Prediction of Material Properties from Chemical Structures. The Clearing Temperature of Nematic Liquid Crystals Derived from Their Chemical Structures by Artificial Neural Networks

Helge Kränz, Volkmar Vill, and Bernd Meyer*,†

Institute for Organic Chemistry, University of Hamburg, Martin-Luther-King-Pl. 6, 20146 Hamburg, Germany

Received August 22, 1996[⊗]

The prediction of properties of molecules just on the basis of their chemical structures is desirable to selectively make molecules that have the wanted properties, like biological activity, viscosity, or toxicity. Here, we present an example of a new way to predict a property from the chemical structure of a chemically heterogeneous class of compounds. The clearing temperatures of nematic liquid−crystalline phases of 17 383 compounds were used to train neural networks to derive this material property directly from their chemical structure. The trained neural networks were subsequently tested with 4345 structural patterns of molecules unknown to the networks to assess their predictive value. The clearing temperatures were predicted by the best network with a standard deviation of 13°.

INTRODUCTION

The prediction of material properties of chemical compounds has been achieved on the basis of empirical relations by associating parameters derived from the structure of the molecule with biological activity (SAR and QSAR). Recently, neural networks have been used in chemistry[1] to extract information from complex patterns, like NMR spectra[2−5] or mass spectra.[6] An encoding scheme was developed to predict phosphorus NMR shifts[7] using artificial neural networks. A graph representation of alkanes from C_6 to C_{10} was used to train a neural network to predict six thermodynamic parameters.[8] Using descriptors accessible through molecular modeling of each compound a neural network was shown to predict boiling points and critical temperatures.[9] Using simple molecular structural considerations a neural network was devised to predict the biological activity of chemical compounds.[10] Counterpropagation neural networks were designed to model and predict activities of carboquinones and of benzodiazepines based on physicochemical parameters.[11] A combination of a genetic algorithm and a neural network was developed for quantitative structure−activity relationship.[12] We have shown that artificial neural networks can be used to predict the transition temperatures of smectic liquid crystals.[13] Here, we demonstrate the use of artificial neural networks to convert the information of chemical structures in a large data set directly into physical properties.

Normally, a feed forward-back propagation network[2,3,14] consists of two layers of neurons, one hidden and the output layer plus one input layer. The neurons are connected between the layers but not within a layer. This type of neural networks can easily handle one-dimensional information. However, the chemical structure that will be presented to the input layer generally has a higher dimensionality such that it cannot easily be mapped to the input layer. The encoding of a chemical structure into the input layer must be unambiguous in two ways: first, two different structures must not have the same representation, and, second, each structure must have exactly one representation. Furthermore, for a feed forward-back propagation neural network the encoding of the chemical structures has to be translationally invariant. That is, if an encoding of chemical structures were not translationally invariant, the neural network would have to learn that structural elements could occur at any position of the input layer of the neural network. This effectively would require that the neural network must recognize a given structural element at any position in the encoding pattern. This task, if doable at all, would be very time consuming during the training phase.

METHODS

The physical data of more than 50% of all known nematic liquid crystalline compounds were extracted from a database that holds data of almost all compounds that form liquid−crystalline phases.[15] First, the structures of the liquid−crystalline compounds were formally split up into nine groups that represent the terminal groups (T_1, T_2), the linking groups (L_1, L_2), the bridging groups (B_1, B_2), and the rings (R_1, R_2, R_3) (Figure 1). Molecules have consequently a formal structure of T_1-L_1-R_1-B_1-R_2-B_2-R_3-L_2-T_2. Each group has a corresponding vector, which contains the information about the group. The number of different fragments for each group is represented by an equal number of input neurons. Many structures do not contain a representative from each group, and, thus, the encoding for a nematic liquid−crystalline structure results in a maximum of nine activated input neurons (cf. Figure 1 and Table 1). The number of actually activated input neurons in our data set is between 3 and 9. Table 1 shows the number of different elements (fragments) in each group, their functional descriptor, and a short name. The nine vectors are concatenated starting with the vector corresponding to T_1 and ending with the vector corresponding to T_2 to give a vector of 205 elements total that is mapped to an equal number of input neurons. The whole data set was split in an arbitrary ratio of 4:1 of which the bigger set of 17 383 patterns was used for training and the smaller of 4345 was used to test the network. This ratio was chosen to have most molecules in the training data set and still

† FAX: +49 40 4123 2878. Telephone: +49 40 4123 5913. E-mail: bmeyer@chemie.uni-hamburg.de.

