chromosomes are evaluated using the objective function previously defined (1), and only the best individuals are retained for the reproduction procedure.

Selection is the process of determining the number of trials that a particular individual is chosen for reproduction and, thus, the number of offspring that an individual will produce. The algorithm supports two different mechanisms to select individuals: a stochastic universal sampling function, *SUS* [26] and the "roulette wheel" selection, *RWS* [15]. SUS is a single-phase sampling algorithm with minimum spread and zero bias. RWS is a stochastic sampling with replacement. A description of the procedure implemented is in the user's guide of the Genetic Algorithm Toolbox [27–29].

Crossover is the basic operator for reproducing new chromosomes in the GA producing new individuals that have some parts of both parent's genetic material. In this work, multi-point crossover [30] (Figure 3), m crossover positions, $k_i \in \{1, 2, \ldots, 1-1\}$, where k_i are the crossover points and l is the length of the chromosome, are chosen at random with no duplicates and sorted. Then, the bits between successive crossover points are exchanged between the two parents to produce two new offspring. The disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favoring the converge to highly fit individuals early in the search, thus making the search more robust.

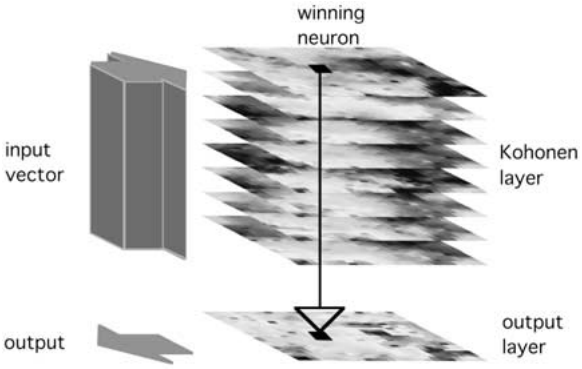


Figure 2. The architecture of CP ANN.

In natural evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. Finally, mutation (Figure 4) is randomly applied with low probability and modifies elements in the chromosomes. When chromosomes are represented by a binary string, as in this case, the mutation operator randomly switches some bit. The mutation serves to create random diversity in the population and it should prevent the algorithm converge towards a non-optimal solution.

Termination

Because the GA is a stochastic search method, it is difficult to formally specify converge criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found, the application of conventional termination criteria becomes problematic. A common practice is to terminate the GA after a specified number of generations and then check the nature of the best members of the population. If no acceptable results are found or the chromosome population is not comparable, the GA may be restarted or a fresh search initiated.

Software and computational details

The proprietary software was developed in MATLAB® (The MathWorks, Natick, MA) using elements from the Genetic Algorithm Toolbox [27, 28, 29] of A. Chipperfield *et al.* (Department of Automatic Control and System Engineering, University of Sheffield, Western Bank, Sheffield, UK). CP ANN is developed in Visual Fortran environment and then build as a MEX-file MATLAB callable. For this study the algorithm was implemented in a Intel® Pentium® III Mobile CPU 1200 MHz processor. The data pro-

Table 1. CP ANN parameters for the synthetic data set.

CP ANN parameter	Value
Dimension of the layer	10
Type of neighborhood corrections	Triangular
Maximal correction factor	0.50
Minimal correction factor	0.01
Epochs	1000

cessing time is largely dependent on the data set size defined by the number of object and variables examined and on the parameters used. For this study the processing time was lower than 15 min for the synthetic data sets, and lower than 210 min for the academic data sets.

Results

Synthetic data

The procedure was first tested on five well defined synthetic data sets, described below. Outliers from a QSAR are chemicals that do not fit the model or are poorly predicted by it [31]. Outliers are always present in experimental data sets and are useful in QSAR development because they assist in establishing the chemical domain of the model [32]. Therefore, in order to simulate real experimental data sets, some outliers were added.

For this analysis the parameters listed in Table 1 and Table 2 were used. These parameters allowed steady and repeatable runs. Figure 5a–e shows the five target functions.

