Evolutionary Optimization in Quantitative Structure−Activity Relationship: An Application of Genetic Neural Networks

Sung-Sau So*,[†] and Martin Karplus*,[†,‡]

Department of Chemistry, Harvard University, 12 Oxford Street, Cambridge, Massachusetts 02138, and Institut le Bel, Université Louis Pasteur, 4, Rue Blaise Pascal, 67000 Strasbourg, France

Received September 25, 1995[⊗]

A new hybrid method (GNN) combining a genetic algorithm and an artificial neural network has been developed for quantitative structure−activity relationship (QSAR) studies. A suitable set of molecular descriptors are selected by a genetic algorithm. This set serves as input to a neural network, in which model-free mapping of multivariate data is performed. Multiple predictors are generated that are superior to results obtained from previous studies of the Selwood data set, which is used to test the method. The neural network technique provides a graphical description of the functional form of the descriptors that play an important role in determining drug activity. This can serve as an aid in future design of drug analogues. The effectiveness of GNN is tested by comparing its results with a benchmark obtained by exhaustive enumeration. Different fitness strategies that tune the evolution of genetic models are examined, and QSARs with higher predictiveness are found. From these results, a composite model is constructed by averaging predictions from several high-ranking models. The predictions of the resulting QSAR should be more reliable than those derived from a single predictor because it makes greater use of information and also permits error estimation. An analysis of the sets of descriptors selected by GNN shows that it is essential to have one each for the steric, electrostatic, and hydrophobic attributes of a drug candidate to obtain a satisfactory QSAR for this data set. This type of result is expected to be of general utility in designing and understanding QSAR.

I. Introduction

Quantitative structure−activity relationships (QSARs) correlate biological activities of candidate drugs with their physicochemical parameters. They have evolved over a period of 30 years from a simple regression model with a few electronic or thermodynamics variables[1] to an important discipline that is being applied to a wide range of problems.[2−4] Efforts by researchers from different fields, publication of dedicated journals, and organization of specialist conferences, as well as the decreasing cost of computer power, are all contributing to rapid advances in this field. Two major developments in QSARs have been made in recent years. First is the introduction of a wide range of novel molecular descriptors, such as molecular connectivity and other graph theory based topological indices,[5−7] molecular similarity matrices derived from similarity calculations of electrostatic, shape, or other physicochemical parameters,[8−11] and the application of molecular fields in a three-dimensional lattice environment. Of particular interest is the CoMFA approach[12] which complements the usual two-dimensional descriptors with three-dimensional information. Second, many sophisticated feature mapping techniques have been introduced for the determination of QSARs that go significantly beyond the original linear regression analysis. Principal component analysis,[13] nonlinear mapping,[13−15] partial least squares,[12,16] and neural networks[17−20] are a few examples. Most recently, hybrid approaches which integrate various optimization and mapping methods have begun to be investigated. Rogers *et al.*[21] and Luke[22] combined genetic algorithms with regression analysis; Sutter *et al.* developed the generalized simulated annealing method that makes use of a simulated annealing algorithm and a neural network.[23] In each case, the potential of the new hybrid approach was demonstrated by the development of improved QSAR models, often for a problem that had been studied previously with more standard techniques. In the present study we report a hybrid method (GNN) that combines a genetic algorithm with a neural network. It is shown to be superior to all published approaches for the Selwood data set,[24] which has become a standard for testing QSAR.

Once the biological activities of a series of related candidate drug has been determined, QSAR models are typically constructed in several steps. The first step is the tabulation of experimental or computational physicochemical parameters which provide a description of the similarities and differences of the compounds under investigation. This process is generally straightforward because many of the available computer-aided molecular design (CAMD) packages[25,26] are developed to deal with this kind of calculation, often with great ease. In many cases, a standard set of descriptors chosen from experience is used. Although it seems likely that improved descriptors could be introduced, that is not our present concern. The next step is to apply a statistical or pattern recognition method to correlate these molecular descriptors with the observed biological activities. This is often a complex task, particularly when the number of descriptors exceeds the number of compounds in the data set, so that one is dealing with an undetermined problem where undesirable overfitting can result.[17,18,27] This problem can be avoided by preprocessing the descriptor set with a feature selection routine that determines which of the descriptors have a significant

[†] Harvard University.
[‡] Université Louis Pasteur.
[⊗] Abstract published in *Advance ACS Abstracts,* March 1, 1996.

influence on the activity of a given compound. In the past the selection was made by a human expert who relied on experience and scientific intuition, or by a correlation analysis of the data set that applied statistical methods such as forward selection or backward elimination.[24] When the dimensionality of the data is high and the interrelations between variables are convoluted, human judgment can be unreliable. Also, a simple forward or backward stepping algorithm fails to take into account the information that involves the combined effect of several features, so that the optimal solution is not necessary obtained.[21,28] This suggests the need for a method which is applicable to complex multivariate data, is easy to use, and, of course, supplies a good solution to the problem. Genetic algorithms, which are clearly well-suited to tackle problem of this kind, were introduced to the field of QSARs to address this need.[21,22] After the most relevant features have been selected, the final stage of the QSAR model building is executed by a feature-mapping procedure. Traditionally this has been a multiple linear regression analysis, which, by its name, is a linear method. Nonlinear correlation in the data had to be explicitly dealt with by a predetermined functional transformation. Unfortunately the introduction of nonlinear or cross-product terms in a regression equation often requires knowledge that is not available. Moreover, it adds to the complexity of the problem and often leads to no significant improvement in the resulting QSAR. To overcome this deficiency in linear regression, researchers have begun to use intrinsically nonlinear techniques such as the neural networks.[29,30] Several QSAR studies with neural networks have demonstrated that they fit existing data well and often lead to a model that predicts new compounds with high accuracy.

In this paper we propose a QSAR approach that makes use of a genetic algorithm to select the descriptors and a neural network as the tool to correlate the selected descriptors with activity. The integration of these two optimization methods is shown to lead to a significant improvement over existing methods for a well-studied example. We suggest that the genetic neural network (GNN) algorithm is close to the limit of what the traditional Hansch-type QSAR can achieve. The next major advance in QSARs is likely to be based on the optimized use of three-dimensional data.

Section II presents the method that we have developed. The results are presented and analyzed in Section III. Section IV outlines the conclusions.

II. Method

Genetic Algorithms. The genetic algorithm is used to select the features that are most significant for the molecular data set. Genetic algorithms are stochastic optimization methods that have been inspired by evolutionary principles.[31] The distinctive aspect of a genetic algorithm is that it investigates many possible solutions simultaneously, each of which explores different regions in parameter space.[32] In this paper two variants, genetic function approximation (GFA) and evolutionary programming (EP), were tried following the studies of Rogers *et al.*[21] and Luke[22] with a few minor modifications. In both implementations an individual in the population is represented by a string encoding the selected features. In the original studies, the fitness of the individual was determined by a function related to the residual error in the regression analysis of the training data. Here we try a variety of fitness functions which are proportional to the residual error of the training set, the test set, or even the cross-validation set from the neural network simulations.

