3 Genetic Algorithms in Feature Selection

R. LEARDI
Istituto di Analisi e Tecnologie Farmaceutiche ed Alimentari, Università di Genova, via Brigata Salerno (ponte), I-16147 Genova, Italy

Starting from the original algorithm, several changes made genetic algorithms a powerful tool in feature selection. A full validated version has also been implemented, to check that the selected subset is really predictive and not due only to chance correlations (very easy when working with a large variables/objects ratio).

Hybrid algorithms are conceptually very simple: after a certain number of generations of genetic algorithms, the best experimental condition so far found undergoes a 'classical' method of optimization (in the case of feature selection, stepwise selection); the results thus obtained can enter the population, and then a new genetic algorithm is started with the updated population. This approach allows further improvement of the performance of the genetic algorithm.

The application of genetic algorithms to two quantitative structure–activity relationship data sets will be presented, and the results will be compared with those described in literature.

KEYWORDS: *genetic algorithms; feature selection; regression; full validation.*

ABOUT FEATURE SELECTION

One of the main problems when elaborating large data sets is the detection of the relevant variables (i.e. the variables holding information) and the elimination of the noise. Roughly speaking, each data set can be situated somewhere in between the following two extremes:

(a) data sets in which each variable is obtained by a different measurement (e.g. clinical analyses);
(b) data sets in which all the variables are obtained by a single measurement (e.g. spectral data).

In, *Genetic Algorithms in Molecular Modeling* (J. Devillers, Ed.)
Academic Press, London, 1996, pp. 67–86.
ISBN 0-12-213810-4

It can be easily understood that while for data sets of type (b) the only goal of feature selection is the elimination of noise, together with the simplification of the mathematical model, for data sets of type (a) it is very worthwhile to try to reduce as much as possible the number of variables involved, since this also means shorter analysis time and lower costs.

The procedure of feature selection, apparently so simple, is indeed very dangerous and needs a very careful validation, to avoid the risk of overestimating the predictive ability of the selected model; in such cases, when using it on new data, one can be deceived, discovering that it has no predictive ability at all (Lanteri, 1992).

This is mainly due to the random correlations: if you try to describe 10 hypothetical objects with 100 random X variables and a random response, you will surely have some X variables perfectly modelling your response. This risk is of course higher when the ratio variables–objects is very high: this is the typical case of quantitative structure–activity relationships (QSAR), when only a few molecules are described by several tens of molecular descriptors.

APPLICATION OF GENETIC ALGORITHMS TO FEATURE SELECTION

Genetic algorithms (GAs) can be very easily applied to feature selection; in this paper it will be shown that very good results are obtained with a 'tailor-made' configuration, in which the classical GA has been slightly modified taking into account several peculiarities of this particular problem (Leardi *et al.*, 1992; Leardi, 1994).

In a GA applied to feature selection, the structure of the chromosomes is very simple, since each X variable is a gene coded by a single bit (0 = absent, 1 = present) (Lucasius and Kateman, 1991; Leardi *et al.*, 1992; Lucasius *et al.*, 1994).

The response to be maximized is the percentage of predicted variance (in the case of a regression problem) or the percentage of correct predictions (in the case of a classification problem). As an example, suppose we have 10 variables, and we want to find the best subset of variables to be used with partial least squares (PLS). If the chromosome to be evaluated is 0010011001, then the response will be the variance predicted by the PLS model computed by taking into account variables 3, 6, 7, and 10.

The usual way to evaluate the predictive ability of a model is cross-validation with a predefined number of deletion groups (e.g. 5). When performing a process of feature selection, one has to be aware that such a 'prediction' is only partially validated, since the objects on which the selection is performed are also those on which the prediction ability is computed, and this does not get rid of the problem of random correlation (we will see later on how a full-validated GA has to be). The reliability of this approach is therefore highly dependent on the objects–variables ratio: the results obtained when this is very high are really very near to full-validation; on the other side, when only a few objects are present, the results are almost meaningless.

CLASSICAL METHODS OF FEATURE SELECTION VS GENETIC ALGORITHMS

The only way to be sure of selecting the best subset of variables (of course, without taking into account the problem of random correlations) would be to compute all the possible models. The only limitation to this approach is given by the fact that, with n variables, $2^n - 1$ combinations are possible; as a consequence, this method becomes not applicable when the variables are more than just a few. To give an idea, with 30 variables (a rather small data set) more than 1 billion combinations are possible (computing one model per second, it would take 34 years!).

The most commonly used technique is the stepwise approach, which can run forward or backward.

In the forward version, the variables are selected and added to the model one at a time, until no improvement in the results is obtained. In the backward version, the first model to be computed is the one with all the variables; the variables are then selected and removed one at a time. A mix of these two approaches is also used, in which each addition of a variable by the forward selection is followed by a backward elimination.

The stepwise techniques have two main disadvantages:

- each choice heavily affects the following choices (e.g., in the forward version, once one of the variables has been selected, all the models that don't contain it will never be taken into account);
- the final result is expressed by a single combination, and then no choice is given to the user.

GA, in their favour, always allow the exploration of the whole experimental space: due to the occurrence of the mutations, each possible combination can occur at any moment.

The result obtained by GA is a whole population of solutions: the user can then choose the one he prefers, taking into account at the same time the response and the variables used (the user could, for example, be interested in limiting as much as possible the use of certain variables or in taking into account a model that can also have good theoretical explanations).

From this it is also evident that a relevant feature of GA is the ability to detect several local maxima, and then to give a good idea about the presence of several regions of interest.