Data set I

For this data set we generated one vector of 100 random numbers uniformly distributed between 0 and 1, x_1. We also generated nine vectors of normally distributed random numbers (zero mean, unit variance) in order to simulate Gaussian noise. The response was modeled as $\hat{y}_1 = f(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$. Variables x_2, $x_3, x_4, x_5, x_6, x_7, x_8, x_9$ and x_{10} represent only Gaussian noise. The true model is:

$$y_1 = x_1 \tag{3}$$

Five more random generated objects were added to this data set in order to simulate outliers. As example, an extract of the data set I is shown below:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y_1
1	0.950	0.226	0.934	0.916	0.829	0.013	0.276	0.907	0.217	0.970	0.950
2	0.231	0.580	0.264	0.602	0.166	0.310	0.368	0.759	0.652	0.715	0.231
...	...	...	...	...	...	...	...	...	...	...	...
100	0.988	0.393	0.740	0.784	0.170	0.069	0.196	0.930	0.233	0.065	0.988
101	0.583	0.592	0.432	0.986	0.540	0.853	0.787	0.310	0.008	0.375	0.253
102	0.423	0.120	0.634	0.473	0.623	0.180	0.619	0.269	0.397	0.374	0.585
103	0.516	0.038	0.803	0.903	0.686	0.032	0.016	0.536	0.650	0.484	0.524
104	0.334	0.459	0.084	0.451	0.677	0.734	0.891	0.163	0.085	0.969	0.163
105	0.433	0.870	0.945	0.805	0.877	0.537	0.762	0.211	0.769	0.342	0.486

This data set allowed us to evaluate the real ability of the GA in exploring 10-dimensional variable hyperspace in the presence of noisy information. The system was able to recognize the correlation between the actual variable x_1 and pointed out as irrelevant the remaining nine variables. The function is represented in Figure 5a.

Data set II

A set of independent variables was simulated by generating nine vectors of 100 random numbers uniformly distributed between 0 and 1. One more vector was added as Gaussian noise, x_{10}. The target function is a linear combination of nine variables:

$$y_2 = x_1 + x_2 + x_3 + x_4 + x_5 - x_6 - x_7 - x_8 - x_9 \tag{4}$$

Five outliers were added to this data set. We displayed in Figure 1b the function.

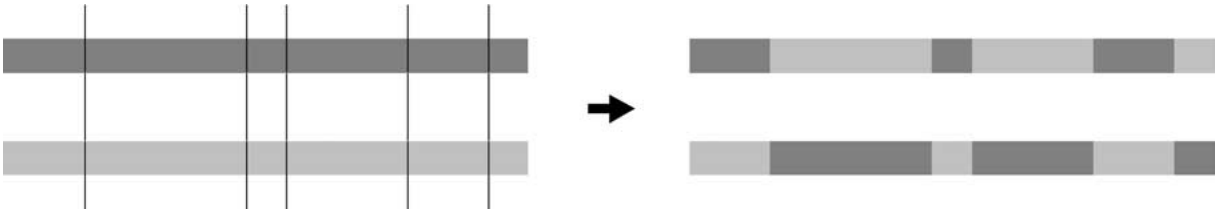


Figure 3. Multi-point Crossover (m = 5).

Table 2. GA parameters for the data sets.

GA parameter	Synthetic data	Academic data set I	Academic data II
Chromosome number	5	10	12
Chromosome size	10	34	50
Number of generation	100	250	300
Mechanism of selection	SUS	SUS	SUS
Rate of individuals to be selected	1	1	1
Probability of crossover	0.7	0.7	0.7
Crossover point number	2	2	2
Probability of mutation	0.02	0.02	0.02

In this case the presence of numerous relevant variables makes the exploration of the hyperspace a difficult task. For this particular problem a backward elimination could probably faster achieve the aim because only one of the variables should be discarded but it remains the problem to stop the procedure. GA correctly interpreted 90% of the variables, in fact all the variables but x_4 were correctly identified.

Data set III

For this data set, we used the same procedure as before to simulate five variables, with uniformly distributed random numbers between 0 and 1, and five variables representing Gaussian noise. In this case the complexity of the response y is slightly increased (Figure 5c):

$$y_3 = x_1 + 2x_2 + 4x_3 - x_4 - 3x_5 \tag{5}$$

Again, five outliers were added. The target function is still linear, but the importance and influence of each variable in the response is different. The system correctly distinguished nine variables and misclassified only one variable (x_1).

Data set IV

For this data set, we again generated 2 vectors of 100 random numbers that were used as input variables and eight vectors of 100 random numbers that were used as Gaussian noise. The true model is:

$$y_4 = x_1^2 - \log_{10}(x_1) + 3x_2 \tag{6}$$

In this case the efficiency of the objective function, i.e. CP ANN, was analyzed. The dependence of the response y_4 to the variables x_1 and x_2 is very complex and highly non-linear. The system correctly selected all the actual variables.

Data set V

This data set was generated as the previous. We displayed in Figure 5e the true model:

$$y_5 = x_1 + x_2^2 + \log_{10}(x_3) + \frac{1}{(x_4 + 1)} - 2x_5 \tag{7}$$

Five more objects were randomly generated and added to the data set as outliers. In the final test five variables (x_1, x_2, x_3, x_4, x_5) are involved in the definition of the response y_5 and five variables represent only Gaussian noise. The target function presents both linear and non-linear correlation. This makes this analysis a strong and reliable test for both hyperspace exploration and chromosome selection. In this case it correctly interpreted 80% of the variables, i.e. x_2 and x_4 were not selected as relevant variables.