⊗ Abstract published in *Advance ACS Abstracts,* November 1, 1996.

S0095-2338(96)00482-9 CCC: $12.00

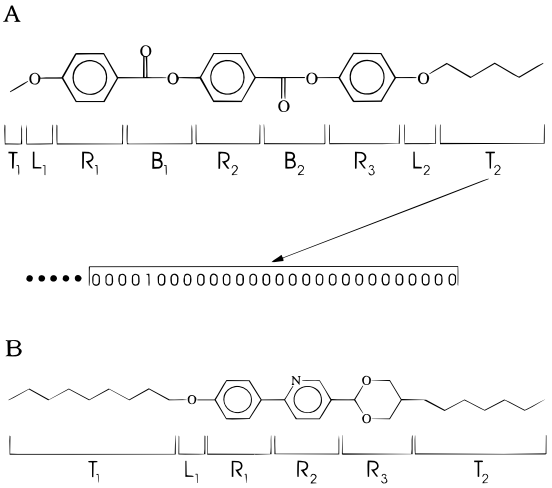

Figure 1. Two examples of liquid−crystalline compounds are shown. Panel A shows an example for a liquid−crystalline compound with representatives of all groups present such that nine input neurons are set to the "on" state. The group names are shown below the structure. As an example, the pentyl substructure, which forms the terminal group T_2, is the fifth of 28 possible substructures in this group such that 28 input neurons represent the group T_2 and the fifth of these is set to "on", while the others are "off". This set of input neurons forms the rightmost part of the input vector (cf. Table 1). The compound shown in panel B has no representatives in groups B_1, B_2, and L_2, which corresponds to six input neurons turned "on" in the groups T_1, L_1, R_1, R_2, R_3, and T_2, respectively. Because the structural encoding used here has no unambiguous direction, each asymmetrical pattern was represented to the neural network in two ways beginning at either end such that the groups T_1-L_1-R_1-B_1-R_2-B_2-R_3-L_2-T_2 were presented to the input neurons once left to right and the next time right to left. The representation of an asymmetrical compound differs from its reverse representation such that the network has to learn both representations to be able to predict the clearing temperature independent of the direction of writing for new structures. Only ≈5% of the molecules in the database were symmetrical.

Table 1. Chemical Structures Encoded into a Vector That Maps the Structures to the Input Neurons of the Neural Networks[a]

group	name	no. of different substructures represented in data set
first terminal	T_1	28
first linking	L_1	6
first ring	R_1	33
first bridge	B_1	19
second ring	R_2	33
second bridge	B_2	19
third ring	R_3	33
second linking	L_2	6
second terminal	T_2	28

[a] For this, the molecules were fragmented into nine different groups (left column), whose short name is also given (middle column). The number of different representatives in each group is shown in the right column. In total, there are 205 possible elements of which each molecule can only contain up to nine (one of each group).

enough members in the test data set to allow a statistical treatment. All molecules that are composed of groups that were each represented at least five times in the database were selected for the neural network analysis representing about 50% of all nematic liquid crystalline compounds. Five representatives for each group were arbitrarily chosen to ensure a high probability that the neural network could identify the properties of a given functional group from at least several compounds. It is not meaningful to have the neural network make decisions based on only one occurrence of a functional group in the training set. The encoding scheme used here can easily be applied to other classes of chemical compounds if they can be separated in several groups. Then, each group can be encoded as shown here what leads to an encoding that can be used as an input for feed forward-back propagation neural networks. Examples for other classes of encodable molecules are peptides and prostaglandins.

We used two variants to map the clearing temperature to the vector of output neurons. First, the clearing temperature was mapped to many neurons to resemble a thermometer, and, second, the clearing temperature was mapped to only one neuron, which was interpreted as an analog device. The interval of the clearing temperatures reaches from the minimum (T_1) to the maximum (T_2) temperature of all compounds used for training and testing. In the first case, the resulting vector of output neurons has T_1-T_2 elements that map to one degree each. The first element represents the temperature T_1 and every subsequent element in the "on" state raises the temperature by one degree. If the clearing temperature is T the first $T-T_1$ elements are set to "on" while the remaining neurons are "off". The temperature range of our neural network reached from −45 °C to 356 °C. To avoid the mapping of the extreme temperatures of −45 °C corresponding to no activated neuron and 356 °C corresponding to all output neurons activated ten more output neurons were added at each side representing a temperature range from −55 °C to 366 °C. Therefore, this temperature mapping results in 421 output neurons. In the second method, the range of the clearing temperature is mapped to one neuron that can only represent values between 0 and 1. A value of 1 in this neuron represents a temperature of 356 °C and a value of 0 a temperature of −45 °C.