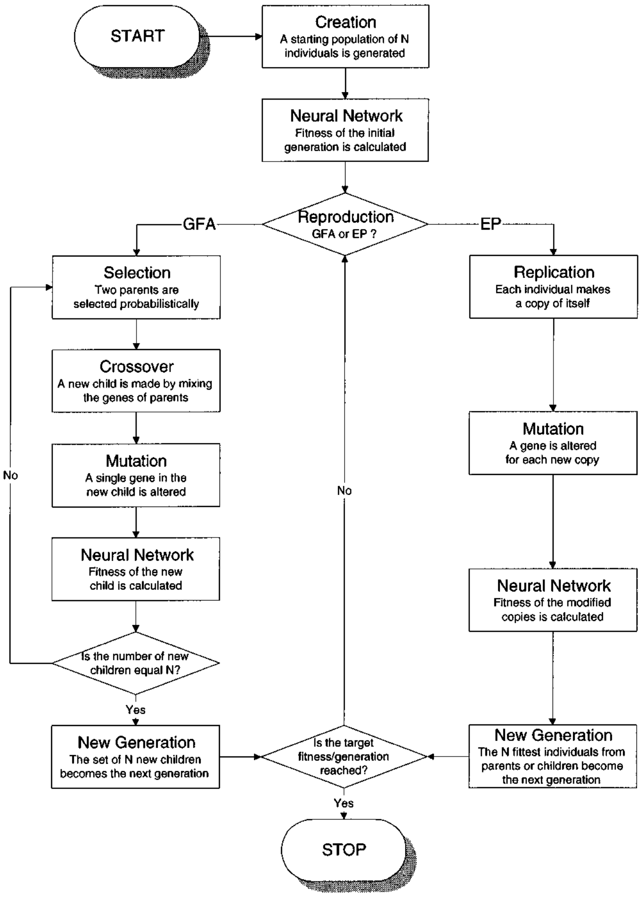

START
Creation
A starting population of N individuals is generated
Neural Network
Fitness of the initial generation is calculated
GFA
Reproduction
GFA or EP ?
EP
Selection
Two parents are selected probabilistically
Crossover
A new child is made by mixing the genes of parents
Mutation
A single gene in the new child is altered
Neural Network
Fitness of the new child is calculated
Is the number of new children equal N?
No
Yes
New Generation
The set of N new children becomes the next generation
Replication
Each individual makes a copy of itself
Mutation
A gene is altered for each new copy
Neural Network
Fitness of the modified copies is calculated
New Generation
The N fittest individuals from parents or children become the next generation
No
Is the target fitness/generation reached?
Yes
STOP

Figure 1. Flow diagram describing the strategy for the GFA and EP algorithms. The GFA enhancements, stochastic remainder, and elitism, are omitted in the diagram for clarity. See Figure 2 for the detailed descriptions of the reproduction strategy.

The basic design of the two genetic algorithms is summarized in the flow diagram shown in Figure 1. The first step in a genetic algorithm is to create a gene pool of *N* individuals. Each individual encodes the same number of descriptors; the descriptors are randomly chosen from a common list and in a way such that (1) no two individuals can have exactly the same set of descriptors and (2) all descriptors in a given individual must be different. The fitness of each individual in this generation is determined by a user-specified fitness function, and in the present case, this fitness score is computed by a neural network. The next step, where GFA differs from EP, is the reproduction process. In GFA, a sexual reproduction takes place so that the new offspring contains characteristics from both of its parents (Figure 2a). Two individuals are selected probabilistically on the basis of their fitness scores and serve as parents. Next, in a crossover each parent contributes a random selection of half of its descriptor set and a child is constructed by combining these two halves of “genetic code”. Finally, this child is subjected to a random mutation in one of its genes; i.e., one descriptor is replaced by another. This selection−crossover−mutation process is repeated until all of the *N* parents in the gene pool are replaced by their children. The fitness score of each member of this new generation is again evaluated, and the reproductive cycle is continued until a desired number of generations or target fitness score is reached. In our GFA−neural networks (GFA−NN), two modifications are made to the original implementation. The first is a stochastic reminder method[32] which allows an individual with above average fitness to be reproduced at least once. The second is the inclusion of elitism,[32] which protects the fittest individual in any given generation from crossover or mutation during reproduction. The genetic

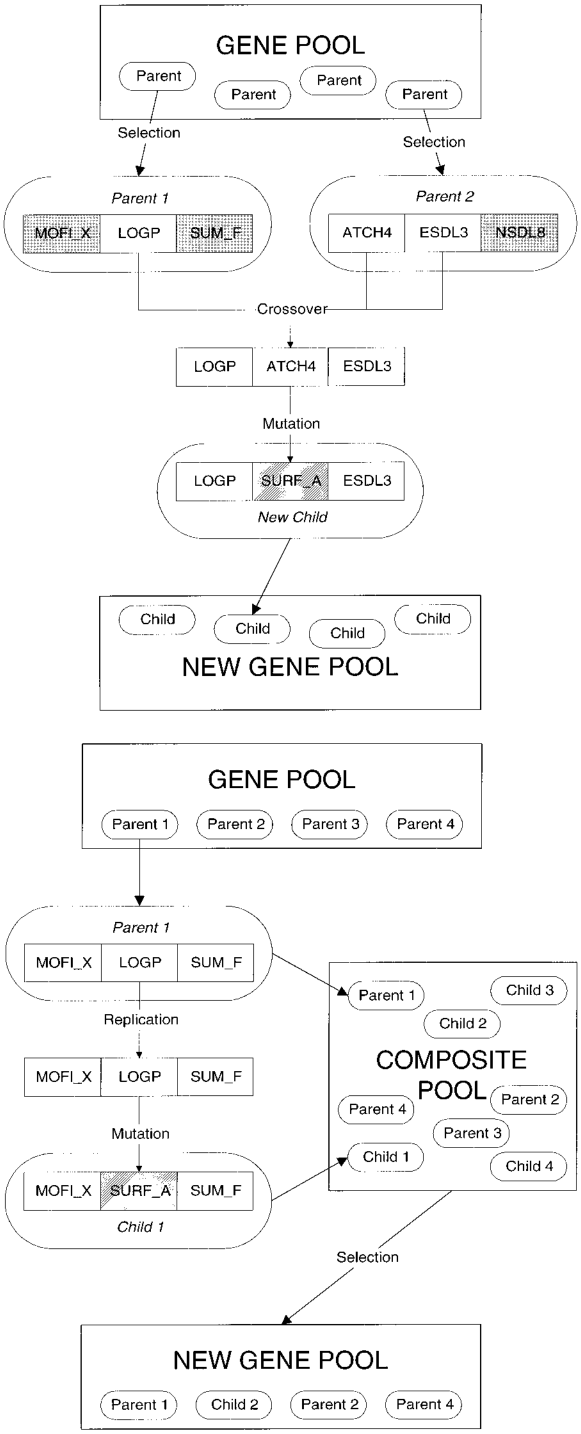


Figure 2. (a) Schematic diagram describing the sexual reproduction strategy in GFA algorithm; the reproduction of one generation is illustrated. (b) Schematic diagram describing the asexual reproduction strategy in EP algorithm. In this example Parent 1, Parent 2, Parent 4, and Child 2 are the fittest individuals in the composite pool, and they are selected as the next set of parents.

content of this individual simply moves on to the next generation intact.

Unlike GFA, EP goes through an asexual reproduction procedure so that the characteristics of each new offspring are derived from a single parent (Figure 2b). Each parent produces a child that is initially a replica of itself. The child is then subjected to a point mutation and its fitness is determined. The N fittest individuals from the composite pool containing both parents and children constituted the next generation. In this scenario the least fit parents are replaced by the fittest children, and the average fitness of the system always increases (or remained constant upon convergence) with evolutionary time.