Results of the analysis for all target functions are summarized in Table 5.

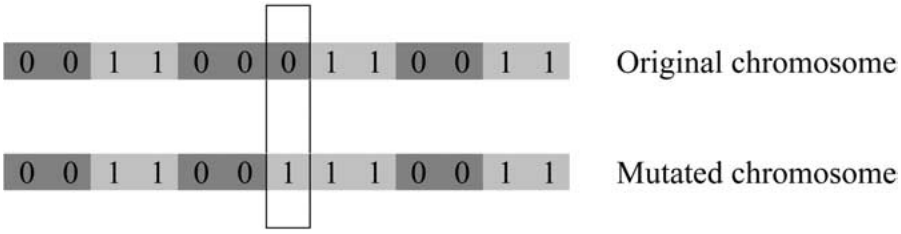


Figure 4. Binary mutation.

Academic data

We also evaluated the method on real academic data sets, which allowed us to extent the analysis and the conclusions to actual data sets for QSAR studies.

Academic data set I

This data set contains 92 chemical compounds and the corresponding chronic dose rate that would give half the animals tumors (TD50), data are referred to mouse. The original data set was collected by Gold and colleagues [33] and contains more than 1200 chemicals. We limited the compounds to be evaluated to those containing an aromatic ring and a nitrogen linked to the aromatic ring. Chemicals with a non-carcinogenic effect on mouse were discarded. The data set was supplemented by 34 molecular descriptors as described in previous work [34]. The output (TD50) was transformed as follow:

$$y_6 = \log\left(MW \cdot \frac{1000}{TD50}\right) \tag{8}$$

where MW is the molecular weight, and then normalized between 0 and 1 using a range scaling procedure, in order to have a more continuos output space and refer to the moles of the chemical rather than the weight [19].

From the 34 descriptors calculated, 15 were selected using the parameters listed in Tables 1 and 2: HOMO, LUMO, heat of formation, dipole moment, Randic Index, Wiener Index, Kier & Hall connectivity index order 0, Kier & Hall connectivity index order 4, log D at pH 2, log D at pH 7.4, first principal moment of inertia, third principal moment of inertia, Kappa simple index 1, Kappa alpha index 2, and electrotopological sum.

Using the parameters listed in Table 1 and 46 molecules random selected as training set, two models were developed exploiting once all 34 descriptors and then the 15 selected descriptors. The models were finally test on the remaining 46 molecules. The determination coefficient (R^2) for the test set exhibit a considerable improvement using the selected descriptors (see Figure 7 and Table 4).

Academic data set II

Debnath *et al.* [35] collected the mutagenic activities of a set of aromatic and heteroaromatic amines in *S. typhimurium* TA98+S9 microsomal preparation. Basak and Mills [36] supplemented the data set by molecular descriptors including topostructural, topochemical, geometrical and quantum chemical indices. This data set contains 95 chemical compounds with their mutagenic activities and 50 variables.

Parameters listed in Tables 1 and Table 2 were used to select a subset of relevant variables through ANVAS and, as result, 26 descriptors were selected: ${}^{3D}W$, ${}^{3D}W_H$, kp0, kp1, kp2, kp3, LUMO, ${}^4\chi^v$, ${}^5\chi^v_{Ch}$, ${}^{10}\chi^v_{Ch}$, ${}^5\chi^b_{CP}$, SIC_4, I_{ORB}, ASZ_5, SHCsats, Sumdel$I$, SHsOH, NumHBd, Gmin, SddsN, NHBint9, SssNH, I^W_D, ${}^4\chi$, ${}^4\chi_{PC}$, DSN_1.

The model developed exploiting the subset obtained showed a relevant increasing in the prediction ability of 47 random selected object not used in training compared to the rough model including all the variables (Figure 7 and Table 4).