Usually the performance of a GA is evaluated only in terms of how often and in how much time it leads to the detection of the global optimum. Apart

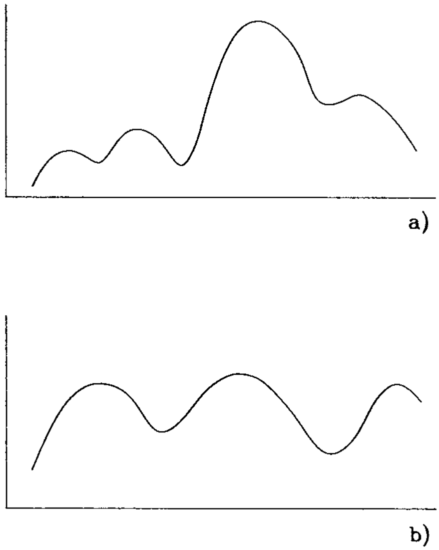


Figure 1 The problem of localizing the global maximum: in (a) the global maximum is by far better than the other local maxima, while in (b) the three maxima are equivalent, and the difference among them is only due to the noise.

from the fact that one should know where it is (people cannot afford to wait 34 years to check which is the best combination in a 30 variables data set), this can be sensible in the case shown in Figure 1a, in which the global maximum is much better than the other local maxima.

In the case of Figure 1b, on the contrary, the three local maxima are totally equivalent, and then it would be absolutely nonsense to look only for the global one. The best algorithm would be the one giving the information that three equivalent optima are present: the user will then be free to choose the most convenient one.

Another very important consideration is linked to the fact that we are always working with experimental data, affected by an experimental error; as a consequence, going back to Figure 1b, another set of measurements would produce a slightly different result, according to which the order of the three local maxima could be changed.

CONFIGURATION OF A GENETIC ALGORITHM FOR FEATURE SELECTION

It is clear that, in comparison with the configuration of GA usually employed for the resolution of other kinds of problems, the configuration of a GA devoted to feature selection must take into account some peculiarities. Some of them are the following:

- The time required by the evaluation of the response is much higher than the time required by GA itself: indeed, to evaluate the response means to perform a full multivariate analysis (e.g. multiple linear regression (MLR), partial least squares or linear discriminant analysis), with different deletion groups to obtain a predicted value (though only in partial validation). In a reasonable amount of time, working on a PC (all the computing times will be referred to a 486 PC with mathematical coprocessor, 33 MHz), only a few thousands of chromosomes can be evaluated, compared with the classical applications in which tens of thousands of generations are produced.

  Furthermore, since we are dealing with experimental data, one has to be aware of the possibility of random correlation and random predictions: the effect of both is higher the more combinations are tested (and then the more chromosomes are evaluated).

  An algorithm allowing a fast improvement of the response function is then required. This is obtained by reducing the population size to about 30 chromosomes (usual sizes range between 50 and 500 elements) and by introducing the highest possible degree of elitism (population k + 1 is formed only by the two offspring of a pair of parents; they can immediately replace the two lowest ranked elements of population k, and they can coexist with their parents).

  There are several advantages deriving from this configuration: once a good chromosome has been found, it immediately enters the population, and then can be immediately picked up as a parent, without having to wait for the evaluation of the other chromosomes of its same generation; a low size population has a lower exploration ability and a higher exploitation ability, since the best chromosomes are more frequently selected as parents; the high elitism allows all the best chromosomes found to be kept.

  Another modification to the original algorithm is the creation of a library of chromosomes in which all the already evaluated chromosomes are listed. A chromosome will be evaluated only if it never had a 'twin', since a library search is by far less time consuming than the evaluation of a response.

- The 'quality' of a subset of variables is related both to the value of the response and to the selected variables (how many and which ones). In many cases it can be important to have solutions containing as few variables as possible, since it will lead to an easier mathematical model and sometimes to lower costs.

  It is therefore interesting to evaluate how the response changes as a function of the number of selected variables. To do that, the following rule has been added: the best chromosome using the same number of variables is protected, regardless of its position in the ordered population, unless a chromosome with a lower number of variables gives a better response. Applying this rule, a protected chromosome can be 'killed' only by another chromosome giving a better response by using the same (or a lower) number of variables.

At the end of the run, the user can see very easily the evolution of the response as a function of the number of selected variables and then decide what is the best compromise between the number of variables and the response function.

- Usually, the best combinations select only a small percentage of the variables, and then in a chromosome the number of 0s is much higher than the number of 1s. This is rather peculiar of this kind of optimization, since in numerical optimization the number of 0s and 1s is on average the same. The usual initialization of the population, in which each bit is given a value of 0 or 1 at random with the same probability, would be almost inapplicable in this case, since the initial population would be formed by chromosomes corresponding to subsets containing as an average half of the variables of the data set.

  Apart from the fact that in some cases these chromosomes could not be evaluated (e.g., in MLR the number of variables must be lower than the number of objects), such a situation would lead to two main disadvantages: the computation of the response for a model containing a higher number of variables requires a much higher time, and then a much lower number of chromosomes can be evaluated in the same time; since the presence of a 'bad' variable amongst several 'good' variables would be almost unnoticed, that chromosome would have a very high response and then that variable would stay undisturbed in it.