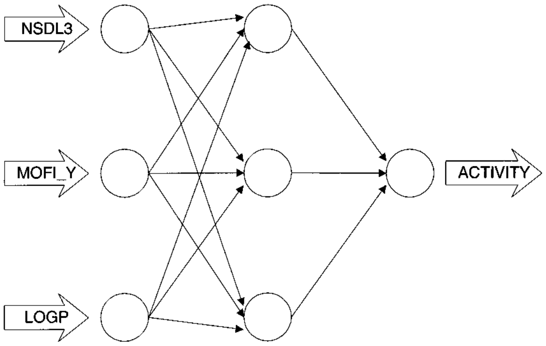


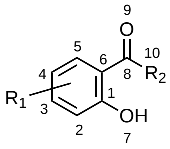
Figure 3. A prototype back-propagation neural network used in this study. It takes the three descriptors chosen by genetic algorithm (GFA or EP) as inputs and is trained against the target biological activity with a steepest descent learning algorithm.

Neural Networks. The neural network is used to perform model-free mapping of molecular descriptors with biological activity. Neural networks are computer-based simulations which contain some elements that appear to exist in living nervous systems. There have recently been numerous applications of neural network applications to QSAR, as well as many other chemical problems.[33] For details of the neural network implementation, readers are referred to relevant references or standard texts.[34,35] In this study we used a traditional steepest descent back-propagation method; a faster pseudo second derivative method for training has now been implemented and will be used in future applications. In all calculations, the same random number seed was used to initialize the network weights in the range -1.0 to $+1.0$. The learning rate and momentum parameters were set at 0.1 and 0.9 initially. An adaptive scheme[35] that optimized these two learning parameters during training was used to achieve faster convergence. The input and output vectors of the data set were scaled to take values between 0.1 and 0.9.

Earlier applications of neural networks to QSAR[17,18] have indicated that ρ, the ratio of the number of data points to the number of adjustable weights in the neural network, plays a crucial role in determining the predictive quality of the model. It is known that a neural network with an insufficient number of weights is not able to extract the relevant correlation in the data set. The analysis fails at the training stage, and unreliable predictions can result. Conversely, if the number of weights in a neural network is large compared with the amount of data, then overfitting becomes a problem. The network has the capacity to memorize the entire data set and behaves effectively as a lookup table. Thus, it is important to configure networks that make an optimum compromise between the need for generalization and the problem of memorization. Empirical studies have suggested values between 1.8 and 2.2 as the appropriate range for ρ,[17,18] and we have selected a neural network architecture in accord with this value. To allow for direct comparisons with the previous studies of the Selwood data set, all of the QSAR models used in this study were limited to three descriptors. The neural network (Figure 3) was configured with three input, three hidden, and a single output node (3$-$3$-$1). This gives rise to 16 adjustable parameters (4 biases for hidden and output nodes and 12 weights between layers) and a ρ value of 1.9 since there are 31 data points.

Model Evaluation and Fitness Functions. In this paper the quality of the fit of the training data is reflected by its residual rms error, RmsE (eq 1), or correlation coefficient, R (eq 2). The more important measure, however, is the predictive quality which is estimated by a cross-validation procedure. This method, also known as jackknifing or leave-one-out (LOO) analysis, systematically removes one data point at a time from the training set. A model is constructed on the basis of this reduced data set and is subsequently used to predict the removed sample. This procedure was repeated for all data

Table 1.[a] Structures of the Antimycin Analogues in This Study



compd	R_1	R_2	compd	R_1	R_2
1	3-NHCHO	$NHC_{14}H_{29}$	**17**	3-NHCHO	NHC_6H_{13}
2	5-NHCHO	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**18**	3-NHCHO	NHC_8H_{17}
3	5-NO_2	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**19**	3-$NHCOCH_3$	$NHC_{14}H_{29}$
4	5-SCH_3	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**20**	5-NO_2	$NHC_{14}H_{29}$
5	5-$SOCH_3$	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**21**	3-NO_2	$NHC_{14}H_{29}$
6	3-NO_2	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**22**	3-NO_2-5-Cl	$NHC_{14}H_{29}$
7	5-CN	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**23**	5-NO_2	NH-4-$C(CH_3)_3C_6H_4$
8	5-NO_2	NH-4-(4-$CF_3C_6H_4O$)C_6H_3	**24**	5-NO_2	$NHC_{12}H_{25}$
9	3-SCH_3	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**25**	3-NO_2	$NHC_{16}H_{33}$
10	5-SO_2CH_3	NH-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**26**	5-NO_2	NH-3-Cl-4-(4-ClC_6H_4NH)C_6H_3
11	5-NO_2	NH-4-(C_6H_5O)C_6H_4	**27**	5-NO_2	NH-4-(3-$CF_3C_6H_4O$)C_6H_4
12	5-NO_2	NH-3-Cl-4-(4-ClC_6H_4CO)C_6H_3	**28**	5-NO_2	NH-3-Cl-4-(4-$SCF_3C_6H_4O$)C_6H_3
13	5-NO_2	NHs-4-(2-Cl-4-$NO_2C_6H_3O$)C_6H_4	**29**	5-NO_2	NH-3-Cl-4-(3-$CF_3C_6H_4O$)C_6H_3
14	5-NO_2	NHS-3-Cl-4-(4-ClC_6H_4O)C_6H_3	**30**	5-NO_2	NH-4-(C_6H_5CHOH)C_6H_4
15	3-SO_2CH_3	NHSd-3-Cl-4-(4-$CH_3OC_6H_4O$)C_6H_3	**31**	5-NO_2	4-ClC_6H_4
16	5-NO_2	NH-3-Cl-4-(4-ClC_6H_4S)C_6H_3			

[a] Data taken from ref 24.

Table 2. Physicochemical Descriptors in the Selwood Data Set

symbol	descriptions
ATCH 1−10	partial atomic charges for atoms 1−10
ESDL 1−10	electrophilic superdelocalizabilities for atoms 1−10
NSDL 1−10	nucleophilic superdelocalizabilities for atoms 1−10
DIP_MOM,DIP_X/Y/Z	dipole moment and vectors in X/Y/Z-direction
VDW_VOL	van der Waals volume
SURF_A	surface area
MOL_WT	molecular weight
MOFI_X/Y/Z	principal moments of inertia in *X*/*Y*/*Z*-direction
PEAX_X/Y/Z	principal ellipsoid axes in *X*/*Y*/*Z*-direction
S8_DX/Y/Z	substituent dimensions of atom 8 in *X*/*Y*/*Z*-directions
S8_1CX/Y/Z	substituent coordinates of atom 8 in *X*/*Y*/*Z*-directions
LOGP	calculated log partition coefficient for octanol/water
M_PNT	measured melting point
SUM_F	sum of *F* substituent constants
SUM_R	sum of *R* substituent constants

points so that a complete set of predicted values and the corresponding cross-validated variables (XRmsE and XR) are obtained.