Discussion

Synthetic data

Figure 5 shows the transformed variables X_i in the target functions I (a), II (b), III (c), IV (d) and V (e). Transformed variables are the variables derived from the original ones considering the true model intrinsically linear. For instance, in the target function III the transformed variables for x_1, x_2, x_3, x_4, and x_5 are:

$$\begin{aligned} X_1 &= x_1 \quad X_2 = 2x_2 \quad X_3 = 4x_3 \quad X_4 = -x_4 \\ X_5 &= -3x_5 \end{aligned} \tag{9}$$

Figure 5f displays some common statistical measures for the transformed variables, as range, interquart-

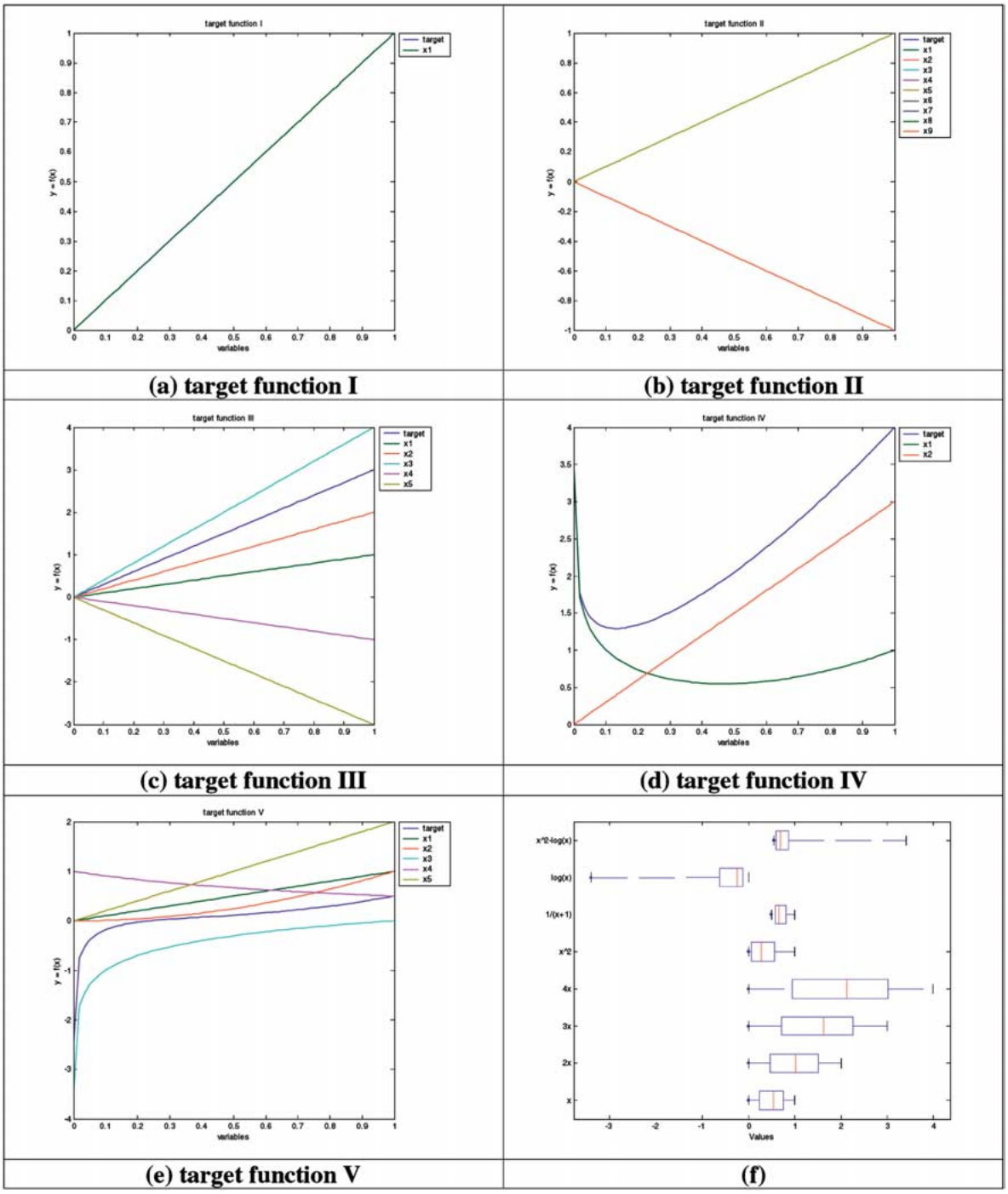
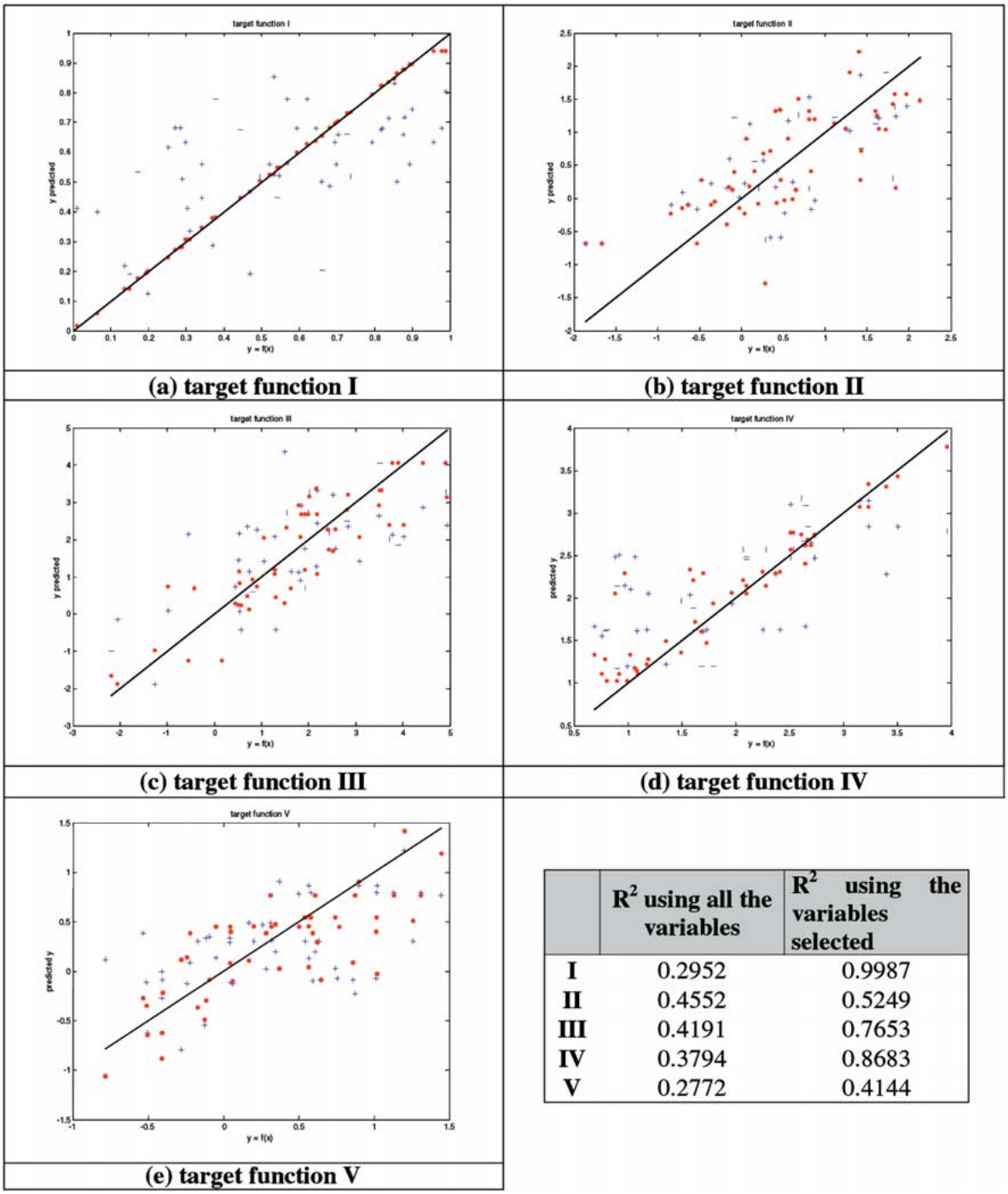