The learning rate was initially set to 2.0 and was subsequently decremented every 500 cycles (a cycle is the presentation of all patterns of the training set including the weight updates to the network) to reach a minimum of 0.01 at the end of the training. Learning was measured by the decline of the root mean square (RMS) of the output neurons of all patterns of the training set:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{j=1}^{n_{\mathrm{pat}}}\sum_{i=1}^{n_{\mathrm{out}}}(x_{ij}-y_{ij})^2}{n_{\mathrm{pat}}\cdot n_{\mathrm{out}}}}$$

x_{ij} denotes the calculated value of output neuron i of pattern j, y_{ij} the corresponding expected value, and n_{pat} and n_{out} are the number of patterns in the training data set and the number of output neurons, respectively. The training was terminated when no further convergence could be achieved (cf. Figure 2). At this point the learning rate was 0.01.

The number of neurons in the input layer was set to 205, the number of hidden neurons was set to 100, and the number of neurons in the output layer to 421 or 1, respectively. Two different networks have been trained: network A, which had 421 neurons on the output layer, and network B, which had one neuron on the output layer. Network A was trained for 20 000 cycles requiring 500 h of CPU time on a six processor Silicon Graphics Power Challenge (SC900), and network B was trained for 50 000 cycles requiring 350 h of CPU time,

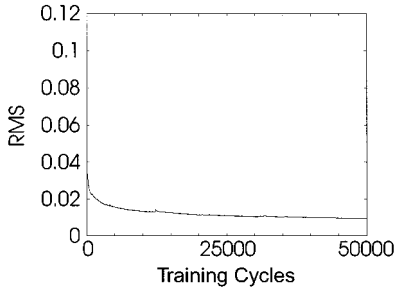

Figure 2. The decrease of the RMS with the number of training cycles is shown for the training of network B. The learning rate was decreased every 500 cycles (see text). When no further significant reduction of the RMS could be obtained, the training was terminated (after 50 000 cycles). The RMS values obtained during training of network A produced a similar curve.

respectively. This is equivalent to up to 14×10^6 weight updates per second and processor.

RESULTS AND DISCUSSION

We tested several variations of the network architecture to determine that one suited best for this task of correlating chemical structure with clearing temperatures. First we optimized the layout of the output layer. We encoded the chemical structures as described above and chose 100 hidden neurons and encoded the clearing temperature on the output layer in one neuron. The clearing temperature was mapped onto the full range of the neuron from 0.0 to 1.0. This type of network converged (network B) and gave good results (cf. below). Then we used a different mapping of the range of clearing temperatures to neuron values of between 0.3 and 0.7 to use the more linear parts of the sigmoid functions. The results of this network were much poorer than those obtained using the first network. This behavior can easily be explained by the instability of the neural network for neuron values at about 0.5. The feed forward-back propagation algorithm stabilizes neurons with values of about 0.0 and 1.0 but destabilizes those with intermediate values. When trying to map the range of the clearing temperatures to a neuron range between 0.95 and 0.9995 we obtained again much worse results compared to the network where we had used the full range of the neurons.

Alternatively, we tried to develop a completely different way of mapping the clearing temperature to the output layer. Here we tried to map the clearing temperature to many output neurons to simulate a digital thermometer. Each of the output neurons represents a temperature step of one degree. The results of this net were about as good as those of the first net, so we decided to use both networks to obtain predictions because this network contains redundant information when a clearing temperature is not predicted correctly (cf. Figure 3). On the other hand, this net needs much more computation time than the others. To reduce this amount of time, we decided to test a third variant of encoding the clearing temperatures to the output layer. Now, we encoded the temperature bitwise, i.e., a temperature of 1 was mapped to "000000001", 2 to "000000010", 3 to "000000011", and so on. Nine output neurons were needed for this experiment. But the results after finishing the training were much poorer than all of the others. So we got two encodings of the clearing temperature of the same quality.