$$\text{residual rms error (RmsE)} = \sqrt{\frac{\sum_{i=1}^{N}(\text{activity}_{\text{calc},i} - \text{activity}_{\text{obs},i})^2}{N}} \qquad (1)$$

$$\text{correlation coefficient } (R) = \frac{\sum_{i=1}^{N}(\text{activity}_{\text{calc},i} - \overline{\text{activity}_{\text{calc}}})(\text{activity}_{\text{obs},i} - \overline{\text{activity}_{\text{obs}}})}{\sqrt{(\sum_{i=1}^{N}\text{activity}_{\text{calc},i}{}^2 - N\,\overline{\text{activity}_{\text{calc}}}{}^2)(\sum_{i=1}^{N}\text{activity}_{\text{obs},i}{}^2 - N\,\overline{\text{activity}_{\text{obs}}}{}^2)}} \qquad (2)$$

The fitness function of a model in a genetic-neural network (GNN) simulation is defined by use of one of the above statistical variables in the following way:

$$\text{fitness}_{\text{RmsE}} = \frac{1}{\text{RmsE}} \qquad (3a)$$

$$\text{fitness}_R = 1 + R \qquad (3b)$$

The range of fitness scores spanned by the two types of fitness functions are different: with $\text{fitness}_{\text{RmsE}}$ the score is between 0 and ∞, and with fitness_R it is between 0 and 2. For either definition, the higher the score of a model, the fitter it would be and consequently the more probable would be its survival in a long run.

Data Set. The series of 31 antifilarial antimycin analogues (Table 1) reported by Selwood[24] were used in this study. Each compound was parameterized with 53 physicochemical descriptors that are listed in Table 2; the descriptors used correspond to those selected by Selwood. The QSARs of this set of compounds have been extensively studied, so that detailed comparisons can be made with other results. Selwood performed a forward-stepping multivariate regression analysis on the data set and generated a three-descriptor regression QSAR. Wikel and Dow made a descriptor selection by extracting information from the connecting weights of a full neural network simulation and constructed an improved regression QSAR.[36] Rogers and Hopfinger integrated genetic programming with regression analysis in their formulation of the genetic function approximation (GFA) algorithm.[21] They found some linear QSAR models that give much better results than the Selwood or the Wikel models. Recently, Luke tried a different genetic approach, known as evolutionary programming (EP),[22] and found some good models that GFA had missed.

Implementation. A computer program in the C++ programming language has been written to perform the tasks of neural network computation and genetic reproduction outlined above. All calculations were done on Hewlett-Packard 735/125 workstations. The time requirement for the training of a single three-descriptor neural network was approximately 1 CPU second. A typical genetic neural network simulation with

Table 3. Comparison of Linear regression and Neural Networks[a]

model	descriptors			regression		neural network	
				R	XR	*R*	XR
Selwood	ESDL10	LOGP	M_PNT	0.737	0.667	0.813	0.665
Wikel	ATCH4	MOFI_X	LOGP	0.774	0.679	0.836	0.710
GFA1	MOFI_Y	LOGP	SUM_F	0.849	0.804	0.869	0.798
GFA2	ESDL3	SURF_A	LOGP	0.848	0.803	0.871	0.751
GFA3	ESDL3	MOFI_Y	LOGP	0.838	0.777	0.839	0.728
EP1	MOFI_Y	LOGP	SUM_F	0.849	0.804	0.869	0.798
EP2	ESDL3	SURF_A	LOGP	0.848	0.803	0.871	0.751
EP3	MOFI_Z	LOGP	SUM_F	0.847	0.804	0.865	0.808
EP4	ESDL3	MOFI_Y	LOGP	0.838	0.777	0.839	0.728
EP5	ESDL3	MOFI_Z	LOGP	0.838	0.781	0.830	0.598

[a] The regression correlation coefficients (R) and cross-validated correlation coefficients (XR) for the reported models are given. Some of the models derived from the two different genetic algorithms are identical. GFA1 = EP1, GFA2 = EP2, and GFA3 = EP4. There are only seven distinct models. While we realize that perhaps no more than two decimal points are significant for these correlation coefficients, we report our results to the third decimal point for comparison with earlier QSAR of this data.

a population of 300 individuals going through 50 generations required 4−5 CPU hours.

III. Results

Neural Network Simulations on Regression Models. Several three-descriptor multiple linear regression QSAR models have been suggested by the four previous studies described in the Data Section. Our initial investigation used the same models and replaced regression analysis by neural networks. The results are shown in Table 3. In most cases, the neural network models marginally outperformed regression models in fitting training data. However, comparison of the cross-validation results shows that all but two of the networks exhibited considerably inferior predictive power than the corresponding regression models. We recall that the choice of descriptors in these models, particularly the last eight, is optimized in conjunction with a multivariate linear regression analysis. The fact that the regression models are satisfactory QSARs in terms of both training (R) and prediction (XR) suggests that the chosen sets of descriptors have an approximately linear relationship with biological activity. To investigate this possibility a neural network monitoring scheme was used.[18] First, the neural network model that was under investigation was trained in a normal way. After training, the variation of the output value (in this case the activity) was monitored on changing the value of one input while keeping the remaining network inputs at a constant value. This procedure was repeated for all other network inputs. The plot of the functional dependence of MOFI_Y, LOGP, and SUM_F for the top genetic-regression model (GFA1/EP1) is shown in Figure 4. These response curves were correlated with linear models, as depicted by the straight dotted lines in the same plot. The R^2 values of 0.72, 0.90, and 0.99 obtained for MOFI_Y, LOG, and SUM_F, respectively, demonstrate that this model is essentially linear. However, this does not explain the poorer performance of the neural networks, which should also be able to treat linear relationships. One possible reason for this somewhat surprising result, as suggested by a reviewer, is that the neural network result is based on a single simulation, so that the prediction may be derived from a poorly optimized run.

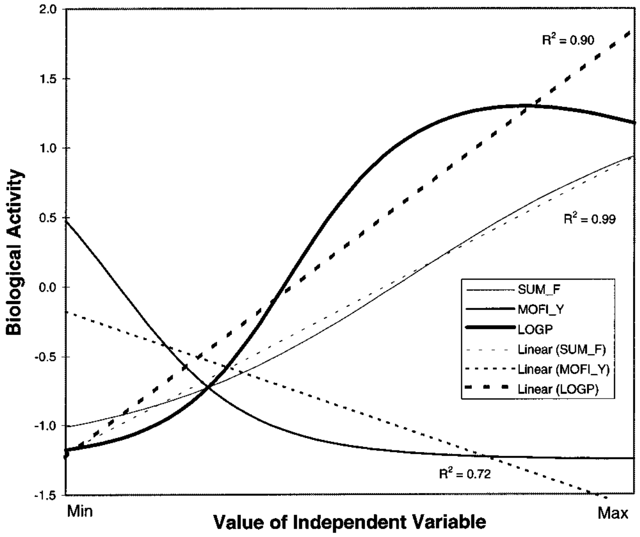


Figure 4. Biological activity as a function of the individual physicochemical parameter for the best genetic-regression model (GFA1 and EP1). The best fit linear models (dotted lines) and their R^2 values are also shown.