Figure 5. (a) transformed variables in target function I. (b) transformed variables in target function II. (c) transformed variables in target function III. (d) transformed variables in target function IV. (e) transformed variables in target function V. (f) statistical information of the transformed variables: the box is the interquartile range (the difference between the 75^{th} and 25^{th} percentile of the data); the line in the middle of the box is the sample median; the lines extending the box show the extent of the rest of the sample.



	R^2 using all the variables	R^2 using the variables selected
I	0.2952	0.9987
II	0.4552	0.5249
III	0.4191	0.7653
IV	0.3794	0.8683
V	0.2772	0.4144

Figure 6. Prediction of the test set by a model developed using all the variables (+) and only the variables selected (.) for the target function I (a), the target function II (b), the target function III (c), the target function IV (d) and the target function V (e). The table presents the determination coefficient (R^2) of the test sets for the models developed.

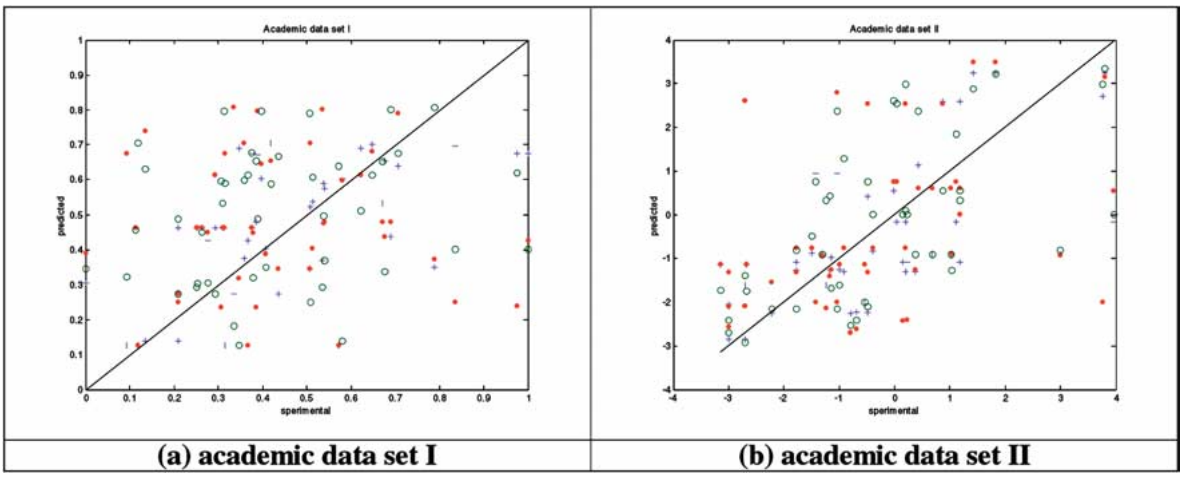


Figure 7. Prediction of the test set by a model developed using all variables (.), variables selected using ANVAS (+), and variables selected by other methods (o) for the academic data set I (a) and the academic data set II (b).

Table 3. Correlation coefficients.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_1	0.9758	0.0802	0.0040	0.1106	−0.1210	−0.1790	−0.0129	−0.0778	−0.1786	−0.1665
y_2	0.4329	0.3373	0.4044	0.4514	0.1684	−0.4643	−0.4359	−0.2237	−0.3973	−0.0816
y_3	0.2604	0.3653	0.6137	−0.0377	−0.5240	−0.1055	−0.1072	−0.1067	−0.1604	0.1262
y_4	−0.0452	0.8589	−0.1104	0.0273	−0.0287	−0.1359	−0.0669	0.0869	−0.0706	0.0200
y_5	−0.1610	−0.0522	−0.1057	−0.3536	−0.0595	0.1135	0.1326	0.0526	0.0461	−0.0840

Table 4. Determination coefficient (R^2) of the test sets for the models developed.

	R^2 using all variables	R^2 using variables selected by ANVAS	R^2 using variables selected by other method
I	0.0007	0.2967	0.0382
II	0.4040	0.6371	0.6030

ile range (the difference between the 75th and 25th percentile of the data) and median.

Analyzing these plots (Figure 5) it is possible to comment on the errors in the selection of the variables.

In the target function V we notice (Figure 5e and Figure 5f) that X_2 and X_4 have the lowest variability in respect to the others transformed variables. In this case the role of these input variables, i.e. x_2 and x_4, in the target function is not as important as the other variables and the procedure is not able to detect their correlation with the response y_5. Moreover, if we look carefully to Figure 5e we notice that the transformed variables X_2 and X_4 are the inverse of the other in the observed interval, making their contribute similar to a constant (std $= 0.1821$, mean $= 1.0363$).

In the case of target function II (Figure 5b) all the actual variables have the same importance in the response y_2; the difficulty in selecting x_4 is probably related to its particular distribution. In fact if two different variables are 'similar' or correlated, i.e. $x_{a,i} \cong x_{b,i}$ for $\forall\ i$ where x_a and x_b are vectors/variables, the procedure may recognize just one of them. For instance, if x_a and x_b are 'similar' vectors, the response $y = x_a + x_b$ can be easily mistaken for $y = 2x_a$ or $y = 2x_b$.

For the target function III similar considerations apply. X_1 and X_4 have the lowest variability and the contribution of x_1 to the response y_3 is probably merged with the linear contribution of the other variables.

Table 5. Overview of the results on the synthetic data set.

	Description	Actual variables	Noisy variables	Variables selected by GA	GA NER%	Variables selected by MLRA	MLRA NER%
I	Linear correlation	x_1	x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}	x_1	100	x_1	100
II	Linear correlation	x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9	x_{10}	x_1, x_2, x_3, x_5, x_6, x_7, x_8, x_9	90	x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9	90
III	Linear correlation	x_1, x_2, x_3, x_4, x_5	x_6, x_7, x_8, x_9, x_{10}	x_2, x_3, x_4, x_5	90	x_1, x_2, x_3, x_5	90
IV	Non-linear correlation	x_1, x_2	x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}	x_1, x_2	100	x_2	90
V	Non-linear correlation	x_1, x_2, x_3, x_4, x_5	x_6, x_7, x_8, x_9, x_{10}	x_1, x_3, x_5	80	x_4	60

Models were developed using CP ANN trained with and without variable selection and in all the cases the selection of the variables showed a significant improvement of the performances in modeling (Figure 6).

