Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Molecular Graphics and Modelling 23 (2004) 1–14

Journal of Molecular Graphics and Modelling
www.elsevier.com/locate/JMGM

Generation of QSAR sets with a self-organizing map

Rajarshi Guha, Jon R. Serra, Peter C. Jurs [*]

Department of Chemistry, Penn State University, 152 Davey Laboratory, University Park, PA 16802, USA

Received 31 July 2003; received in revised form 1 December 2003; accepted 3 March 2004
Available online 27 April 2004

Abstract

A Kohonen self-organizing map (SOM) is used to classify a data set consisting of dihydrofolate reductase inhibitors with the help of an external set of Dragon descriptors. The resultant classification is used to generate training, cross-validation (CV) and prediction sets for QSAR modeling using the ADAPT methodology. The results are compared to those of QSAR models generated using sets created by activity binning and a sphere exclusion method. The results indicate that the SOM is able to generate QSAR sets that are representative of the composition of the overall data set in terms of similarity. The resulting QSAR models are half the size of those published and have comparable RMS errors. Furthermore, the RMS errors of the QSAR sets are consistent, indicating good predictive capabilities as well as generalizability.

Keywords: Self-organizing map; QSAR modelling; Sphere exclusion method

1. Introduction

Self-organizing maps (SOM) [1] are a class of unsupervised neural networks whose characteristic feature is their ability to map non-linear relations in multi-dimensional data sets into easily visualizable two-dimensional (2D) grids of neurons. SOMs are also referred to as self-organized topological feature maps since the basic function of a SOM is to display the topology of a data set, that is, the relationships between members of the set. SOMs were first developed by Kohonen in the 1980s, and since then they have been used as pattern recognition and classification tools in various fields including robotics [2], astronomy [3] and chemistry.

Neural networks have been used quite extensively in chemistry [4] and chemometrics. There are numerous examples of studies using SOMs as the underlying network. Applications in chemistry include spectroscopy [5–8], prediction of NMR properties [9] and prediction of reaction products [10,11].

SOMs have also been applied to studies in the field of QSAR/QSPR [12]. The fundamental premise of QSAR studies is that structurally related (similar) compounds will have similar properties. Determining *similarity* is a complex task, and many methods exist such as principal components analysis (PCA) and hierarchical cluster analysis. The fact that a SOM is able to extract topological information from a data set makes it a valuable tool for detecting similarities in a data set. Thus, it is to be expected that neighboring neurons in a 2D SOM grid will be similar to each other. If each neuron in such a SOM grid can be assigned a molecule, groups of *similar* molecules could be found.

Many studies have used a SOM to perform the actual QSAR [13–16] analysis by detecting relationships between structures and activities of interest. Other applications use SOMs at different stages of the QSAR study, for example, the use of an SOM to choose the best subset of molecular descriptors [17,18] to perform a QSAR analysis. However, another important step in QSAR study is the generation of training, cross-validation and prediction sets. Many methods have been used including random selection, activity-ranked binning and sphere exclusion algorithms [19]. Various forms of neural networks have also been employed in the selection of training sets. Examples include Kennard–Stone [8,20], D-Optimal [8] and Kohonen [8,20–22] neural networks. Set selection is also an important step in QSAR modeling of chemical libraries. Most strategies for this are based on a combination of principal components analysis for dimen-

[*] Corresponding author. Tel.: +1-8148653739; fax: +1-8148653314.
E-mail address: pcj@psu.edu (P.C. Jurs).

1093-3263/$ – see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmgm.2004.03.003

sionality reduction followed by statistical molecular design (SMD) [23–25]. The technique of SMD has also been combined with hierarchical design in the study of chemical libraries [26].

The goal of this study is to implement a set generation technique, utilizing a SOM together with whole molecule descriptors, to initially classify the data set and subsequently use this classification to generate training, cross-validation and prediction sets for QSAR studies whose composition would mirror the overall composition of the entire data set. This technique should lead to the generation of QSAR models that exhibit equal or higher validity than models generated from subsets developed with random selection or activity-ranked binning. The distribution of the members of the training and prediction sets (with respect to each other in *descriptor space*) is also studied by calculating a molecular diversity index [27]. In addition, the results from SOM-generated QSAR sets are compared to results obtained using QSAR sets created using traditional activity binning as well as sets created using a sphere exclusion algorithm described by Golbraikh and Tropsha [19].

2. Theory of the SOM

A Kohonen self-organizing map is an unsupervised neural network that uses only the independent variables of the data set, here molecular structure descriptors. The SOM can be viewed as an elastic net of points, which are molded to the specific features of the compounds used for training. Training occurs as the SOM's neurons compete with each other for selection. At each training iteration, the selected neuron and its neighbors are modified to resemble the applied example compound.

Although SOMs can appear in a variety of forms [1], this study implemented a map in the form of a square grid. In order that each neuron has the same number of neighbors the grid is designed so that it wraps around the edges, effectively transforming the grid of neurons into a torus. However, for ease of visualization and discussion we will refer to the arrangement as a square grid.

Each compound in the training set is represented by a vector

$$X_i = (x_{i1}, x_{i2} \dots x_{in}),$$

where n is the number of molecular structure descriptors employed. Each neuron on the square SOM grid is also a vector

$$M_i = (m_{i1}, m_{i2} \dots m_{in}),$$

where n is the number of descriptors in each member of the training set. The neurons on the grid are initialized with random vectors. The size of the grid is chosen by trial and error, guided by a rule of thumb described by Chen and Gasteiger [11], which states that the number of neurons should be approximately one to three times the number of examples in the training set.

The training process for a SOM is iterative. Each training iteration involves comparing each member of the data set to all the neurons in the grid and determining the grid neuron that is closest, in terms of Euclidean distance

$$d_{pq} = \sqrt{\sum_{i=1}^{n} (x_{pi} - m_{qi})^2}$$

to the submitted neuron. The grid neuron that is most similar to the input vector is the winner. Then, the winning neuron and the surrounding neurons are modified, according to this equation:

$$m_i(t+1) = m_i(t) = h_{ci}(t)[x(t) - m_i(t)],$$

where t represents training iterations, m_i represents the winning neuron and x represents the data set member. Here $h_{ci}(t)$ is termed the neighborhood kernel, and it determines which neurons are neighbors and how such neighboring neurons will be modified. Neurons that are further away (in a topological sense) from the winning neuron are modified to a smaller degree than neurons that are closer. The simplest neighborhood kernel is the *bubble* function [1,18] (also referred to as a fixed window) which is non-zero for the neighborhood but zero elsewhere. The map in this study implemented a Gaussian kernel [1], defined as

$$h_{ci} = \alpha(t)\exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right),$$

where $\sigma(t)$ is the neighborhood radius at time t which monotonically decreases with time. Thus, the number of neurons considered to be neighbors decreases as training progresses. The term $||r_c - r_i||$ represents the Euclidean distance between the winning neuron and the neighboring neuron. Thus, neighbors closer to the selected neuron will undergo a larger modification than neurons further away from the selected neuron. $\alpha(t)$ is the learning factor and it influences the extent to which a neuron should be modified. Initially, neurons within a large radius surrounding the selected neuron are considered neighbor neurons. The radius of the neighborhood is decreased in successive training iterations, and in the last stages of training only the nearest neighbors of the selected neuron are modified. The effect of this variable neighborhood function is that in the early stages of training the neurons are modified on a global scale, which leads to a global ordering. Near the end of training, the smaller neighborhood results in fine-tuning of the map features. The neighborhood function thus controls the sensitivity of the map.