Another question concerns the suitability of the descriptors in the data set for testing a method like a neural network that can take into account nonlinear relationships. It is not unexpected that a number of descriptors, like ESDL3, LOGP, and SUM_F, as well as many other preferred regression variables reported in Table 2 of the study of Rogers and Hopfinger, are linearly correlated with biological activity ($|R| > 0.5$). However, many descriptors do not exhibit such correlation, for example, ESDL4, ATCH10, and NSDL3 ($|R| < 0.2$). Neither the GFA nor the EP linear regression studies incorporated these nonlinear features to their top models. Since these models appear to ignore nonlinear descriptors, it is doubtful that they made the best use of the data set. It is for this purpose that a neural network, with its ability to optimize nonlinear relationships, was introduced.

Genetic Neural Network Simulations. Two sets of evolutionary simulations with the GNN were performed. In these two runs, a genetic algorithm, either GFA or EP, was used as a feature-selecting tool, and a neural network was employed for the feature mapping. Both simulations, referred as GFA−NN and EP−NN, contained a population of 300 individuals which evolved for 100 generations. The fitness score for each individual was the reciprocal of the RmsE of the neural network training set (eq 3a). Figure 5 shows a graph of the best and the average fitness at different generation. In the EP−NN run, both the best and the average scores seemed to converge after 10 generations. The GFA−NN simulation also reached the same best fitness score around the same time, but its average fitness remains essentially at its initial value. This behavior results from the fact that the simple elitist GFA−NN algorithm only retains the best model and continues to create a fresh set of models randomly at every generation.

These simulations demonstrated some significant differences between the two genetic algorithms. The elitist GFA−NN algorithm and the EP−NN algorithm discovered the same best model at approximately the same evolution time (Figure 5). However, the remain-

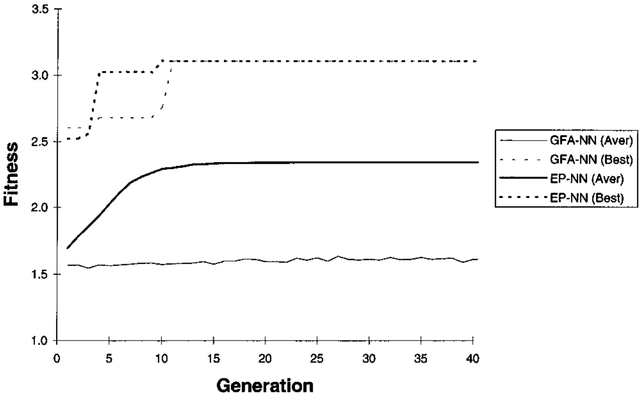


Figure 5. The average and the best fitness of the two genetic neural network simulations as a function of generation. The GFA and EP systems reach the same best fitness around the 11th generation. The behavior of the two average fitness curves are markedly different. In these two GNN runs, the fitness score is simply the reciprocal of the RmsE of the training data.

ing GFA−NN models were significantly less fit than those which were derived from EP−NN. An examination revealed that the other high-ranking EP−NN models had been discovered by GFA−NN, but these models were destroyed by crossover or mutation during reproduction and did not survive in a long run. Thus, the GFA−NN result at the 100th generation represented the best model that had been discovered up to that point, plus a group of mediocre models. Either of two modifications on the GFA−NN algorithm can lead to a set of models that is identical to that found by the EP−NN simulation. First, we could extend elitism by requiring the system to keep not only the best individual, but also some other high-ranking individuals. Second, we could adopt a scheme which gradually decreased the probability of mutation, the rate of crossover, or both. This would mean that an extra parameter had to be introduced.[22] Consequently, we used only the EP−NN hybrid system in subsequent studies.

The top six models from each of the two runs are shown in Table 4. Consistent with the previous studies, the presence of LOGP in all 12 models provided further evidence that it was an essential descriptor in this data set. Other descriptors found in some of regression studies are MOFI_Y, MOFI_Z, M_PNT, and ATCH4. However, the linear descriptor ESDL3 does not appear and the nonlinear NSDL3 appears in the neural network models. There is strong evidence that this latter descriptor cannot be described by a linear term. Attempt to construct regression QSAR with a set of descriptors that works well with a neural network is often futile, and it is particularly true if the set contains a nonlinear input.[23] This is the case for these new sets of descriptors. Both correlation coefficients and cross-validation results are unsatisfactory when determined by multiple linear regression (Table 4).

The best GNN models are better than any of the published models. Not only is the fit of the training data superior but also more predictive. The best model, which is common to both genetic neural network simulations, is particularly impressive. Its cross-validated correlation coefficient (0.866) is higher than the fitted correlation coefficient (0.849) of the best linear regression model.

The best neural network model was examined in more detail. A functional dependence analysis of individual descriptors was performed with a neural network monitoring scheme,[18] and the results are shown in Figure 6. The two descriptors LOGP and MOFI_Y were still approximately linear, although there was a reduction in linearity from that in the regression model (compare Figure 4 and Figure 6). This difference is due to the coupling among the different descriptors. For NSDL3 it is very clear that it cannot be satisfactorily modeled with a linear term; the fit shown in Figure 6 is very poor ($R^2 = 0.12$). Biological activity plots of this type can be used as an aid in drug design. In the present case biological activity reaches a maximum value when NSDL3 = 1.70 and MOFI_Y = 14500 and seems to be insensitive to LOGP once the value of this descriptor reaches 7.1. It would be interesting to design new analogues which have these descriptors optimized at the corresponding critical values and to determine whether this led to improved potency. The neural network has predicted that such compounds, if they can be made, will have biological activities exceeding the highest value in the present data set.

Completeness and Efficiency of Genetic Algorithm. Although a genetic algorithm investigates many possible solutions simultaneously, there is no guarantee that the best solution can always be found. An exhaustive search is the only method which guarantees such solution, but this is often impossible in QSAR, as well as many other problems where genetic algorithms are used. Fortunately, the use of relatively few descriptor inputs in the current study permitted us to perform an exhaustive enumeration of all possible combinations of three-descriptor models.

A complete solution of the current problem requires ranking of all 23 426[37] three-descriptors models based on their fitness scores. This exhaustive enumeration was done, and Table 5 shows the results of the ranking of all the models listed in Tables 3 and 4 with their fitness scores. At the end of the EP−NN simulation all of the 50 best models found in the exhaustive search were discovered, and only one (the 95th) model out of the first 100, and 4 out of top 150, were not found. For practical purposes it is unlikely that researchers would be able to consider more than a reasonable tier (perhaps 50 or so) of QSAR models. The fact that the EP−NN algorithm has discovered the top tier models makes this set of solutions complete in a practical sense.

The efficiency in finding good solutions is a known strength of genetic algorithms. This is clearly evident in the present results. The optimal solution that was revealed by the exhaustive search was discovered as early as the 10th generation by the current EP−NN system: the top 10 models at the 14th and the top 50 models at the 27th. Computationally this is much less expensive than the exhaustive method that required calcultion of 23 426 neural network models. On the EP−NN run, only 3300 (300 × 11, taking account of the zeroth generation pool) such calculations with a neural network would be needed for 300 individuals evolving over 10 new generations.

Simulation with an Alternative Fitness Function. The genetic neural network simulation described so far employed a fitness function that was inversely proportional to the residual error of the training set.