  The solution applied to this problem is very simple: at the stage of the creation of the initial population, the probability of having a '1' is much lower than that of having a '0'. As a consequence, the original population is formed by chromosomes corresponding to subsets of only a few variables each (as a guideline, for each gene the probability of being a '1' is set to 5/number of variables; on average, in each chromosome 5 variables will be selected). This means that during the first stages a much higher number of chromosomes is evaluated and that 'bad' variables can be more easily discarded, since one of them can be enough to significantly worsen the response of a chromosome in which only a few 'good' variables are selected. Within each run, the combination of these small, highly informative 'building blocks' will lead to a gradual increase in the number of selected variables, until when it will stabilize around the 'optimal' number.
- One of the advantages of GA is that at the end of a run a population of possible solutions is obtained. As an example, a final population

Table I *Final population of a GA run on the ANTIM data set.*

#	Resp.	Selected variables
1	95.80	2, 7, 13, 17, 20, 50, 51, 52
2	95.01	2, 7, 13, 15, 20, 50, 51, 52
3	94.48	2, 7, 13, 20, 50, 51, 52
4	94.36	2, 9, 13, 20, 50, 51, 52
5	94.13	1, 2, 13, 17, 20, 46, 50, 51
6	93.93	2, 9, 20, 50, 51, 52
7	93.87	1, 7, 13, 20, 47, 50, 51
8	93.83	2, 20, 38, 50, 51
9	93.66	1, 2, 13, 17, 20, 50, 51
10	93.58	2, 17, 20, 38, 50, 51
11	93.24	2, 7, 13, 16, 20, 46, 50, 51, 52
12	93.21	1, 7, 20, 39, 50, 51
13	93.09	2, 7, 14, 17, 20, 50, 51, 52
14	92.92	2, 7, 13, 16, 20, 50, 51, 52
15	92.88	2, 7, 13, 20, 38, 50, 51, 52
16	92.76	2, 9, 13, 17, 20, 50, 51
17	92.73	1, 2, 13, 20, 50, 51, 52
18	92.57	1, 7, 20, 47, 50, 51, 52
19	92.57	1, 7, 13, 20, 39, 50, 51
20	92.55	2, 7, 9, 20, 50, 51, 52
21	92.27	1, 7, 20, 38, 39, 50, 51
22	92.24	2, 13, 20, 50, 51, 52
23	92.18	2, 20, 44, 50, 51, 52
24	91.89	1, 7, 13, 16, 20, 50, 51, 52
25	91.84	2, 7, 13, 20, 46, 50, 51, 52
26	91.78	1, 7, 13, 20, 47, 50, 51, 52
27	89.07	2, 20, 50, 51
28	79.64	50, 51, 52
29	40.74	50, 52
30	30.21	52

obtained by a GA applied to the ANTIM data set (see later in the text) is the one shown in Table I. If we look at chromosome 3, we can see that the variables selected by it are a subset of the variables selected by chromosomes 11, 14, 15, and 25. Since chromosome 3 gives the best result, chromosomes 11, 14, 15, and 25 are useless (they give a less good result by using the same variables as chromosome 3, plus some more). The same thing happens also with other chromosomes, so that we can say that the only 'useful' chromosomes are the ones reported in Table II. Only 21 chromosomes out of 30 bring forward valuable information. In such a case, the presence of 'useless' chromosomes produces bad effects for two main reasons: the information obtained by the final population is reduced since not all the chromosomes are relevant; the 'useless' chromosomes are very similar to some other chromosomes,

Table II *'Useful' chromosomes from the population of Table I.*

#	Resp.	Selected variables
1	95.80	2, 7, 13, 17, 20, 50, 51, 52
2	95.01	2, 7, 13, 15, 20, 50, 51, 52
3	94.48	2, 7, 13, 20, 50, 51, 52
4	94.36	2, 9, 13, 20, 50, 51, 52
5	94.13	1, 2, 13, 17, 20, 46, 50, 51
6	93.93	2, 9, 20, 50, 51, 52
7	93.87	1, 7, 13, 20, 47, 50, 51
8	93.83	2, 20, 38, 50, 51
9	93.66	1, 2, 13, 17, 20, 50, 51
12	93.21	1, 7, 20, 39, 50, 51
13	93.09	2, 7, 14, 17, 20, 50, 51, 52
16	92.76	2, 9, 13, 17, 20, 50, 51
17	92.73	1, 2, 13, 20, 50, 51, 52
18	92.57	1, 7, 20, 47, 50, 51, 52
22	92.24	2, 13, 20, 50, 51, 52
23	92.18	2, 20, 44, 50, 51, 52
24	91.89	1, 7, 13, 16, 20, 50, 51, 52
27	89.07	2, 20, 50, 51
28	79.64	50, 51, 52
29	40.74	50, 52
30	30.21	52

and so the exploration potential is very much reduced; as a consequence, the risk of being stuck on a local maximum is increased.

The following rule has been added: if chromosome A uses a subset of the variables used by chromosome B, and the response of chromosome A is higher than the response of chromosome B, then chromosome B is discarded.

THE HYBRIDIZATION WITH STEPWISE SELECTION

Generally speaking, the classical techniques are characterized by a very high exploitation and a very poor exploration: this means that, given a starting point, they are able to find the nearest local maximum, but, once they have found it, they get stuck on it. On the contrary, techniques such as GA have a very high exploration and a rather low exploitation: this means that they are able to detect several 'hills' leading to different local maxima, but that it is not very easy for them to climb up to the maximum.

It is therefore rather intuitive to think that coupling the two techniques should produce a new strategy having both high exploration and high exploitation, since the application of the classical technique to one of the solutions found by GA should lead to the identification of the local maximum near which the chromosome was lying (Hibbert, 1993).