The next task was to figure out how the chemical structures could be mapped to the input layer efficiently. First we used

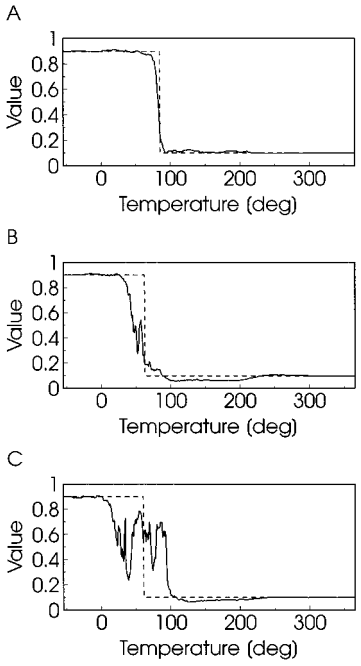

Figure 3. The three panels exemplify test results for network A showing in the top two panels the typical (93.2%) quality of the prediction of the clearing temperature. The solid lines indicate the calculated values of the output neurons and the dashed line the experimentally determined clearing temperature. The bottom panel represents the network responses with more than one transition over the thresholds of 0.3, 0.4, 0.5, 0.6, and 0.7 which occur in 6.8% of the cases. The two traces of the network responses shown at the top could be analyzed automatically in contrast to the trace shown at the bottom which was considered as "not recognized".

the encoding described in the methods section where we assigned one neuron each to the presence of a given structural fragment in the molecule. The terminal groups, for example, were represented by 28 input neurons equivalent to 28 different fragments in this part of the molecule. In total 205 input neurons were used to represent the chemical structures of the molecules that form nematic liquid−crystalline phases. To reduce the training effort, we also explored an alternative encoding scheme where each group is only represented by one input neuron, resulting totally in nine input neurons. Each neuron was fed with an integer number between 0 (fragment is absent, the group is empty) and the maximum number of fragments in this group where each integer was representing a given fragment within the group. The results of this training were very poor. The low quality of the testing results are obvious if one considers that two structures that were encoded by similar numbers could in fact be very different in their clearing point. This is also supported by the fact that neural networks are not very good in differentiating based on the relative intensity of the input patterns. Consequently, we decided to use the first encoding for the final networks. There are many other ways to encode chemical structures into a vector, but most of these encode each atom separately which would lead to an enormous number of input neurons: a liquid crystalline compound can hold easily 70 atoms each of which can be occupied by 10 different atom types leading to 700 input neurons. Considering different binding modes for each atom multiplies the complexity such that several thousand input neurons were necessary. Also, it is not easy to implement this encoding scheme translationally invariant. An alternative way to

encode structures with a few input neurons is to use a graph theoretical approach,[8] but different atom types cannot be handled easily by this approach. Other approaches exist in the literature that can be used to encode chemical structures into a neural network: first, an encoding of the chemical structure using parameters accessible by molecular modeling[9] and, second, an encoding based on data used in electron diffraction[16] that produces a fixed number of values.

Third we had to figure out how many neurons in the hidden layer produce the best results. We tested several networks with all parameters kept fixed except for the number of hidden neurons that were varied in the range between 5 and 100 hidden neurons. The best results were obtained for 100 hidden neurons. A neural network with 100 hidden neurons was the maximum size compatible with a reasonable amount of computing time.

At last, we tested several different values for the momentum and the learning rate. We discovered that varying the momentum in the range of 0.05−0.95 had almost no effect on the results, so we set it to a value of 0.5. The effect of the learning rate is much more significant. There are two effects that we observed with a constant learning rate: First, if the learning rate is set to a high value (e.g., 2.0) the network did not converge, and, second, if it is set to a small value (e.g., 0.05) it converges, but the convergence is very slow and it does not converge to a useful minimum. So we decided to vary the learning rate between a high value (2.0) and a low value (0.01). The learning rate was started at 2.0 for all networks and was reduced in equal steps after 200 cycles (network A) and 500 cycles (network B), respectively, to reach a final value of 0.01.

The connections between the layers and the biases of the neurons in the hidden and output layer were initialized randomly with numbers between −0.05 and 0.05. We tested several different initializations of the weights and biases, which, however, led to the same quality of the final results.

As a result of the optimization studies described above we use two networks that gave the best overall performance and whose properties are described in the following section. The first network has 205 input neurons, 100 hidden neurons, and 421 output neurons (network A), and the other network has the same input and hidden layer architecture but has only one output neuron (network B).