Table 4. Two Sets of Top Models Deriving from Genetic Neural Network Simulations[a]

model	descriptors			neural network			regression	
				RmsE	*R*	XR	*R*	XR
EP−NN1	NSDL3	MOFI_Y	LOGP	0.322	0.919	0.866	0.777	0.284
EP−NN2	ATCH4	ATCH7	LOGP	0.330	0.914	0.755	0.660	0.494
EP−NN3	NSDL3	MOFI_Z	LOGP	0.338	0.910	0.830	0.782	0.296
EP−NN4	LOGP	M_PNT	SUM_F	0.348	0.905	0.762	0.779	0.685
EP−NN5	NSDL3	VDW_V	LOGP	0.349	0.916	0.753	0.691	0.276
EP−NN6	NSDL8	MOFI_Y	LOGP	0.349	0.907	0.860	0.786	0.470
GFA−NN1	NSDL3	MOFI_Y	LOGP	0.322	0.919	0.866	0.777	0.284
GFA−NN2	NSDL9	VDW_V	LOGP	0.353	0.901	0.814	0.696	0.329
GFA−NN3	ATCH5	NSDL9	LOGP	0.368	0.892	0.726	0.698	0.372
GFA−NN4	NSDL10	MOFI_Y	LOGP	0.383	0.883	0.789	0.786	0.598
GFA−NN5	ATCH3	LOGP	M_PNT	0.415	0.862	0.768	0.755	0.637
GFA−NN6	ATCH3	S8_1DY	LOGP	0.424	0.855	0.688	0.740	0.634

[a] The models are ranked by their residual rms error (RmsE) in training. Also reported are the correlation coefficients (*R*) and the cross-validated correlation coefficient (XR) of the neural network models and their regression counterparts.

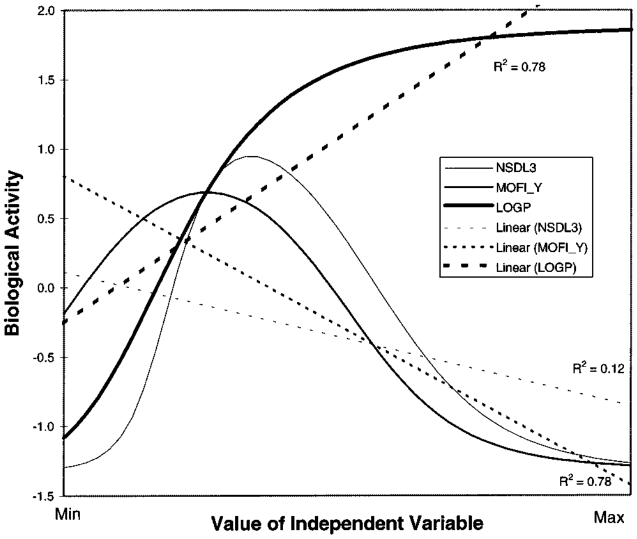


Figure 6. Biological activity as a function of the individual physicochemical parameter for the best genetic neural network model. The best fit straight line (dotted lines) and their r^2 values are also shown.

Table 5. Ranking of All the Models Listed in Tables 3 and 4 Based on an Exhaustive Enumeration of All Three-Descriptors Neural Network Models (a Total of 23 426)

model	fitness score	rank
Selwood	2.052	532
Wikel	2.189	268
GFA1	2.478	62
GFA2	2.502	54
GFA3	2.257	187
EP1	2.478	62
EP2	2.502	54
EP3	2.446	71
EP4	2.257	187
EP5	2.203	248
EP−NN1	3.109	1
EP−NN2	3.027	2
EP−NN3	2.962	3
EP−NN4	2.876	4
EP−NN5	2.864	5
EP−NN6	2.862	6
GFA−NN1	3.109	1
GFA−NN2	2.830	7
GFA−NN3	2.719	13
GFA−NN4	2.614	23
GFA−NN5	2.410	82
GFA−NN6	2.361	108

One apparent weakness in this fitness function is that it yielded models which were optimal for the training data but it did not always lead to good predictions for test data. To address this question an alternative fitness function was tried with a genetic neural network simulation (EP−NN/T). Three representative data points belonging to the low-, medium-, and high-activity categories were removed from the training set, and they were treated as the test set. The reason why we removed so few data points was that we wanted to keep the configuration for the neural network at 3−3−1. Removal of too many compounds from the training set would either lead to a substantial drop in the ρ ratio or force a change in the network configuration.

We used a fitness function that was based on the overall rms error of the test set. An additional criterion was imposed so that a model would be invalidated if the residual rms error of the training set did not fall below an acceptable threshold value of 0.39. This measure was introduced to prevent the inclusion of models which gave fortuitous test set predictions without adequate background training. The results are shown in Table 6. The predictions made for the three test compounds are not included in the calculation of XR because they were used in model optimization. Three of these top seven models were obtained in our first EP−NN study (Table 4); the same best model was found. However, in contrast to the first study, all other high-ranking models had high values of cross-validated correlation coefficients. All of the top five EP−NN/T models are not only effective in data fitting; they also display a high predictive power that is superior to the best genetic regression model. It is evident that the new fitness definition favors the propagation of the models which are more predictive. Since the aim of QSAR is to make accurate predictions, introductions of such a test set in the optimization may be of general utility.

Jackknife Validation in Optimization: The Ultimate Method. The final GNN simulation of this study was to obtain a list of three-descriptor models which were optimized for their predictive capacity. There were two objectives in this study. First, because the real goal of QSAR studies is to formulate models which make accurate predictions, it is important that such models can be found. Second, this would serve as a validation of our preceding work, particularly the results of the EP−NN/T simulation. To achieve this, we needed to run a GNN simulation with a fitness function that was related to the predictiveness. The cross-validated correlation coefficient was a suitable candidate. In this new simulation (codename EP−NN/

Table 6. QSAR Obtained by the EP−NN/T Simulation Using a Fitness Function That Is Inversely Proportional to Residual rms Error of the Test Set[a]

model	descriptors			test set RmsE	train set RmsE	neural net model	
						R	XR
EP−NN/T1	NSDL3	MOFI_Y	LOGP	0.305	0.337	0.919	0.875
EP−NN/T2	NSDL3	MOFI_Z	LOGP	0.404	0.349	0.910	0.850
EP−NN/T3	MOFI_X	LOGP	SUM_F	0.449	0.368	0.893	0.837
EP−NN/T4	NSDL8	MOFI_Y	LOGP	0.452	0.324	0.907	0.872
EP−NN/T5	NSDL8	MOFI_Z	LOGP	0.520	0.334	0.888	0.866
EP−NN/T6	NSDL10	LOGP	SUM_F	0.542	0.346	0.880	0.782
EP−NN/T7	NSDL9	MOFI_Y	LOGP	0.580	0.344	0.892	0.863

[a] The correlation coefficients (*R*) and the cross-validated correlation coefficient (XR) of the neural network models are reported in the last column. As a measure of data validation, the predictions made for the three test set compounds are discarded in the XR calculation because of their role in the model construction.

Table 7. Result from the Final Jackknife Validation (EP−NN/X) Run[a]

model	descriptors			model XR
EP−NN/X1	NSDL3	MOFI_Y	LOGP	0.866
EP−NN/X2	NSDL8	MOFI_Y	LOGP	0.860
EP−NN/X3	NSDL8	MOFI_Z	LOGP	0.850
EP−NN/X4	NSDL9	MOFI_Y	LOGP	0.843
EP−NN/X5	NSDL3	MOFI_Z	LOGP	0.830
EP−NN/X6	MOFI_X	LOGP	SUM_F	0.827
EP−NN/X7	NSDL8	VDW_V	LOGP	0.823
EP−NN/X8	NSDL8	SURF_A	LOGP	0.822

[a] The top six models have previously been identified in the EP−NN/T simulation.