The most used classical technique is stepwise selection, which can be used in its forward or backward form. The former adds one variable at a time, while the latter removes one variable at a time, and the procedure continues until step n + 1 leads to results not better than step n.

Both forms of stepwise selection can be very useful when combined with GA. Starting from one of the chromosomes of the population, the backward procedure allows 'cleaning' of the model, by eliminating those variables that give no relevant information but that have been selected by chance. On the other hand, the forward procedure allows those variables to enter that give relevant information but that have never been selected. Such a hybrid GA alternates generations of GA with cycles of stepwise selection, in the backward or in the forward form.

Once more, the main risk connected to this step is overfitting: a stepwise selection pushed too hard will very much improve the fitting to the training set, but can lead to a loss of true predictivity. To limit this danger, only those models are accepted that produce a relevant decrease of the prediction error (compared with that of the 'parent' model). A good value for this threshold seems to be 2%, and of course it has to be higher the higher is the risk of overfitting (i.e. with large variables/objects ratios or with a very high noise).

After their evaluation, the new models deriving from the stepwise selection and fulfilling the previous requirement will be considered as new chromosomes, and then undergo the usual steps of check of subsets and insertion into the population. According to this consideration, the sentence 'the procedure continues until step n + 1 leads to results not better than step n' will become 'the procedure continues until step n + 1 leads to results not *significantly* better than step n' (the word 'significantly' has to not be interpreted in a 'statistical' way).

Since the variables present in a chromosome are generally a small subset of the total variables, it is evident that a step in backward selection will be much faster than a step in forward selection. As a practical example, if our data set has 100 variables and our initial solution is formed by 10 variables, a backward step will require the estimation of 10 new models, each one lacking one of the selected variables. A forward step will require the estimation of 90 new models, each one with one of the unselected variables. Furthermore, since our aim is to reduce as much as possible the number of variables used, and since it is highly probable that, due to the randomness connected to the generation of the chromosomes, some non-relevant variables have also been selected, it is evident that the backward strategy should be more frequently used.

Having defined b as the frequency of backward stepwise and f as the frequency of forward stepwise, with f a multiple of b, the general structure of the algorithm is the following:

(1) start with GA;

(2) after the evaluation of b chromosomes perform backward stepwise;

(3) if $b = f$ then perform forward stepwise;
(4) go to 1.

As a general guideline, $b = 100$ and $f = 500$ are values allowing to obtain a good compromise between exploration and exploitation.

Another problem is to decide when to stop with the stepwise selection. The general algorithm for stepwise selection on v variables, starting from a subset of s selected variables, is the following:

(1) evaluate the model with s variables;
(2) for t times ($t = s$ in backward, $t = v - s$ in forward) evaluate the models with p variables ($p = s - 1$ in backward, $p = s + 1$ in forward);
(3) if the best model evaluated in step 2 is better than that evaluated in step 1 (producing a decrease of the prediction error greater than the predefined value), then go to step 1, with the corresponding subset of variables being the starting model; else end.

The stepwise selection is usually performed on the best chromosome. It can happen anyway that since the last cycle of stepwise the GA didn't find any new better chromosome. In this case, the first chromosome has already undergone stepwise selection, and it would then be useless to repeat it. A downward search among the chromosomes of the population would detect the first one which has not yet undergone stepwise, and this one will be the starting chromosome.

Table III shows a comparison among classical methods, GAs and hybrid GAs.

Table III *Comparison between different strategies.*

Classical methods
– Perform local search, finding a local maximum
– At every step the domain in which the search takes place is reduced
– Produce a single result (the 'best' result)

Genetic algorithms
– No local search
– At every moment every point of the experimental domain can be explored
– Produce a series of almost equivalent results (the user can choose the 'best')

Hybrid genetic algorithms
– Perform local search, finding local maxima
– At every moment every point of the experimental domain can be explored
– Produce a series of almost equivalent results (the user can choose the 'best')

THE PROBLEM OF FULL-VALIDATION

The approach so far described has one main drawback: since the prediction and the choice of the variables are performed on the same objects, the results obtained are not fully validated. Therefore, they cannot be considered as expressing the real predictive capability of the model, and this overestimation increases with the ratio variables/objects and with the number of chromosomes evaluated.

As a confirmation, try to create a data set with, say, 10 objects, 100 X variables and 1 response, in which each datum is a random number. With such a dimensionality, whatever the technique of feature selection you choose, you will always find a subset of variables almost perfectly 'predicting' the response. It is anyway evident that the only fact that you could find a good relationship between the X variables and the response doesn't mean that you can rely on that model to predict the response of other objects of the same data set (i.e. objects whose response is a random number).

The only way to be sure of the predictive ability of a model is to test it on objects that never took part to its definition. A good approach could be the following:

- divide the objects into *d* deletion groups;
- for *i* = 1 to *d* outer cycles
  - perform the GA on the objects not constituting deletion group *i*
  - test the chromosomes of the final population on the objects of deletion group *i*;
- next *i*;
- evaluate the predictions in the *d* outer cycles. As a first guess, one can say that the real predictive ability of the model computed with all the objects will be not worse than the prediction obtained in the worst outer cycle. More information will also be given by the difference between the partial and the full validated predictions obtained with the same subset of variables (the lower it is, the higher the stability of the model and then the probability that the results obtained can have the same value also on a new data set coming from the same population).