After the training, both networks were tested with the test set of 4345 structural patterns unknown to the network. In case of network B the clearing temperature was calculated directly from the value of the output neuron. In case of network A the calculation of the clearing temperatures is more difficult. Ideally, the output neurons of network A should result in a step function (cf. Figure 3). Because of the great number of test patterns an algorithm was devised to identify the point where the step, and thus the clearing temperature, is located. A neuron in its "on" state should have a value of 0.9 and in its "off" state 0.1. We accepted a tolerance of ±0.1 for each neuron. Consequently, a neuron was considered to be in the "off" state if its value was in the range of 0.0−0.2 and in its "on" state if its value was in the range of 0.8−1.0. As the next step the smoothness of the curves was assessed by counting the number of transitions over the threshold values of 0.7, 0.6, 0.5, 0.4, and 0.3, respectively. The values chosen are arbitrary values that spread evenly over the range were the step occurs in the step function. A transition is defined if the value of neuron

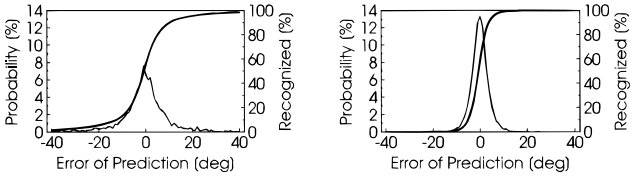

Figure 4. The two panels show histograms of the deviation of the predicted clearing temperature relative to the experimentally observed clearing temperature after the training of network B. Also shown are the integration curves that indicate how many structural patterns have been recognized so far. The left panel shows the distribution of the errors when the neural network was tested with 4345 patterns that were unknown to the network. The right panel shows the same distribution for the training set of 17 383 patterns. The distributions show clearly that a high proportion of the predicted clearing temperature is predicted in a narrow range of errors. Obviously, the knowledge base of the neural network represents the training set better than the test set as evidenced by the narrower distribution shown in the right panel.

(i) is greater than or equal to the threshold value and the value of following neuron ($i + 1$) is less than the threshold value. If only one transition occurs over each threshold, the step in the step function, and thus the clearing point, was assigned to the number of the neuron just before the transition over 0.5. These curves (about 85% of all patterns for network A) are generally smooth (cf. Figure 3A) and have a steep descent. After having identified all output pattern from the neural network that had simple transition modes we computed the RMS of all patterns recognized by this criteria:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}}$$

with x_i denoting the predicted clearing temperature, y_i denoting the experimentally measured clearing temperature, and N the number of identified patterns.

To identify the patterns that were not recognized in the first step (about 15%), we checked whether the difference between the first transition over 0.7 and the last over 0.3 was less than twice the standard deviation calculated in the first step. If a pattern matched this criterion the average between the first transition over 0.7 and the last over 0.3 was taken as the clearing temperature. Using this criterion, we additionally assigned a clearing temperature to 8.2% of all patterns. Thus, the recognition rate was in total 93.2%. An example for a pattern which is assigned by the second procedure is shown in Figure 3B. Finally, the RMS value was adjusted using all recognized patterns. The percentage of all patterns not recognized was 6.8%. One example of these patterns is shown in Figure 3C.

The error distribution (cf. Figure 4) of the difference between predicted and experimental clearing temperature for network B results in a curve similar to a normal distribution. It is obvious that the error distribution for the training data set is narrower than that one for the test data set. Throughout the training of the neural network the distribution of the training set was narrower than that of the testing data set.

The clearing temperatures of the patterns unknown to the networks were predicted with a standard deviation of 13° for network A and 16.4° for network B. The quality of the training as measured by the standard deviation of the test

Table 2. Results of Testing Both Networks with 4345 Unknown Molecules[a]

max. error	±5%	±10°	±15°	±20°	±50°	±100°
network A (relative to all patterns presented)	48%	68%	80%	85%	92%	93%
network A (relative to all patterns recognized)	51%	72%	85%	91%	99%	99.97%
network B	54%	77%	85%	90%	95%	99.54%

[a] The errors are grouped histogram-like. The main differences between the networks can be found in the region of smaller deviations. For small deviations the results of network B are much better than those obtained with the other network. Network A had 421 output neurons operating as a digital thermometer, and network B had only one output neuron operating as an analog device. Network A provided through the high number of output neurons redundant information to test for the correctness of the result because a sequence of neurons up to the clearing temperature has to be "on" and the following neurons have to be "off". Thus, deviation of individual neurons that are supposed to be within each sequence of "on" or "off" neurons indicates an instability in the prediction. The two entries for network A are given relative to the total number of structural patterns presented and relative to the portion of these structural patterns that were recognized by the network (cf. text).