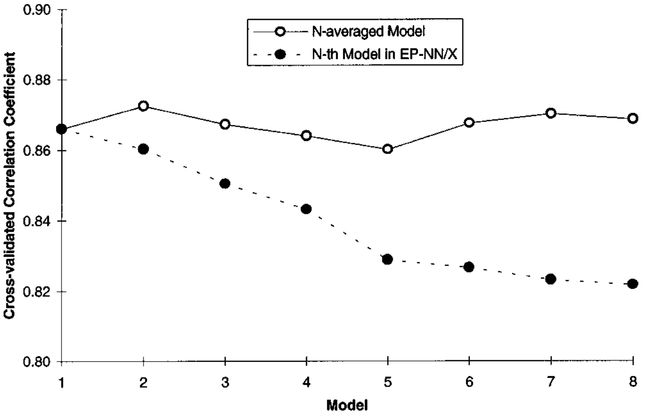


Figure 7. Diagram showing the cross-validated correlation coefficient plotted against the *N*-averaged composite model (○) and the *N*th best model (●) in EP−NN/X simulation.

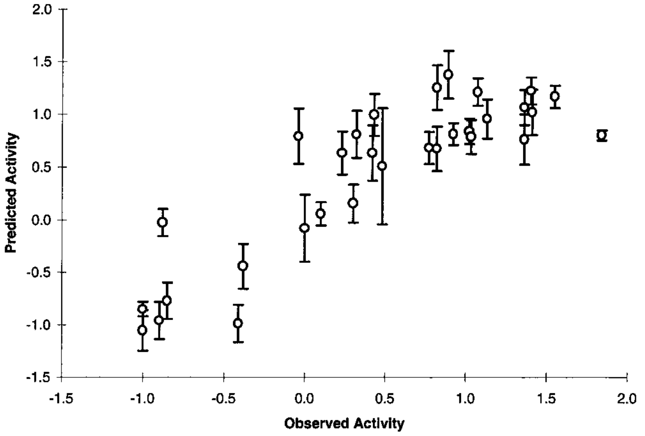


Figure 8. Plot of predicted activity versus observed activity. The predictions are made by averaging the results of the eight models listed in Table 7; the error bar corresponds to one standard deviation from the averaged value.

X), a population of 300 individuals were again used. The fitness of a given model was directly proportional to its cross-validated correlation coefficient (eq 3b).

The models derived from this run are listed in the descending order with their cross-validation scores (Table 7). Six of the seven high scoring models discovered in the EP−NN/T simulation represent the six best models in this limiting case. This is an important validation of the EP−NN/T run because a jackknife analysis of such complexity requires large computing resources. The current simulation, which used over 150 CPU hours on a fast machine, was only feasible because of the small data set and the simplicity of the model. The results indicate that a much less expansive alternative (EP−NN/T) leads to solutions of similar predictive quality.

Construction of a Grand Model. The GFA study of Rogers *et al.*[21] reported that by averaging the outputs of the regression equations of several good models the correlation coefficients of the training set would increase. This is an interesting observation, but it is important to know if the cross-validated correlation coefficients is also improved by the same treatment. The predicted output values from several high-scoring EP−NN/X models discovered in the last study were averaged, and the behavior of the new cross-validated correlation coefficients was investigated.

Two plots are shown in Figure 7. The first series of data, denoted by the filled circles, represents the cross-validated correlation coefficient of the *N*th best model in the last study. The second series of data, denoted by the open circles, shows the cross-validated correlation coefficient of the composite model obtained by averaging a number of models. On averaging the best and the second-rated model, the correlation coefficient increased from 0.866 to 0.872. Further averaging of up to eight models showed that the correlation coefficients fluctuate but are confined within a narrow range from 0.86 to 0.87, regardless of the predictive quality of the added model.

In the present simultaneous averaging of the predictions of a few higher performance models leads to a very small gain in predictiveness; it is not clear that the gain is statistically significant. However, we believe that predictions made by this approach are better. First, the scope of predicted values given by different models permits an error estimation to be made for the prediction. Figure 8 shows the average predicted activities against observed values with error bars set to one standard deviation of the predicted ranges using the eight models listed in Table 7. Furthermore, this type of averaging provides an attractive means to make greater use of information in the data set without the possibility of overfitting.

Choice of Descriptors. Receptor−drug binding affinity and drug transport are two of the important attributes of a drug. For many years medicinal chemists have attempted to model these two properties with empirical physicochemical parameters. It is recognized that in receptor−drug binding the dominant factors are the steric and electrostatics interactions between the two molecules. It is for this purpose that many bulk and electronic properties have been introduced into the parameterization of drug candidates. Drug transport through membrane, by contrast, is most successfully modeled by a hydrophobicity parameter. The parameter of choice is the partition coefficient between octanol and water because of the ease in performing either direct measurements or numerical computations.

The choice of descriptors of the highly predictive models (Table 7) discovered in this study were examined, and some interesting similarities and differences were observed. It was noted that all models contained a descriptor in each of the bulk, electronic, and hydrophobic categories. Since LOGP was the sole hydrophobicity parameter in the descriptor set, it was always included. The steric bulk of the drug candidates was represented either by a directional measure of moments of inertia for a specific principal axis or by a global measure in terms of van der Waals volume or surface area. The electronic information was provided by the values of NSDL at atom 3, where there is considerable structural variations (Table 1), or at the carbonyl group. The latter might involve its ability to form hydrogen bonds, or more speculatively, its role as a site of nucleophilic attack involved in drug metabolism. These results show that the choices of descriptors in our models makes good chemical sense. Furthermore, it suggests that for this particular data set, a small number of descriptors is sufficient to construct a good predictive QSAR, provided the crucial information on steric, electrostatics, and hydrophobic properties is adequately described.

IV. Concluding Discussion

A novel QSAR tool, GNN, that combines a genetic algorithm with a neural network is applied to the Selwood data set. GNN models with fitting and prediction ability that exceed all published models are obtained. The effectiveness of the evolutionary programming algorithm is demonstrated by selection of the best sets of descriptors. The selection is shown to be optimal in terms of both completeness and efficiency when the result is compared with an exhaustive enumeration benchmark. The key strength of a neural network is its ability to allow for flexible mapping of the selected features by manipulating their functional dependence implicitly. Unlike regression analysis, neural network handles both linear and nonlinear relationships without adding complexity to the model. This capability offsets the larger computing time required by a neural network simulation because it avoids the need of examining separately each possible nonlinearity.[19] Furthermore, a neural network permits the evaluation of the functional dependence of the descriptors, which can be an aid in future drug design.

Another important result from this study is that a simple partitioning of data into training and test sets led to a result mimicking that of genuine jackknifing. Since our choice of test set is somewhat arbitrary, it is possible that result is not of general validity. A jackknife procedure remains the ultimate validation method. When it becomes computationally intractable, multiple cross-validation (or “leave-one-out”), in which a series of random partitionings of training and test data are used, can serve as an alternative strategy.[38]

An advantage of a genetic algorithm is its ability to generate multiple predictors that are of comparable quality. An attempt is made in this study to combine these multiple QSARs. A composite model is constructed by merging the predictions obtained from several high scoring GNN models. In the present case this approach does not lead to any improvement that is of statistical significance, probably because the best three-descriptor GNN models are already highly predictive. The choice of descriptor sets for the QSAR is examined, and it is shown that they include the steric, electrostatic, and hydrophobic attributes of each molecule. The result suggests that a small number of chemically meaningful descriptors will provide the most predictive QSAR.