The actual modification is controlled by the learning factor, $\alpha(t)$. The learning factor is a function that monotonically decreases from 1 to 0 as training progresses. Once $\alpha(t)$ reaches zero, training stops. Kohonen [1] mentions several ways of modifying $\alpha(t)$, and the implementation used in this study employs a constant decrement:

$$\alpha(t+1) = a(t) - 0.01,$$

which implies that after 100 training iterations α will be zero. This represents an upper limit on the number of training iterations.

The implementation consisted of a 13 × 13 grid. The data set we used to test this method consisted of 333 molecules. According to Chen and Gasteiger [11] the grid should contain 333–999 neurons. This translates to grid sizes ranging from 18 × 18 up to 31 × 31. However, we noted that for grids larger than 15 × 15 the SOM converged to a configuration in which the training set was mapped relatively evenly over the grid with little apparent clustering. In addition, the use of larger grids increased the running times significantly. The method of choosing a grid size does appear to be arbitrary. However, given the fact that following Chen's rule of thumb produced grids with hardly any clustering observable, we felt that examining smaller grid sizes was justified. Using a 13 × 13 grid of neurons the SOM usually required 80–90 training iterations for the grid neurons to converge to their final values. Depending on the number of descriptors used to represent each compound, this took approximately 3–6 min on an AMD 750 MHz Duron processor running RedHat Linux 7.3.

After the SOM was trained, the results were analyzed to detect clusters of neurons. In this context, a cluster refers to neurons that have similar Euclidean distances from each other. As mentioned by Satoh et al. [10], ''recognition of boundaries of clusters in a Kohonen network is a difficult task''. This was implemented by considering two neurons having a distance less than a user-specified value to be a part of the same cluster. Starting with an arbitrary neuron, we assigned an arbitrary class label. Next, we considered the distances to all the nearest neighbor neurons. Using the rule mentioned above, the neighboring neurons were assigned classes; either the same class as the initial neuron or the opposite class. This procedure was then repeated with all the neurons in the grid. An example of the grid layout after cluster detection (using three different threshold values) is shown in Fig. 1. The diagrams are based on the grid generated using the BCUT and 2D autocorrelation descriptor combination.

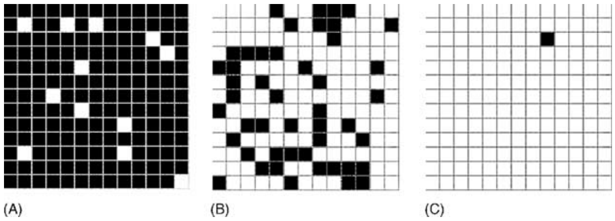


Fig. 1. A graphical representation of the SOM after the cluster detection step using the BCUT and 2D autocorrelation Dragon descriptor subset. Black and white squares represent the individual classes. Grids A, B and C were obtained by setting the threshold value to 0, 1.2 and 3.6, respectively. Grid B was used to generate the final QSAR sets for this Dragon [30] descriptor subset.

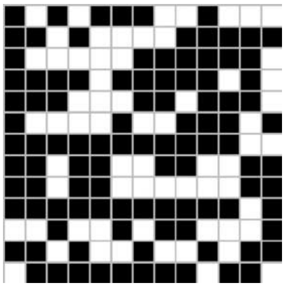


Fig. 2. A graphical representation of the distribution of whole data set on the grid after it has been divided into two classes based on the BCUT and 2D autocorrelation Dragon [30] descriptor combination. Black and white squares represent the two different classes.

The final step in this procedure was to assign classes to the actual data set members by submitting each data set vector to the trained grid. The class of the closest grid neuron (in terms of Euclidean distance) was assigned to the data set member.

The result of the cluster detection procedure was to divide the data set into two classes. Fig. 2 shows the how the classified data set is distributed over the SOM. As mentioned before, the arbitrariness of cluster detection lies in the fact that the user must specify a *distance threshold* value. Too small a value or too large a value results in all the data set members being assigned to the same class. As the threshold value progresses from zero to larger values the SOM generates a bulk class containing the majority of the data set members and a minor class. At one point, the populations of both classes will be approximately equal, and then with further increase of the threshold value the populations once again get skewed. It is thus clear that the threshold value cannot be chosen at random. Below we describe the method that we employed to arrive at a threshold value.

It should be noted that the classification of the data set by the SOM is not intended to correspond to a classification based on any structure–activity relationship. The aim of the classification is to simply divide the data set into two sets differing in structural features, as characterized by whole molecule descriptors.

3. Using the SOM to create sets

In the present study, the SOM was used to generate training, prediction and cross-validation sets (hereafter referred to collectively as QSAR sets) for QSAR studies using the ADAPT [28,29] software system. Previously, these sets had been generated by randomly selecting the requisite number of molecules from the binned (based on activity) data set. However, due to the random selection process, the binning procedure does not necessarily create sets that represent the composition of the whole data set. Yan and Gasteiger [22] have used a SOM to select QSAR sets, in which sets were created by simple selection of grid points. As a result their method is similar to the sphere exclusion technique in that there is a correspondence between the training and prediction set points in descriptor space. However, the technique described by Yan and Gasteiger [22] does not necessarily maintain a correspondence between the composition of the QSAR sets and the overall data set. Our method emphasizes the use of characteristic features of the data set to create sets whose composition would mirror the overall data set. This is achieved by using the SOM to divide the data set into two classes, based on the molecular structure descriptors representing the compounds of the data set. These two classes thus represent the SOM classification of the whole data set into a major and minor class (say, Class I and Class II, respectively).