data set improved until the end of the training where it converged to an approximately constant value. More detailed information about the distribution of the errors is shown in Table 2. Obviously, the majority of the clearing temperatures could be predicted within a narrow range of the actual values independent of the number of output neurons. For network B the distributions of the errors for the test set and the training set are shown in Figure 4 as histogram plots. As expected the results of the training set were better than the results of the test set which is also indicated by the standard deviation of 3.8° for the training set and 16.4° for the test set in network B. The reason for the deviation of the predicted from the actual temperatures may have many causes. Besides a possibly limited capability of the neural network approach there are external factors influencing the quality of the training set. First, there will most likely be some errors in the database as well as in the underlying literature, and, second, the neural net may not have an adequate number of molecules in the training set to represent the breadth of nematic liquid crystals.

The results show that a neural network can be used to predict the clearing temperature of liquid crystals with a small standard deviation. This can be extremely helpful in chemical synthesis because a preselection can be made between liquid−crystalline compounds that have a desired clearing temperature and those that have not. If one takes the prediction quality of ±20° as relevant to direct chemical syntheses one obtains a ratio of ≈9:1 in favor of selecting the correct structure by applying this neural network. If one has a higher requirement for the prediction quality of, say, ±5°, one still obtains a selection ratio of ≈1:1. As a consequence, synthesis for molecules with a given physical property can be planned much more efficiently. Furthermore, once the training has been completed, the knowledge retrieval by a neural network is considerably faster than with classical methods of database searches. Even though the neural network shown here does not allow a general structure encoding it can predict the clearing temperatures of more than 160×10^9 different molecules if one calculates all permutations of fragments possible by the encoding chosen for this network. On our computer the neural network can predict the properties of about 1.9×10^6 molecules per hour CPU time.

CONCLUSIONS

Classical methods rely on the knowledge of explicit rules, which must be derived by the scientist, to predict properties of unknown materials. The neural network does not need the explicit definition of rules, but it determines correlations of the chemical structural patterns to the output properties on its own. The prediction of properties from chemical structures presented here does not require elaborate calculations of the conformations of molecules nor of their aggregation states to derive the clearing temperature. The method of deriving material properties with neural networks can be applied to many other problems where the physical property is related to the chemical structure. Structural classes like steroids, peptides, and prostaglandins could be correlated with their biological activity. Also, polymers and copolymers lend themselves to an approach like this to predict material properties from the chemical structures. The development of a scheme that could potentially be used to generalize the structural encoding of molecules is underway.

REFERENCES AND NOTES

(1) Burns, J. A.; Whitesides, G. M. *Chem. Rev.* **1993**, *8*, 2583−2601.
(2) Meyer, B.; Hansen, T.; Nute, D.; Albersheim, P.; Darvill, A.; York, W.; Sellers, J. *Science* **1991**, *251*, 542−544.
(3) Radomski, J. P.; Halbeek, H. v.; Meyer, B. *Nat. Struct. Biol.* **1994**, *1*, 217−218.
(4) Clouser, D. L.; Jurs, P. C. *Anal. Cim. Acta* **1994**, *3*, 221−231.
(5) Mitchell, B. E.; Jurs, P. C. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 58−64.
(6) Curry, B.; Rumelhart, D. E. *Tetrahedron Comput. Methodol.* **1990**, *3*, 231−237.
(7) West, G. M. J. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 806−814.
(8) Gakh, A. A.; Gakh, E. G.; Sumpter, B. G.; Noid, D. W. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 832−839.
(9) Egolf, L. M.; Wessel, M. D.; Jurs, P. C. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 947−956.
(10) Burden, F. R. *Quant. Struct.-Act. Relat.* **1996**, *15*, 7−11.
(11) Peterson, K. L. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 896−904.
(12) So, S.; Karplus, M. *J. Med. Chem.* **1996**, *39*, 1521−1530.
(13) Schröder, R.; Kränz, H.; Vill, V.; Meyer, B. *J. Chem. Soc., Perkin Trans. 2* **1996**, 1685−1689.
(14) Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing*; MIT Press: Cambridge, 1986; Vol. 1.
(15) Vill, V. *Liquid Crystals*; Springer: Berlin, 1992−1995; Landolt-Boernstein, New Series, Vol. IV, Chapter 7.
(16) Schuur, J. H.; Selzer, P.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 334−344.

CI960482R