In the case of a large number of objects, a much simpler approach is the 'single evaluation set'. The objects are split into a training set, on which the model is computed, and a validation set, on which the model will be validated. Apart from the problem deriving from the fact that the two sets must be homogeneous, this approach will never take into account the information brought by the objects of the evaluation set, and therefore it is hardly applicable in the case of QSAR, since the number of molecules under study is usually not so great to allow to renounce *a priori* to the information brought by some of them.

As one can see, the need of a good validation becomes more and more important at the same conditions at which it becomes more and more difficult to perform it!

TWO QSAR EXAMPLES

Two elaborations will be performed on two QSAR data sets published in the literature, and the results obtained after the application of GA will be compared with the conclusions of the authors of the papers.

Data set ANTIM

Selwood and coworkers published a paper (Selwood *et al.*, 1990) in which a set of antifilarial antimycin analogues described by 53 X variables was studied, with the goal of correlating the structure with the antifilarial activity by a MLR model (Table IV).

The authors used objects 1–16, available at a first time, as training set, and objects 17–31, available only at a following time, as evaluation set. They wanted to use a MLR model, without using PLS; to reduce the number of variables they used the stepwise technique, selecting variables 24, 50, and 51. Figure 2 shows the plot of the observed vs. computed (for the training set) or predicted (for the evaluation set) values. Even discarding objects 17, 18, and 24, that are clear outliers, the predictive ability surely cannot be satisfying: the predicted variance is only 29.74%, with a residual mean square error in prediction (RMSEP) of 0.653.

A deeper analysis of the training and of the evaluation sets reveals that one of the reasons of such a poor performance is that some objects to be predicted are out of the domain spanned by the training set. Figure 3 shows the PC1-PC2 plot of the 31 molecules (each described by the 53 X variables). Though this plane explains only 45.69% of total variance, it is very clear that objects 17, 18, 21, 22, 25, and 31 are well outside the space encompassed by the objects of the training set (1–16). Since their leverage is very high, the prediction error on these objects will be very high, whatever the model computed with objects 1–16. As a consequence, these objects too should be discarded from the evaluation set, that is then composed by the following 8 objects: 19, 20, 23, 26, 27, 28, 29, and 30.

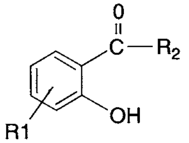
On this evaluation set, a much more acceptable value of 58.38% of explained variance (RMSEP = 0.567) has been obtained.

GAs have then been applied, both in the 'partial validated' and in the 'full validated' form, with the goal of comparing the predictive ability of the selected subsets. Each version has been run five times, and then the subset giving the best result on the training set has been checked on the evaluation set.

Table IV *Data set ANTIM.*

Compounds:

R1	*R2*
1) 3-NHCHO	$NHC_{14}H_{29}$
2) 3-NHCHO	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
3) 5-NO_2	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
4) 5-SCH_3	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
5) 5-$SOCH_3$	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
6) 3-NO_2	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
7) 5-CN	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
8) 5-NO_2	NH-4-(4-$CF_3C_6H_4O)C_6H_4$
9) 3-SCH_3	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
10) 5-SO_2CH_3	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
11) 5-NO_2	NH-4-$(C_6H_5O)C_6H_4$
12) 5-NO_2	NH-3-Cl-4-(4-$ClC_6H_4CO)C_6H_3$
13) 5-NO_2	NH-4-(2-Cl-4-$NO_2C_6H_3O)C_6H_4$
14) 5-NO_2	NH-3-Cl-4-(4-$CH_3OC_6H_4O)C_6H_3$
15) 3-SO_2CH_3	NH-3-Cl-4-(4-$ClC_6H_4O)C_6H_3$
16) 5-NO_2	NH-3-Cl-4-(4-$ClC_6H_4S)C_6H_3$
17) 3-NHCHO	NHC_6H_{13}
18) 3-NHCHO	NHC_8H_{17}
19) 3-$NHCOCH_3$	$NHC_{14}H_{29}$
20) 5-NO_2	$NHC_{14}H_{29}$
21) 3-NO_2	$NHC_{14}H_{29}$
22) 3-NO_2-5-Cl	$NHC_{14}H_{29}$
23) 5-NO_2	NH-4-$C(CH_3)_3C_6H_4$
24) 5-NO_2	$NHC_{12}H_{25}$
25) 3-NO_2	$NHC_{16}H_{33}$
26) 5-NO_2	NH-3-Cl-4-(4-$ClC_6H_4NH)C_6H_3$
27) 5-NO_2	NH-4-(3-$CF_3C_6H_4O)C_6H_4$
28) 5-NO_2	NH-3-Cl-4-(4-$SCF_3C_6H_4O)C_6H_3$
29) 5-NO_2	NH-3-Cl-4-(3-$CF_3C_6H_4O)C_6H_3$
30) 5-NO_2	NH-4-$(C_6H_5CHOH)C_6H_4$
31) 5-NO_2	4-ClC_6H_4



Descriptors:

1–10) partial atomic charges for atoms 1–10

11–13) vectors (X, Y, and Z) of the dipole moment

14) dipole moment

15–24) electrophilic superdelocalizability for atoms 1–10

25–34) nucleophilic superdelocalizability for atoms 1–10

35) van der Waal's volume

36) surface area

37–39) moments of inertia (X, Y, and Z)

40–42) principal ellipsoid axes (1, 2, and 3)

43) molecular weight

44–46) parameters describing substituent dimensions in the X, Y, and Z

47–49) parameters describing directions and the coordinates of the center of the subsituent