As described above, the threshold value controls the population of the two classes. We initially ran the SOM with the threshold value set to zero. The output of this run reported the distances between all the neurons in the grid. This distance information was used to determine the range of threshold values to be considered in subsequent runs of the SOM. The next step was to run the SOM several times in succession, with threshold values ranging from about 5 to 90% of the maximum distance reported in the initial run. Each run generated a set of class assignments. We considered those runs that generated a bulk class having approximately 80% of the entire data set. The difference between the populations of the bulk and minor class for each of these runs, *D*, was noted. A large jump in the value of *D* was usually seen at one point in the series. This may be seen in Fig. 3, which plots *D* versus the threshold value (represented as a percentage of the maximum distance in the grid when the threshold value is set to zero). The descriptor subset supplied to these SOM runs was the MoRSE–WHIM subset. The classification results from the run that generated the lower value of *D* for the jump were used for the subsequent creation of QSAR sets. From Fig. 3 it is apparent that there is a large jump from 23 to 24% as well from 4 to 5%. However, we did not consider these jumps since the number of molecules in the bulk class for these jumps was not close to 80% of the whole data set. Instead the grid configuration that corresponds to the jump from 9 to 11% had a bulk class that contained 80.1% of the whole data set. Thus the grid results from the run using a threshold value of 11% were used subsequently. After the data set had been classified, the information produced was used to create the actual QSAR sets. The aim of this technique was to generate QSAR sets whose composition mirrored that of the whole data set in terms of overall similarity. At this point the SOM had classified the data set into two classes (Class I and Class II), members of each class being similar to each other but dissimilar to members of the other class.

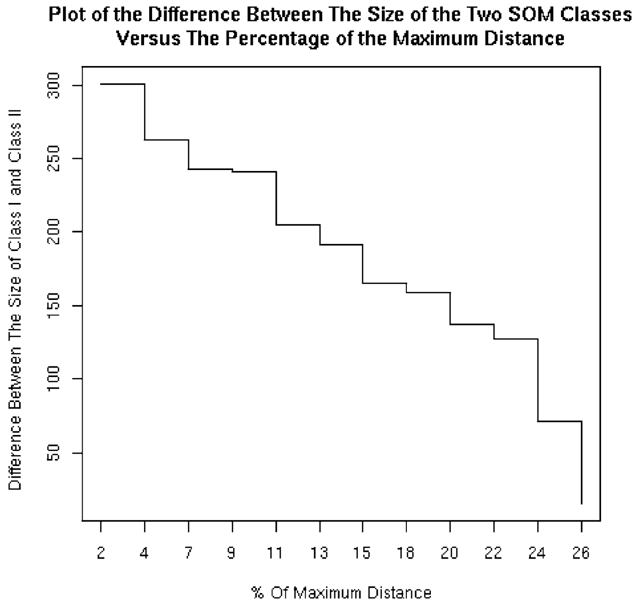


Fig. 3. A plot showing the variation of *D* (the difference in size between major and minor SOM classes) versus the threshold value for the SOM. In this plot the threshold value is represented as a percentage of the maximum distance in the grid for an SOM in which the threshold value was set to 0. The descriptor set used to generate the grids described in the plot was the MoRSE–WHIM Dragon descriptor subset.

Now, for example, say that Class I contains 75% of the whole data set and Class II contains the other 25%. Our premise is that QSAR sets which contain Class I and Class II molecules distributed according to their percentages in the overall data set will be more representative of the overall data set and thus should lead to good predictive models. Continuing with the example, let us assume that we have a data set of 100 molecules and the SOM classifier splits this data set in to 75 molecules in Class I and 25 molecules in Class II. We also assume that for the QSAR sets, the training set should contain 80% of the data set and the cross-validation and prediction sets should each contain 10%. To make the training set composition similar to that of the overall data set it will have 80 compounds, of which 75% (60 compounds) will be from Class I and 25% (20 compounds) will be from Class II. Similarly the cross-validation and prediction sets will each have 10 compounds, of which 75% (eight compounds) will be from Class I and 25% (two compounds) will be from Class II. Due to rounding the final QSAR sets may not have exact number of compounds described but may differ by 1. The breakup of the QSAR sets among the SOM classes discussed above is represented diagrammatically in Fig. 4 with the exact numbers of compounds rounded appropriately.

However, unlike methods such as the sphere exclusion method, discussed below, there is no guarantee that the QSAR sets generated cover the entire descriptor space. Though it is possible that a specific QSAR set is generated by sampling points from a small region of the grid while still covering both classes, it appears that this does not occur. Fig. 5 shows the distribution of the QSAR sets over the grid. As can be seen, the members of each set seem to be relatively evenly distributed over the grid. The diagrams in Fig. 5 are based on the BCUT and 2D autocorrelation descriptor

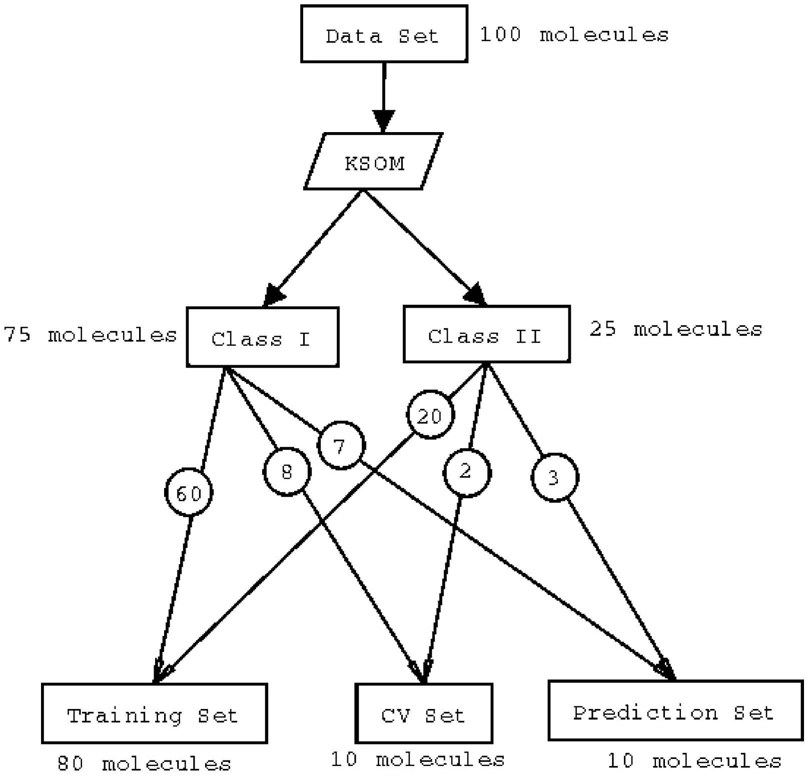


Fig. 4. A diagrammatic representation of the method we use to generate QSAR sets from the SOM classification of the whole data set. The numbers within circles are the number of molecules from that class that present in the specific QSAR set.