A full exploration of the variables hyperspace would involve the generation of t models,

$$t = \sum_k \left(\frac{p!}{k!\,(p-k)!} \right) \tag{10}$$

where p is the total number of variables and k is the maximum dimension of the model, i.e. maximum number of variables involved in the model. In this case, $p = 10$ and $k = 10$, it would require the generation of 1023 models. But the exploitation of GA by ANVAS allowed the generation of only 500 models (Table 2). The performances of ANVAS are then compared to the results of Multiple Linear Regression Analysis (MLRA). The performances of a linear approach can be easily foreseen by simply *eye-balling* the correlation matrix for each model data (Table 3).

For this analysis it is relevant the arbitrary choice of the cut-off value of the correlation coefficient for determining where a variable is or is not relevant. Knowing the true model and observing Table 3 we can notice that none relevant variable has an absolute correlation coefficient below 0.2. Setting the cut-off value to 0.2 means to use the whole information we have about the data sets and to have the best possible selection. But, even in this luckiest case, this analysis misclassify: x_5 in data set II; x_4 in data set III; x_1 in data set IV; x_1, x_2, x_3, x_5 in data set V (Table 5).

Academic data

In this case a comparison with respect to the true model is not possible. Therefore the method was evaluated simply on the ability in prediction of the model developed exploiting the obtained selection of variables, compared with similar models developed using all the variables, and a variable selection obtained through traditional methods. It is important to notice the superiority of the method proposed as regards to traditional methods.

For the first data set, principal component analysis (PCA) was used to select a smaller set of descriptors [34] and then a similar CP ANN was trained using the same parameters and the same molecules for the training set. The prediction ability of the remaining test set shows an improvement in the performances ($R^2 = 0.0382$) but was not comparable with the selection obtained by ANVAS ($R^2 = 0.2968$) (Figure 7a and Table 4).

In the second case the selection was obtained using regression methodologies. Within the numerous selection performed by Basak and Mills [36] we choose the one that gave the best results in their models. Again similar CP ANN were trained using the same parameters and the same object and then test on the remaining molecules of the data set. Once again the selection obtained by ANVAS showed the best results (Figure 7a and Table 4).

Conclusions

Table 5 summarized the results of the analysis for the synthetic data sets. The Non Error Rate percentage (NER%) is computed considering the number of variables correctly interpreted by ANVAS as relevant or not, out of the whole pool. It is important to underling that identifying irrelevant variable as irrelevant is also a correct identification.

A new algorithm suitable to select relevant variables in a problem domain has been elaborated. This algorithm, derived from the GA concepts for hyperspace exploration was combined with a CP ANN to derive a specific score index, evaluating the quality of the selection. At each generation of the GA different training and test sets were randomly generated. The fitness score of each chromosome was derived by the determination coefficient of the test set. This strategy can slow down the speed of convergence of the algorithm but assures the selection of descriptors subsets that lead to general and suitable models, by preventing over-fitting.

The selection power of the proposed ANVAS was tested on synthetic data sets. 100 objects described by 10 variables were generated and correlated by five different target function to the response *y*, five objects were added and used as outliers. The method allowed deriving relevant subsets of descriptors in all cases (Table 5). Two models were developed for each target functions (Figure 6); the first one was trained without descriptor selection and the second one was developed through exploitation of the descriptors selected by ANVAS. The examination of the results confirmed immediately that the GA selection procedure allowed a notable improvement of the ability in prediction of the models. We want to point out that ANVAS was able to recognize linear and non-linear relations between variables and response in data sets holding both poor and abundant information.

The method was also tested on real academic data sets. The performances of the subset of variables obtained with ANVAS was compared with the variable selection obtained in previous work on the same data sets. Rough models were developed under the same conditions, but the variables exploited. In both the cases the selection resulting from the method proposed gave the best results.

Finally, it is known it is time consuming for variable selection by using the genetic algorithm approach than that by using other methods. But when the descriptor pool is large, as in the case of QSAR studies, the advantages by using genetic algorithm will be distinctive.

Acknowledgements

This work is partially funded by the EU under contract HPRN-CT-1999-00015.