50) calculated log P

51) melting point

52–53) sums of the F and R substituent constants

Response: antifilarial activity

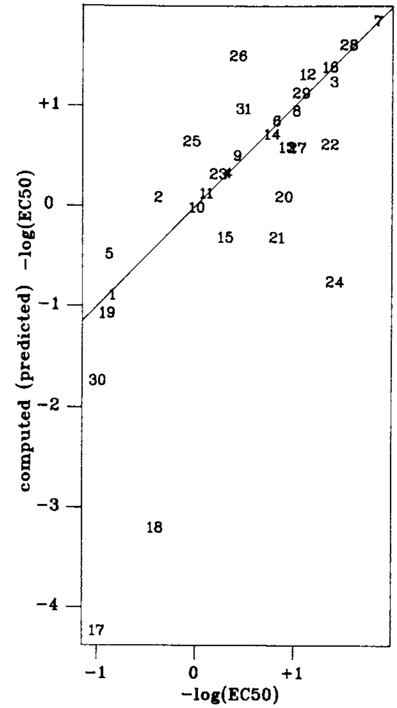


Figure 2 Data set ANTIM: observed vs. computed (obj. 1–16) or predicted (obj. 17–31) values (variables 24, 50, and 51, stepwise selection).

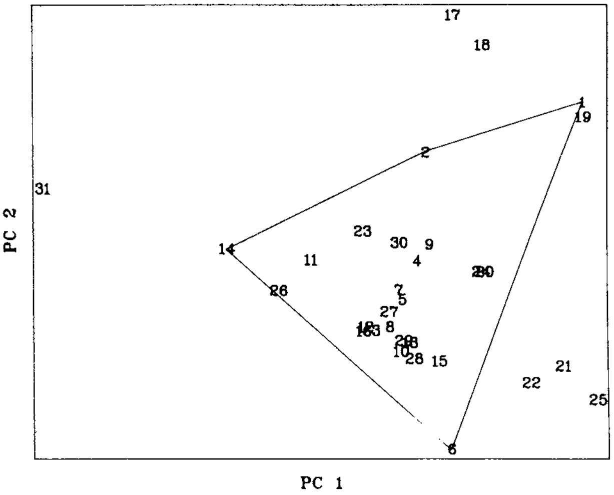


Figure 3 Data set ANTIM: the plot PC1 vs. PC2 shows that some objects of the evaluation set are well outside the domain spanned by the objects of the training set (1–16).

Table V *Configuration of the genetic algorithm.*

	Partial validation	Full validation
size of the population	30	20
probability of initial selection	0.0943	0.0943
probability of cross-over	0.5	0.5
probability of mutation	0.01	0.01
frequency of backward elimination	100	100
frequency of forward selection	500	–
minimum reduction in prediction error to accept stepwise	5%	5%
threshold value	–	70%
outer cycles	–	5
deletion groups	5	5
stop criterion	(a) 1 hour (b) 2' (+ 1 stepwise)	300 ev. (+ 1 stepwise)

Table V shows the configurations of the algorithm. All the parameters are set in such a way that the growth of the response is as fast as possible. The minimum reduction in prediction error to accept a model deriving from stepwise selection has been set a higher value (5%), since the variables–objects ratio is very large and the data are rather noisy.

The GA in partial validation (stop criterion 1h) selected variables 7, 13, 16, 19, 45, 50, and 51, with a response (cross-validated variance on the training set) of 98.67%. When used to predict the objects of the validation set, the predicted variance was only 22.59% (RMSEP 0.773). This is a confirmation that with such a large variables–objects ratio the partial validation used as response criterion can lead to highly overfitted models, with no real predictive ability at all.

When observing the evolution of the response in time, one can see that, after having improved very much in the first few minutes, it reaches a plateau. The interpretation of this phenomenon is very simple: the algorithm, whose configuration is planned to reach in a short time a very high response, can recognize very quickly the general structure of the data; from that moment on, the small improvements are mainly due to the adaptation of the models to that particular data set (random correlations, random predictions, structure of the deletion groups, . . .).

Of course, the main problem is to detect when to stop. A great help in taking that decision can come from a randomization test. In it, the responses in the response vector are randomized, so that each molecule will have the activity of another molecule. A GA is run on that data set, and the evolution of the response is recorded. One can consider that these results are noise, and the results obtained with the 'true' data set are information + noise; as a consequence, the stop criterion will be easily detected as the moment in

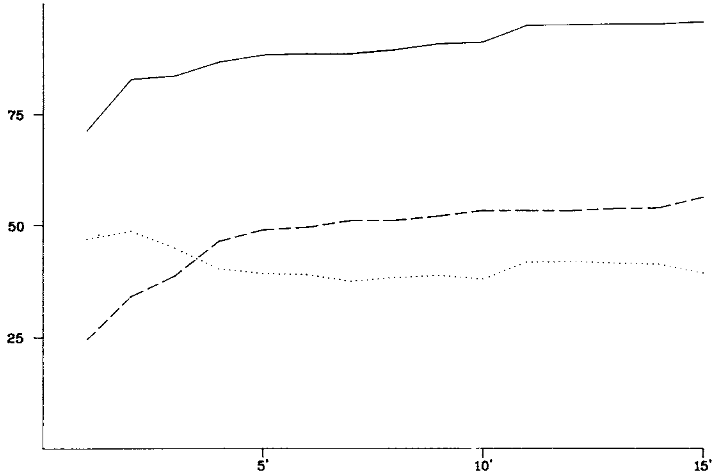


Figure 4 Data set ANTIM: evolution of response (predicted variance in partial validation) vs. time. Solid line: 'true' data set; dashed line: 'randomized' data set; dotted line: difference (each line is the average of 5 runs; in the case of randomization test 5 different randomizations have been performed).

which the difference between the 'true' and the 'random' response is the greatest (Figure 4).