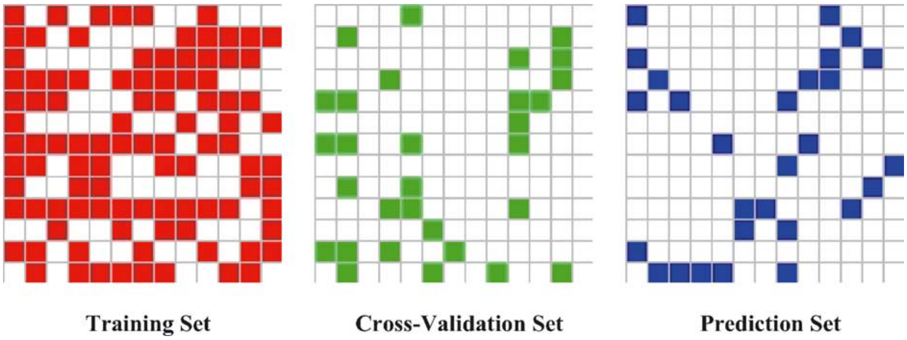


Fig. 5. The three diagrams represent the distribution of the QSAR sets over the surface of the SOM. The grid was trained with the BCUT and 2D autocorrelation Dragon [30] descriptor combination.

combination. The other QSAR sets generated from other Dragon [30] descriptor combinations investigated generated similar plots.

4. Sphere exclusion

This method described by Golbraikh and Tropsha [19] uses the concept of molecular diversity [27] coupled with a sphere exclusion algorithm to generate training and prediction sets which satisfy these criteria: points in the training and prediction sets should be close (in terms of descriptor space) to each other, and the training set should be diverse, as measured by the value of its diversity index [27].

Golbraikh describes three types of sphere exclusion algorithms. A brief summary of the general sphere exclusion algorithm follows. For a training set with N compounds and described by K descriptors, the compound with the highest activity is first selected and placed in the training set. Next, a radius, R, is calculated. R is given by the formula

$$R_c\left(\frac{V}{N}\right)^{1/K},$$

where V is the volume of the space occupied by the points of the data set in the descriptor space and c is a user defined constant termed the dissimilarity level (DL) [27] and essentially controls the number of molecules placed in the training and prediction sets. To simplify calculations, the descriptor space is normalized using the formula

$$X_{ij}^{n} = \frac{X_{ij} - X_{j,\min}}{X_{n,\max} - X_{j,\min}},$$

where X_{ij} is the non-normalized jth descriptor for the ith molecule and X_{ij}^{n} is the normalized value of the descriptor. Thus after normalization, $V = 1$ and the equation for the radius simplifies to

$$R = c\left(\frac{1}{N}\right)^{1/K}.$$

After a value of R is obtained, a sphere with this radius is centered at the point chosen above, and all compounds that lie within this sphere (except the center point) are included in the prediction set and removed from the data set so as not to be considered later. At this point if there are no more points left to consider the algorithm halts, otherwise the distances from the remaining points to the centers of all the spheres considered so far are calculated. The distance is given by

$$\mathrm{d}_{ij} = \sqrt{\sum_{a=1}^{k}(X_{ia} - X_{ja})^{2}},$$

where X_i and X_j are the descriptor vectors for the ith and jth molecules respectively and k is the number of descriptors. One of the points is chosen to be the center of the next sphere and this process is repeated. The manner of choosing the next point gives rise to three variations of the sphere exclusion algorithm: the point that had the smallest d_{ij}, the point that had the largest d_{ij}, or randomly choosing a point. In this study we implemented the first option. The result of this algorithm is to generate a training and prediction set. Since the ADAPT methodology requires the use of a cross-validation set, we randomly selected the required number of molecules out of the training set to create the cross-validation set.

5. Descriptors for the SOM

The SOM requires that each compound be represented by a set of molecular structure descriptors. We used an external set of descriptors (from the Dragon [30] program), as opposed to the ADAPT descriptors since we wanted to classify the data set in terms of global features, rather than specific structural trends. As a result, various subsets of Dragon descriptors which are holistic in nature were used, rather than ADAPT descriptors, many of which concentrate on specific structural features. Another reason for not using ADAPT descriptors is that the resultant QSAR sets would indirectly contain the information generated by the ADAPT descriptors and thus using same descriptors again during model development would lead to the possibility of biased models (in that the same information that was used to arrange the molecules would be used again when predicting their activity).

Table 1
Type and number of Dragon descriptors used by the SOM to generate training, cross-validation and prediction sets for QSAR models

Descriptor name	Number of descriptors	References
BCUT	123	[31–33]
BCUT and 2D autocorrelation	44	[31–33,35,36,38,39]
BCUT and Galvez topological indices	63	[31–33,40–42]
GETAWAY	128	[43–45]
MoRSE and 2D autocorrelation	173	[35,36,38,39,46,47]
MoRSE and GETAWAY	223	[43–47]
MoRSE and WHIM	139	[46–48,50,53,54]

Thus this technique proceeds in two stages and requires two sets of descriptors, preferably orthogonal. In the first stage, one set of descriptors is used to classify the data set with the SOM leading to creation of training, cross-validation and prediction sets. The second stage involves the generation of the actual QSAR model using the second set of ADAPT descriptors and the training, cross-validation and prediction sets created in the first stage.

As mentioned above the, descriptors for the first stage were taken from the Dragon program. Several combinations of the Dragon descriptors were selected to see if they could provide a holistic description of the molecules. The number of descriptors in each combination was reduced using correlation and identical testing before using them in the SOM algorithm. A brief description of the descriptors used for the SOM clustering follows. The size of each reduced Dragon descriptor set is shown in Table 1.

The BCUT metrics [31–33] are hybrid descriptors derived from the Burden parameters [31] which originally combined the atomic number of an atom and the bond types for adjacent and non-adjacent atoms. The BCUT metrics improve upon the number and type of atomic features that can be encoded. This descriptor has shown significant utility in the measurement of molecular diversity [34].

2D autocorrelation descriptors were chosen. There are three different such descriptors—Moreau–Broto [35–37], Moran [38] and Geary [39]. These descriptors sum products of atom weights of terminal atoms of all paths of a specific path length.

Galvez topological charge indices [40–42]—these descriptors use the distance matrix to evaluate '*charge terms*' which characterize the charge transfer between individual atoms in the molecule.

GETAWAY [43–45]—these descriptors are based on the information contained within the molecular influence matrix [44]. They combine the geometrical information in the influence matrix and topological information in the molecular graph weighted by various atomic properties. As a result there are two sets of GETAWAY descriptors, the H and R GETAWAY, both of which we chose to use in the classification stage.