References

1. Mazzatorta, P., Vračko, M., Jezierska, A. and Benfenati, E J., Chem. Inf. Comput. Sci., 43 (2003) 485.
2. Schultz, T.W., Cronin, M.T.D., Walker J.D. and Aptula, A.O., J. Mol. Struct.: THEOCHEM, 622 (2003) 1.
3. Sabljic, A., Chemosphere, 43 (2001) 363.
4. Baroni, M., Costantino, G., Cruciani, G., Riganelli, D., Valigi, R. and Clementi, S., Quant. Struct.-Act. Relat., 12 (1993) 9.
5. Burden, F.R., Ford, M.G., Whitley, D.C. and Winkler, D.A., J. Chem. Inf. Comput. Sci., 40 (2000) 1423.
6. Tetko, I.V., Villa, A.E. and Livingstone, D.J., J. Chem. Inf. Comput. Sci., 36 (1996) 794.
7. Castellano, G. and Fanelli, A.M., Neurocomputing, 31 (2000) 1.
8. Chatterjee, S. and Price, B., Regression Analysis by Example, Wiley: New York, 1977.
9. Despagne, F. and Massart, D.-L., Chem. Intell. Lab. Sys., 40 (1998) 145.
10. Höskuldsson, A., Chem. Intell. Lab. Syst., 55 (2001) 23.
11. Lindgren, F., Geladi, P. and Wold, S., J. Chemometr., 8 (1994) 377.
12. Sutter, J.M. and Kalivas, J.H., Microchem. J., 47 (1993) 60.
13. Ros, F., Pintore, M. and Chrétien, J.R., Chemom. Intell. Lab. Syst., 63 (2002) 15.
14. Xu, L. and Zhang, W.-J., Anal. Chim. Acta, 446 (2001) 477.
15. Goldberg, D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley: New York, 1989.
16. Hasegawa, K., Miyashita, Y. and Funatsu, K., J. Chem. Inf. Comput. Sci., 37 (1997) 306.
17. Hibbert, D.B., Chemom. Intell. Lab. Syst., 20 (1993) 35.
18. Mazzatorta, P., Vračko, M. and Jezierska, A., Proceedings of SKD 2002, Slovenski Kemijski Dnevi 2002, September 26–27 2002: Maribor, Slovene, 2002, pp. 329–332.
19. Vracko, M., J. Chem. Inf. Comput. Sci., 37 (1997) 1037.
20. Vracko, M., Novič, M. and Zupan, J., Anal. Chim. Acta, 384 (1999) 319.
21. Zupan, J., Novič, M., Li, X. and Gasteiger, J., Anal. Chim. Acta, 292 (1994) 219.
22. Holland, J., Adaptation in Natural and Artificial Systems, The University of Michigan Press: Ann Arbor, 1975.
23. Kinnear, K.E., Advances in Genetic Programming, MIT Press: Cambridge, MA, 1994.
24. Hecht-Neilson, R., Appl. Optics, 26 (1987) 4979.
25. Dayhof, J., Neural Network Architectures, An Introduction, Van Nostrand Reinhold: New York (1990) 192.
26. Baker, J.E., Proc ICGA 2 (1987) 14.
27. Chipperfield, A.J. and Fleming, P.J., IEE Colloqium on Applied Control Techniques Using MATLAB, 14 (1995).
28. Chipperfield, A.J., Fleming, P.J. and Fonseca, C.M., Proc. Adaptive Computing in Engineering Design and Control, Plymouth Engineering Design Center, 21–22 September, pp. 128–133, 1994.
29. http://www.shef.ac.uk/uni/projects/gaipp/ga-toolbox/.

30. Booker, L., Genetic Algorithms and Simulated, Annealing, L. Davis (Ed.), Morgan Kaufmann Publishers: 1987, pp. 61–73.
31. Egan, W.J. and Morgan, S.L., Anal. Chem., 70 (1998) 2372.
32. Schultz, T.W. and Cronin, M.T.D., Environ. Toxicol. Chem., 22 (2002) 599.
33. Gold, L.S., Slone, T.H., Manley, N.B., Backman Garfinkel, G., Hudes, E.S., Rohrbach, L. and Ames, B.N., Environ. Health Perspect., 96 (1991) 11.
34. Gini, G., Lorenzini, M., Benfenati, E., Grasso, P. and Bruschi, M., J. Chem. Inf. Comput. Sci., 39 (1999) 1076.
35. Debnath, A.K., Debnath, G., Shusterman, A.J. and Hansch, J., Environ. Mol. Mutagen., 19 (1992) 37.
36. Basak, S.C. and Mills, D., SAR QSAR in Environ. Res., 12 (2001) 481.