From the data obtained by 5 runs on randomized data and 5 runs on 'true' data, it resulted that the highest difference between the average responses was obtained after 2' (about 300 evaluations). Stopping the elaboration at such an early time reduces very much the overfitting, as shown by the fact that the selected subset (variables 19, 21, 50, and 51) explains 71.50% of the variance of the evaluation set (RMSEP 0.469).

The full-validated version selected variables 16, 19, 50, and 51. With this subset, the variance predicted on the objects of the evaluation set is 79.62% (RMSEP 0.397), significantly better than that obtained with the 3 variables selected by the stepwise technique. Figure 5 shows the plot of the observed vs. computed (for training set) or predicted (for evaluation set) values. In the case of full-validation, the stop criterion is not as critical as in the partially validated version, since the algorithm itself will eliminate the low predicting models.

This example confirms that with a very large variables/objects ratio the partial validation, unless a previous study is performed, leads to highly overfitted models, with no real predictive ability at all; on the contrary, the algorithm based on full validation also in these difficult conditions gives results better than those obtained by classical methods. It is also interesting to notice that the 4 variables selected by the fully validated version are a subset of the 7 variables selected by the partially validated one (stop criterion 1h): it

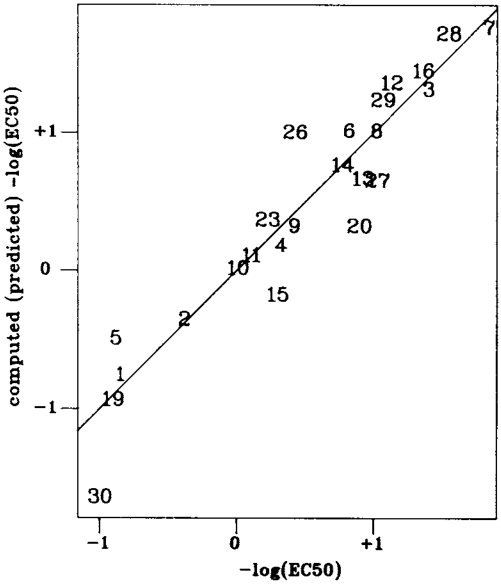


Figure 5 Data set ANTIM: observed vs. computed (obj. 1–16) or predicted values (variables 16, 19, 50, and 51, genetic algorithm).

is therefore very clear that variables 7, 13, and 45 have been selected due to a random correlation with the response.

Data set HEME

Hellberg and coworkers published a paper (Hellberg *et al.*, 1985) in which a data set of halogenated ethyl methyl ethers (HEME) was described by 41 X variables (molecular descriptors) and 2 responses (anesthetic activity and toxicity).

The total number of molecules was 22, but not each of them had both responses: 14 objects had the anesthetic activity, 10 the toxicity, and 5 had none of them. The first response will be studied, and then the data set will be formed by 14 objects, 41 descriptors and 1 response (Table VI).

Due to the very limited number of objects, it is impossible to think to an independent validation set, and then the only possible form of prediction will be cross-validation.

The authors applied PLS to the whole data set, with the 2-component model explaining 86% of the variance in fitting; by cross-validation, the best predictions are obtained with 5 components (predicted variance 75.05%).

A cluster analysis applied to the variables (Figure 6) shows that they form well defined clusters, and that several variables have a very high similarity. At a similarity level of 0.6 the following 9 clusters are identified:

Table VI *Data set HEME.*

Compounds:

1) CH$_3$-CF$_2$-O-CF$_2$Cl
2) CHF$_2$-CH$_2$-O-CH$_3$
3) CH$_2$F-CF$_2$-O-CHF$_2$
4) CH$_2$Cl-CF$_2$-O-CF$_3$
5) CF$_2$Cl-CH$_2$-O-CH$_3$
6) CHCl$_2$-CF$_2$-O-CF$_3$
7) CH$_2$Cl-CF$_2$-O-CF$_2$Cl
8) CFCl$_2$-CF$_2$-O-CHCl$_2$
9) CFCl$_2$-CF$_2$-O-CCl$_3$
10) CHCl$_2$-CF$_2$-O-CHF$_2$
11) CHCl$_2$-CF$_2$-O-CH$_2$F
12) CCl$_3$-CF$_2$-O-CHF$_2$
13) CHFCl-CF$_2$-O-CHCl$_2$
14) CHCl$_2$-CF$_2$-O-CH$_2$Cl

Descriptors:

1) log P
2) (log P)^2
3) molar volume
4) molecular weight
5) mean molecular polarizability
6) Σπ - C1
7) (Σπ)^2 - C1
8) Σπ - C3
9) (Σπ)^2 - C3
10) charge of C1
11) electronegativity of C1
12) charge of C2
13) electronegativity of C2
14) charge of C3
15) electronegativity of C3
16) Q of most E halogen on C1
17) E of most E halogen on C1
18) Q of most E halogen on C2
19) E of most E halogen on C2
20) Q of most E halogen on C3
21) E of most E halogen on C3
22) Q of most acidic hydrogen
23) E of most acidic hydrogen
24) Q of the oxygen
25) E of the oxygen
26) ΔQ - HC
27) ΔE - HC
28) E - CH
29) Σ Q - C1
30) Σ Q - C2
31) Σ Q - C3
32) Σ Q - (Q - C1)
33) Σ Q - (Q - C2)
34) Σ Q - (Q - C3)
35) Σ Q * (Q - C1)
36) Σ Q * (Q - C2)
37) Σ Q * (Q - C3)
38) strongest H-bonder
39) 2nd strongest H-bonder
40) (O) strongest H-bonder
41) (O) 2nd strongest H-bonder