Table 2
Summary of the number of molecules present in the training, cross-validation and prediction sets

Set	Number of molecules	Percentage of molecules
Training	267	80.1
Cross-validation	32	9.6
Prediction	34	10.3
Total	333	100

The sizes of these sets were the same for all the Dragon descriptor subsets investigated.

3D MoRSE [46,47] descriptors are fixed length representations of 3D molecular structure and are based on electron diffraction data. Individual descriptors are obtained by considering different weighting functions as described in the literature.

WHIM [48–54] descriptors describe a molecule in terms of size, shape, symmetry and atom distribution, and are based on a principal components analysis on the centered molecular coordinates with different weighting schemes [53].

Our studies used both individual sets as well as combinations of the above descriptors. The sphere exclusion method also used the same combinations of external Dragon descriptors for the generation of the QSAR sets.

6. Results and discussion

To test this method we generated QSAR models using the 333-compound pcDHFR data set that was studied by Mattioni and Jurs [55]. The structures and activity values for all the molecules are contained in above-mentioned reference. To generate the QSAR sets we fed combinations of Dragon descriptors to the SOM, and its output was used to generate the sets. For each Dragon descriptor subset the sizes of the training, CV and prediction sets were the same. To ensure a large enough training set, 80% of the data set was placed in the training set and the remaining 20% was divided equally amongst the CV and prediction sets. The actual number of molecules in each set is summarized in Table 2. After the QSAR sets were generated, we calculated ADAPT descriptors for the entire data set of 333 molecules. This generated 248 descriptors for each molecule. The number of descriptors was then reduced via objective feature selection (using correlation and identical testing) to generate a reduced pool of 74 descriptors. The reduced pool of ADAPT descriptors was then used with the QSAR sets created from each of the Dragon descriptor combination, to non-linear computational neural network (CNN) models using the ADAPT methodology. In total we used six combinations of Dragon descriptors to generate six non-linear CNN QSAR models (Table 1).

7. Non-linear CNN models

To generate non-linear models, the descriptor subsets selected by a genetic algorithm were fed to a three-layer, fully connected, feed-forward neural network to test fitness. The best neural network models were those that minimized the cost function shown below:

$$\text{cost} = \text{TSET}_{\text{RMS}} + 0.5|\text{TSET}_{\text{RMS}} - \text{CVSET}_{\text{RMS}}|,$$

where TSET_{RMS} is the RMS error for the training set, $\text{CVSET}_{\text{RMS}}$ is the RMS error for the cross-validation set and the factor 0.5 is a cross-validation factor. The above cost function thus favors CNN models which have low training set error but also penalizes models which are unable to generalize by considering the cross-validation set error. The specific value of 0.5 for the cross-validation factor is empirically determined and has been shown to provide good results.

After several of the top (low cost) models were obtained a more rigorous analysis was performed on each model to identify the optimal neural network parameters. The results for the non-linear models are summarized in Table 3. Though the number of descriptors in the two best models (see Table 4) is significantly lower than the number in the published model, they are of similar types, the majority being simple structural counts and topological path descriptors.

The MoRSE–2D autocorrelation Dragon descriptor combination generated QSAR sets which produced a CNN model whose prediction set RMS error was slightly larger than the original value, whereas the prediction set error for the models that were generated from QSAR sets produced by using the GETAWAY and the MoRSE–WHIM Dragon descriptor sets match the predicted value. However, in all cases the RMS errors for the training and cross-

Table 3
Summary of the non-linear CNN models using training, cross-validation and prediction sets created by the SOM and Dragon descriptor combinations

Dragon descriptor	CNN architecture	RMSE			R^2		
		Training set	Cross-validation set	Prediction set	Training set	Cross-validation set	Prediction set
BCUT and 2D autocorrelation	5-3-1	0.63	0.68	0.79	0.68	0.60	0.67
BCUT and Galvez topological indices	5-3-1	0.62	0.62	0.71	0.69	0.66	0.64
GETAWAY	5-2-1	0.68	0.60	0.73	0.64	0.76	0.67
MoRSE–2D autocorrelation	5-3-1	0.63	0.63	0.68	0.68	0.60	0.74
MoRSE–GETAWAY	9-5-1	0.49	0.59	0.76	0.80	0.58	0.80
MoRSE–WHIM	6-5-1	0.60	0.61	0.65	0.75	0.78	0.64
Published results [55]	10-6-1	0.45	0.49	0.66	0.84	0.78	0.64

Table 4
ADAPT descriptors present in the two best non-linear CNN models

Descriptor	Type	Range
MoRSE and 2D autocorrelation		
N7CH	Topo	7.0–28.0
MOLC-8	Topo	0.6–2.8
NDB-13	Topo	0.0–7.0
NAB-15	Topo	6.0–23.0
WPSA-3	Hybrid	17–57.4
MoRSE and WHIM		
V6P7	Topo	2.1–0.5
WTPT-4	Topo	0.0–12.2
N7CH	Topo	7.0–28.0
NDB-13	Topo	0.0–7.0
MDE-23	Topo	0.0–28.1
RPCS	Hybrid	0.0–8.1

N7CH, number of seventh order chains χ index [56–58]; MOLC-8, average distance sum connectivity (topological index J) [59,60]; NDB-13, number of double bonds; NAB-15, number of aromatic bonds; WPSA-3, partial positive surface area multiplied by the total molecular surface area divided by 1000 [61]; RPCS, relative positive-charged surface area [61]; MDE-23, molecular distance edge between primary and secondary carbons [62]; WTPT-4, sum of atom IDs for oxygen [63].

validation set were significantly larger than those for the reported model. The higher cross-validation set error could indicate a loss of generalizability in these models. On the other hand the RMS errors for the training and cross-validation sets generated from the MoRSE–GETAWAY combination are much closer to those reported, though the prediction set error is now significantly larger. However, the attractive feature of the models generated from QSAR sets produced by MoRSE–2D autocorrelation, GETAWAY and MoRSE–WHIM Dragon descriptor sets are that they are 5- or 6-descriptor models. Furthermore, the number of neurons in the hidden layers in these three models are all less than in the published model, indicating a simpler neural network.

Table 4 lists the descriptors present in the two best non-linear models. The two best models have a similar set of ADAPT descriptors when compared to the published model, though none of them include a geometric descriptor. The R^2 values for the two best models (i.e., the models using QSAR sets generated using the MoRSE–2D autocorrelation and MoRSE–WHIM Dragon descriptor combinations) are close to those reported for the best model. These are summarized in Table 5. The R^2 values for the training and cross-validation sets produced by the MoRSE–WHIM combination compare favorably to those published. The R^2 value for prediction set produced by the MoRSE–WHIM combination is a little higher than the reported value, but is not significantly larger. Considering the fact that the R^2 for prediction set produced by the MoRSE–2D autocorrelation combination is the same as that published could indicate that a combination of MoRSE, 2D autocorrelation and WHIM descriptor sets would lead to QSAR sets which would lead to a CNN model with better correlation coefficients overall.