Acknowledgment. We thank David Rogers for supplying the Selwood data set. We are grateful to Steven Li for his advice in the construction of the genetic neural network simulator. This work is supported in part by a grant from the National Institutes of Health. The calculations were done on Hewlett-Packard 735/125 workstations.

References

(1) Hansch, C.; Fujita, T. ρ-σ-π analysis. A method for the correlation of biological activity and chemical structure. *J. Am. Chem. Soc.* **1964**, *86*, 1616−26.

(2) Kubinyi, H. *QSAR: Hansch analysis and related approaches*: VCH: Weinheim, 1993; Methods and principles in medicinal chemistry, Vol. 1.

(3) van de Waterbeemd, H. *Chemometric methods in Molecular Design*; VCH: Weinheim, 1995; Methods and principles in medicinal chemistry, Vol. 2.

(4) van de Waterbeemd, H. *Advanced computer-assisted techniques in drug discovery*; VCH: Weinheim, 1995; Methods and principles in medicinal chemistry, Vol. 3.

(5) Cash, G. G.; Breen, J. J. Correlation of graph-theoretical parameters with biological activity. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 275−9.

(6) Yao, Y.-Y.; Xu, L.; Yang, Y.-Q.; Yuan, X.-S. Study on structure-activity relationships of organic compounds: three new topological indices and their application. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 590−4.

(7) Dang, P.; Madan, A. K. Structure-activity study on anticonvulsant (thio) hydantoins using molecular connectivity indices. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1162−6.

(8) Good, A. C.; So, S.-S.; Richards, W. G. Structure-activity relationships from molecular similarity matrices. *J. Med. Chem.* **1993**, *36*, 433−8.

(9) Good, A. C.; Peterson, S. J.; Richards, W. G. QSAR's from similarity matrices. Technique validation and application in the comparison of different similarity evaluation methods. *J. Med. Chem.* **1993**, *36*, 2929−37.

(10) Seri Levy, A.; West, S.; Richards, W. G. Molecular similarity, quantitative chirality, and QSAR for chiral drugs. *J. Med. Chem.* **1994**, *37*, 1727−32.

(11) Benigni, R.; Cotta Ramusino, M.; Giorgi, F.; Gallo, G. Molecular similarity matrices and quantitative structure-activity relationships: a case study with methodological implications. *J. Med. Chem.* **1995**, *38*, 629−35.

(12) Cramer, R. D. I.; Patterson, D. E.; Bunce, J. D. Comparative Molecular Field Analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959−67.

(13) Camilleri, P.; Livingstone, D. J.; Murphy, J. A.; Manallack, D. T. Chiral chromatography and multivariate quantitative structure-property relationships of benzimidazole sulphoxides. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 61−9.

(14) Hudson, B.; Livingstone, D. J.; Rahr, E. Pattern recognition display methods for the analysis of computed molecular properties. *J. Comput. Aided Mol. Des.* **1989**, *3*, 55−65.

(15) Domine, D.; Devillers, J.; Chastrette, M. A nonlinear map of substituent constants for selecting test series and deriving structure-activity relationships. 2. Aliphatic series. *J. Med. Chem.* **1994**, *37*, 981−7.

(16) Norinder, U. A PLS QSAR analysis using 3D generated aromatic descriptors of principal property type: Application of some dopamine D2 benzamide antagonists. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 671−82.

(17) Andrea, T. A.; Kalayeh, H. Applications of neural networks in quantitative structure-activity relationships of dihydrofolate reductase inhibitors. *J. Med. Chem.* **1991**, *34*, 2824−36.

(18) So, S.-S.; Richards, W. G. Application of neural networks: quantitative structure-activity relationships of the derivatives of 2,4-diamino-5-(substituted-benzyl)pyrimidines as DHFR inhibitors. *J. Med. Chem.* **1992**, *35*, 3201−7.

(19) Ajay. A unified framework for using neural networks to build QSARs. *J. Med. Chem.* **1993**, *36*, 3565−71.

(20) Livingstone, D. J.; Hesketh, G.; Clayworth, D. Novel method for the display of multivariate data using neural networks. *J. Mol. Graphics* **1991**, *9*, 115−8.

(21) Rogers, D. R.; Hopfinger, A. J. Application of genetic function approximation to quantitative structure-activity relationships and quantitative structure-property relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854−66.

(22) Luke, B. T. Evolutionary programming applied to the development of quantitative structure-activity relationships and quantitative structure-property relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1279−87.

(23) Sutter, J. M.; Dixon, S. L.; Jurs, P. C. Automated descriptor selection for quantitative structure-activity relationships using generalized simulated annealing. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 77−84.

(24) Selwood, D. L.; Livingstone, D. J.; Comley, J. C.; O'Dowd, B. A.; Hudson, A. T.; Jackson, P.; Jandu, K. S.; Rose, V. S.; Stables, J. N. Structure-activity relationships of antifilarial antimycin analogues: a multivariate pattern recognition study. *J. Med. Chem.* **1990**, *33*, 136−42.

(25) Chem-X; Chemical Design Ltd: Oxford, U.K.

(26) Cerius-2; Molecular Simulations Inc: Burlington, MA.

(27) Manallack, D. T.; Ellis, D. D.; Livingstone, D. J. Analysis of linear and nonlinear QSAR data using neural networks. *J. Med. Chem* **1994**, *37*, 3758−67.

(28) Brook, R. J.; Arnold, G. C. *Applied regression analysis and experimental design*; Marcel Dekker, Inc.: New York, 1985; Statistics: textbook and monographs.

(29) Maggiora, G. M.; Elrod, D. W. Computational neural networks as model-free mapping devices. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 732−41.

(30) Blank, T. B.; Brown, S. D. Nonlinear multivariate mapping of chemical data using feed-forward neural networks. *Anal. Chem.* **1993**, *65*, 3081−9.

(31) Holland, J. H. *Adaption in natural and artificial systems*; The University of Michigan Press: Ann Arbor, MI, 1975.

(32) Cartwright, H. M. *Applications of artificial intelligence in chemistry*; Oxford University Press: Oxford, 1993.

(33) Zupan, J.; Gasteiger, J. *Neural networks for chemists: An introduction*; VCH Publishers: New York, 1993.

(34) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning internal representations by error propagation*; MIT Press: Cambridge, MA, 1986; Parallel distributed processing, Vol. 1.

(35) Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction to the theory of neural computation*; Addison-Wesley Publishing Co.: Redwood City, CA, 1991.

(36) Wikel, J. H.; Dow, E. R. The use of neural networks for variable selection in QSAR. *Bioorg. Med. Chem. Lett.* **1993**, *3*, 645−51.

(37) There are 53 descriptors in the data set. Since the order of descriptors in the model is not significant, the total number of distinct combinations of three-descriptor models is 53!/(50! × 3!) = 23 426.

(38) Chandonia, J.-M.; Karplus, M. Neural networks for secondary structure and structural class predictions. *Protein Sci.* **1995**, *4*, 275−85.

JM9507035