Response: anesthetic activity

- Cluster 1: variables 1, 2, 3, 4, 5, 8, 9 (bulk, lipophilicity and polarizability);
- Cluster 2: variables 6, 7 (lipophilicity on C1);
- Cluster 3: variables 12, 13, 30, 33, 36 (description of C2);
- Cluster 4: variables 10, 11, 16, 17, 19, 20, 24, 29, 32, 35 (mainly, description of C1);
- Cluster 5: variables 14, 15, 21, 25, 31, 34, 37 (mainly, description of C3);
- Cluster 6: variables 18, 26;
- Cluster 7: variables 22, 23, 27, 28 (most acidic hydrogen);
- Cluster 8: variables 38, 40 (strongest H-bonder);
- Cluster 9: variables 39, 41 (second strongest H-bonder).

Since one can expect a very high degree of redundancy on the data, a rather important reduction in the number of the variables should be possible.

GA has been applied in the full-validated version, with the same structure

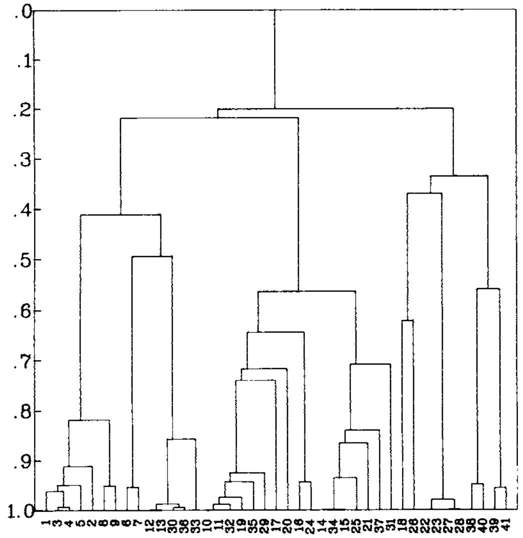


Figure 6 Data set HEME: dendrogram of the variables (distance: absolute value of the correlation coefficient; average linkage, weighted).

of the previous example; the only differences were the initial probability of selection (0.0732) and the threshold (75%). The best result obtained in the five runs selected 6 variables (4, 6, 7, 17, 31, 39); the most predictive model was the one with five components and the outer cycle with the worst prediction in full-validation scored 82.44%. This means that in any of the five outer cycles at least one combination of variables giving a prediction higher than 82% on the independent evaluation set has been found. Of course, since no external evaluation set is present, no direct final comparison of the two models can be performed; anyway, the fact that the 75% threshold has been by far succesfully fulfilled in each outer cycle means that the global predictive ability is at least comparable with that obtained with all the variables.

When taking into account the selected variables, one can see that, except for variables 6 and 7, each of them is in a different cluster; this is another confirmation of the good choice performed by the GA.

ACKNOWLEDGEMENTS

The author wishes to thank Drs Carlos Lucasius and David Rogers for the very useful scientific discussions held during the *Second International Workshop on Neural Networks and Genetic Algorithms Applied to QSAR and Drug Design* (Lyon, France, June 12–14, 1995).

This work received financial support from the Ministry of University and Scientific Research (MURST 40% and 60%) and from the CNR (Italian National Research Council), Comitato Scienze e Tecnologia Informazione.

The source codes of the programs are available from the author upon request.

REFERENCES

Hellberg, S., Wold, S., Dunn III, W.J., Gasteiger, J., and Hutchings, M.G. (1985). The anesthetic activity and toxicity of halogenated ethyl methyl ethers, a multivariate QSAR modelled by PLS. *Quant. Struct.-Act. Relat.* **4**, 1–11.

Hibbert, D.B. (1993). A hybrid genetic algorithm for the estimation of kinetic parameters. *Chemom. Intell. Lab. Syst.* **19**, 319–329.

Lanteri, S. (1992). Full validation procedures for feature selection in classification and regression problems. *Chemom. Intell. Lab. Syst.* **15**, 159–169.

Leardi, R. (1994). Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection. *J. Chemom.* **8**, 65–79.

Leardi, R., Boggia, R., and Terrile, M. (1992). Genetic algorithms as a strategy for feature selection. *J. Chemom.* **6**, 267–281.

Lucasius, C.B. and Kateman, G. (1991). Genetic algorithms for large-scale optimization in chemometrics: An application. *Trends Anal. Chem.* **10**, 254–261.

Lucasius, C.B., Beckers, M.L.M., and Kateman, G. (1994). Genetic algorithms in wavelength selection: A comparative study. *Anal. Chim. Acta* **286**, 135–153.

Selwood, D.L., Livingstone, D.J., Comley, J.C.W., O'Dowd, A.B., Hudson, A.T., Jackson, P., Jandu, K.S., Rose, V.S., and Stables, J.N. (1990). Structure–activity relationships of antifilarial antimycin analogues: A multivariate pattern recognition study. *J. Med. Chem.* **33**, 136–142.