Table 5
Comparison of R^2 values for the training, cross-validation and prediction sets created by the SOM using Dragon descriptors[a]

	Training set	Cross-validation set	Prediction set
MoRSE–2D autocorrelation	0.68	0.60	0.64
MoRSE–WHIM	0.75	0.78	0.67
Published [55]	0.83	0.78	0.64

[a] The models produced were CNN models. See Table 3 for the model architectures.

However, it should be noted that though the R^2 value is a good test for evenly distributed data, it is not always reliable for an unevenly distributed data set as the one used in this study. As a result we feel that the RMS errors provide a more reliable indication of the fitness of a model.

The plot for the predicted versus experimental values for the model generated using QSAR sets produced using the MoRSE–WHIM Dragon descriptor combination is shown in Fig. 6. Molecules in the prediction set were classified as outliers if their predicted value was two standard deviations away from the mean. This criterion led to one outlier, whose structure is shown in Fig. 7. The best, published model also classified a single outlier. Ideally, we would like the same outliers to be detected by either method. Although this is not the case, it should be noted that there is a structural similarity in the outliers presented in Fig. 7. Although the original work does not provide an explanation of why that outlier is not predicted well, the fact that the SOM-based technique predicts a structurally similar outlier indicates that that this technique is able to take into account similarity features of the data set in the creation of the QSAR sets.

An important feature of the two best CNN models (using QSAR sets generated from the MoRSE–2D autocorrelation and the MoRSE–WHIM Dragon descriptor combinations) is

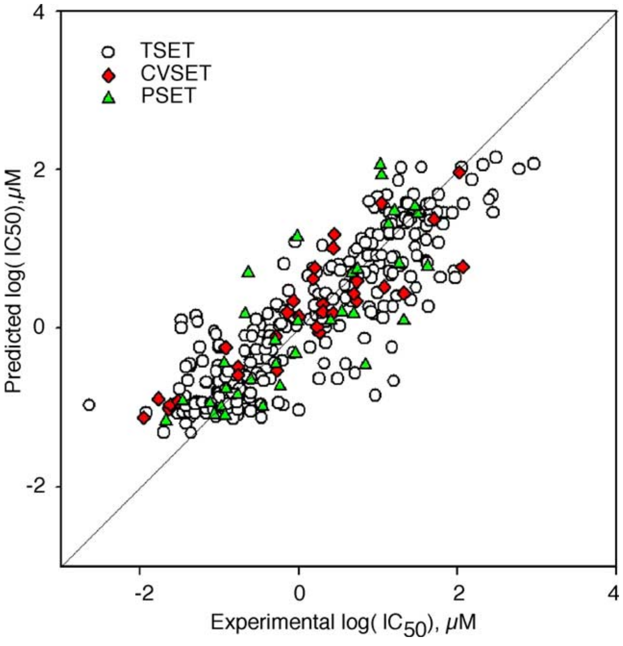
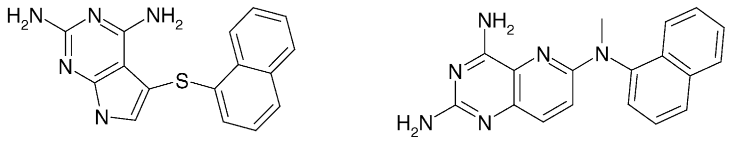


Fig. 6. Plot of experimental vs. predicted log IC_{50} for the 6-5-1 CNN model generated using training, cross-validation and prediction sets created using the SOM and MoRSE–WHIM Dragon descriptor combination.



Outlier detected by best CNN model in this study

Published Outlier for the Best pcDHFR model[55]

Fig. 7. Prediction set outliers.

the consistency between the RMS errors for the training, cross-validation and prediction sets. In many cases a low RMS error for the prediction set would be indicative of a good predictive model. However, at the same time, if the RMS errors for the training and cross-validation sets are much lower than that of the prediction set it could indicate that the model lacks generalizability. Thus one would strive for models that have similar or consistent RMS errors for all the three QSAR sets. As can be seen, this does not occur for the original published results. However, for the best CNN models generated by the SOM-based method, though the RMS errors for the training and cross-validation sets are higher than those reported in the original model, the RMS errors are more consistent over all the three sets. The standard deviation of the RMS errors for the three QSAR sets in the original model is 0.11, whereas the standard deviations in the case of the two best models noted above are 0.02 in both cases. This suggests that the models generated by this method have both sufficient generalizability as well as predictive ability. However, apart from the conclusions regarding the nature of the models themselves these results are indicative of the fact that the QSAR sets that are generated by the SOM are indeed similar to each other and representative of the data set as a whole thus leading to similar predictions made during training and after training (using the external prediction set).

We also reran the original, published 10-6-1 CNN model five times with different QSAR sets generated using activity binning. The results obtained are summarized in Table 6. As can be seen there is a large variation in the RMS errors for the three QSAR sets in each run. Furthermore, when compared to the RMS errors for the best CNN models generated using QSAR sets created by the SOM, we see that the SOM results in general lie midway between the RMS errors from the 10-6-1 models using QSAR sets from activity binning. We believe that this is a good indication for the consistency of results obtained using the SOM to generate representative QSAR sets.

Table 6
A summary of the RMS errors for the 10-6-1 non-linear CNN models using five QSAR sets generated by activity binning

Serial no.	Training set	Cross-validation set	Prediction set
1	0.45	0.59	0.81
2	0.45	0.52	0.73
3	0.44	0.63	0.95
4	0.64	0.64	1.00
5	0.67	0.61	0.95

It thus appears that the technique of using a group of external descriptors coupled with a SOM to generate sets for QSAR modeling do generate improved results. The ability of the SOM to detect similarities in the data set allows us to generate sets that are more representative of the overall data set. As a result, models with fewer parameters (i.e., descriptors) are able to produce results comparable to the original model that had nearly twice the number of parameters and in addition produce consistent RMS errors over the three QSAR sets.

8. Sphere exclusion

For comparison, results of CNN models generated using different QSAR sets created by the sphere exclusion method are presented in Table 7. For each set of external descriptors used, the model with lowest cost is reported. None of the models seem to be significantly better than the published model. The architectures are not significantly simpler than the reported 10-6-1 architecture. However, the R^2 values and RMS errors are comparable though none of the models seem to provide an improvement over the published statistics. In addition, there is no much of a difference in the RMS errors for models that are generated from QSAR sets that were created using different Dragon descriptor combinations. However, when comparing the results from the sphere exclusion method to those obtained from the SOM technique it appears that the SOM generated QSAR sets produce better models in terms of size (i.e., requiring fewer descriptors), with RMS errors being comparable. In addition, the RMS errors for the three QSAR sets in the models generated by the sphere exclusion method do not show much consistency. The RMS error for the prediction set is usually higher than the RMS errors for the training and cross-validations sets by 0.1–0.3. This is similar to the nature of the RMS errors in the original model. Due to the nature of the sphere exclusion algorithm one would expect that the resultant QSAR sets would be similar to each other and thus lead to consistent RMS errors. The fact that it does not is a possible indication that a simple Euclidean distance between individual molecular descriptor vectors is not sufficient to characterize similarity of molecules of in a data set. Thus the sphere exclusion method does not appear to generate QSAR sets

Table 7
Summary of the best non-linear CNN models generated from QSAR sets created using the sphere exclusion algorithm

Dragon descriptor[a]	CNN architecture	RMSE			R^2		
		Training set	Cross-validation set	Prediction set	Training set	Cross-validation set	Prediction set
BCUT and 2D autocorrelation	9-3-1	0.55	0.54	0.87	0.74	0.78	0.33
BCUT and Galvez topological indices	9-8-1	0.46	0.50	0.87	0.83	0.81	0.36
GETAWAY	8-5-1	0.56	0.56	0.63	0.75	0.80	0.67
MoRSE–2D autocorrelation	9-8-1	0.49	0.53	0.68	0.81	0.82	0.68
MoRSE–GETAWAY	8-6-1	0.52	0.58	0.64	0.79	0.84	0.67
MoRSE–WHIM	7-6-1	0.50	0.57	0.82	0.80	0.77	0.52
Published results [55]	10-6-1	0.45	0.49	0.66	0.83	0.78	0.64

[a] The external descriptor set used by the sphere exclusion algorithm to create the training and prediction sets.

that can produce models with both generalizability as well as predictive ability for this data set.

9. Randomization studies

The best non-linear model (i.e., the one generated using QSAR sets produced by the MoRSE–WHIM Dragon descriptor combination) was subjected to randomization tests. The first set of tests involved generating random training, cross-validation and prediction sets. These sets were then used to generate a non-linear model with a 6-5-1 CNN architecture five times (each time using randomly generated sets) and noting the average RMS errors. In addition, the variance between the five individual runs for the random sets was also compared to the variance for five runs of the original QSAR sets that gave the best 6-descriptor CNN model. The correlation coefficient for each of the sets in each of the runs was also compared to the correlation coefficients for the best model. The results are summarized in Table 8. The average correlation coefficient (R^2) for the random training, cross-validation and prediction sets were 0.75, 0.73 and 0.56, respectively. These values would indicate that the KSOM technique is not much better than random set generation. As mentioned above, R^2 is not always reliable for an unevenly distributed data set such as the one used in this study and as a result we feel that the RMS errors provide a better indicator of the goodness of a model. Though the RMS errors for the training and cross-validation sets are comparable, the prediction set RMS error is much larger for the random sets. In addition, comparing the standard deviation in the RMS errors for the five runs for the random and KSOM sets indicates that the KSOM technique is much more consistent. For the original best model the standard deviations for the three sets were 0.005, 0.01 and 0.02, respectively. For the random sets the standard deviations were 0.02, 0.03 and 0.13, respectively indicating that predictions made using the random sets were not consistent over several runs. Once again, we believe that this is evidence for the KSOM's ability to generate good sets based on features of the data set.

The next randomization test consisted of regenerating the best non-linear model (using the ADAPT descriptors as reported in Table 4) but scrambling the dependent variable. With the scrambled dependent variable, the best CNN model was regenerated using the original QSAR sets. This process was repeated five times, each time using a scrambled dependent variable, and the average RMSE and R^2 values for the training, CV and prediction sets for the five runs were noted. It would be expected that the resultant model would have relatively high RMS errors for the three sets, as well as low R^2 values. This was indeed the case with the training, cross-validation and prediction sets having RMS errors of 1.04, 1.00 and 0.97, respectively (Table 9). In addition, the R^2 values for the three sets were 0.17, 0.09 and 0.01, respectively. Compared to the RMS and R^2 values for the best model, it appears that chance correlations played little (if any) part in the results for the best model.

Table 8
Comparison of statistics for training, cross-validation and prediction sets generated randomly versus sets created by the SOM using the MoRSE–WHIM Dragon descriptor combination[a]

	Random sets			MoRSE–WHIM sets		
	Mean RMSE	S.D.	Mean R^2	Mean RMSE	S.D.	Mean R^2
TSET	0.57	0.02	0.75	0.58	0.005	0.74
CVSET	0.59	0.03	0.73	0.57	0.010	0.76
PSET	0.80	0.13	0.56	0.63	0.020	0.63

[a] The statistics are from a non-linear CNN model using a 6-5-1 architecture. The same descriptors were used in both models.

Table 9
RMS errors for a non-linear CNN Model[a] using a scrambled dependent variable using training, cross-validation and predictions sets created by the KSOM using the MoRSE–WHIM Dragon descriptor combination

	Scrambled		Original	
	Mean RMSE	Mean R^2	RMSE	R^2
TSET	1.04	0.17	0.58	0.74
CVSET	1.00	0.09	0.56	0.76
PSET	0.97	0.01	0.59	0.63

[a] The model was generated using a 6-5-1 CNN architecture and the ADAPT descriptors reported for the best non-linear model.

Table 10
A Summary of the RMS errors and R^2 values for 100 runs of the best CNN architecture (6-5-1) using randomly selected ADAPT descriptors[a]

	RMSE		R^2	
	Mean	S.D.	Mean	S.D.
Training set	0.81	0.09	0.47	0.11
Cross-validation set	0.84	0.08	0.36	0.13
Prediction set	0.84	0.09	0.28	0.13

[a] QSAR sets used in these models were created by the KSOM using the MoRSE–WHIM Dragon descriptor combination.

Finally a randomization test was carried out to investigate the role of chance correlations in the genetic algorithm (i.e., the descriptor selection algorithm). This was carried out by generating 100 CNN models using a 6-5-1 architecture and the QSAR sets generated by the SOM (using the MoRSE–WHIM Dragon descriptor subset). However, in each run, six ADAPT descriptors were randomly selected from the reduced pool. One would assume that the RMS errors and R^2 values for the models generated by randomly selecting descriptors would be worse than for the best reported model but not as poor compared to the runs using a scrambled dependent variable. This may be explained by noting that since the dependent variable is not scrambled there will be some correlation with the descriptors selected. However, due to the fact that we randomly select descriptors this correlation will not be as significant compared to descriptor selection using a genetic algorithm, which looks for descriptor subsets that are well correlated with the dependent variable and hence produce models with low cost functions. Thus this test ensures that the specific set of descriptors selected by the genetic algorithm did not arise by chance alone. The results for this test are provided in Table 10. As can be seen, the average RMS error for all three sets are higher than those reported for the best model, though the differences are not as significant compared to the results from the scrambled dependent variable test. The R^2 values are also lower than for the best reported model but are not as poor when compared to the results from the scrambled dependent variable test.

The results from the randomization tests described above thus indicate that chance correlations played little (if any) role in both the descriptor selection algorithm as well in the final model itself.

10. Diversity indices and SOM-generated sets

The SOM was used to prepare training and prediction sets so that the sets would be heterogeneous in nature and representative of the whole data set. The molecular data set diversity index [27] has been developed to quantify the diversity of a data set and the correspondence between training and prediction sets. This metric provides a quantitative estimate of the similarity between the training and prediction sets. Golbraikh describes three quantities—$M_{(\text{test, train})}$, $M_{(\text{train, test})}$ and I_{train}. The quantity of interest here is $M_{(\text{test, train})}$, which measures the diversity of the training set with respect to the prediction set. The value of $M_{(\text{train, test})}$ depends on both the algorithm used to generate sets as well as the distribution of the data set in the descriptor space. In general, lower values of $M_{(\text{test, train})}$ indicate that the points in the prediction set are closer (or correspond better) to the points in the training set. However, the evaluation of $M_{(\text{test, train})}$ depends on the value of an arbitrary value termed the dissimilarity level. Golbraikh does not go into detail regarding the choice of a dissimilarity level. Hence, we calculated $M_{(\text{test, train})}$ values at increasing DL values for each Dragon descriptor combination, plotted them (Fig. 8), and correlated the behavior of the plots with the CNN model statistics. One would expect that for training and prediction sets which correspond well with each other (i.e., a prediction set point corresponds to some training set point) the $M_{(\text{test, train})}$ should rapidly fall to zero with increasing DL values. However, another view would be to consider the training and prediction sets to be well distributed throughout descriptor space of the data set. In such a case the correspondence between the two sets would not necessarily be very good and one would observe higher values of $M_{(\text{test, train})}$ for a given DL value. This might lead one to conclude that such a situation would lead to bad model statistics. However, Fig. 8 indicates otherwise. From the plot we see that the curves for the MoRSE–2D autocorrelation and the MoRSE–WHIM combinations remain constant at an $M_{(\text{test, train})}$ value of 1 for all DL values up to approximately 2 and the MoRSE–GETAWAY combination remains at 1 up to nearly 2.5. From Table 3 we see that the MoRSE–GETAWAY combination has the best training and cross-validation set errors of all the sets tested, but its prediction set error is higher. At the same time, the training and prediction set errors for the MoRSE–WHIM and MoRSE–2D autocorrelation combinations are larger than for the MoRSE–GETAWAY combination—but their order follows the trend in the graph. Sets that remain at an $M_{(\text{train, test})}$ value of 1 for higher DL values appear to lead to lower RMS errors for the training and cross-validation sets.

If one considers the prediction set errors, a similar trend is seen. Sets whose $M_{(\text{test, train})}$ versus DL plots remain at an $M_{(\text{test, train})}$ value of 1 for larger values of DL appear to lead to better prediction set errors. However, this should not be considered as an absolute as the plot for the GETAWAY set does not follow this trend. In fact the prediction set error is equal to that for the MoRSE–WHIM set, but the $M_{(\text{train, test})}$ value drops below 1 for DL values of 0.7 onwards. Thus the values of $M_{(\text{test, train})}$ for the GETAWAY set are lower for a given DL value, indicating a better correspondence between the training and prediction sets. However, though this leads to a good prediction set error, the training and cross-validation set errors are quite large. This could imply that lower $M_{(\text{test, train})}$ values might lead to better prediction set errors but at the same time would lead to a loss of generalizability as evidenced by the training and cross-validation set errors.

Though the use of an arbitrary DL value in the evaluation of $M_{(\text{test, train})}$ values does make interpretation of $M_{(\text{test, train})}$

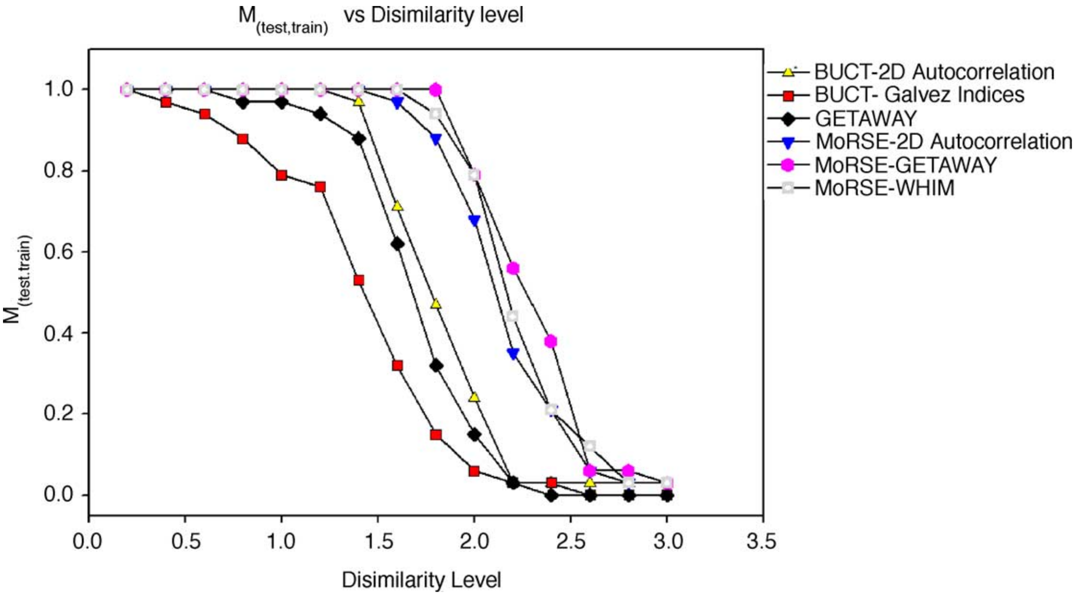


Fig. 8. Plot of dissimilarity level vs. $M_{(test,\ train)}$ for the various Dragon sets studied.

values slightly ambiguous, we feel that the technique we describe does provide some indication as to whether a training set might lead to good training and prediction set errors, based on diversity index information.

11. Conclusion

This study used a Kohonen self-organizing map to investigate whether a similarity-based set generation method would lead to better QSAR models. Multiple runs using different sets of Dragon descriptors were used to generate training, cross-validation and prediction sets, which were in turn used to create QSAR models. The best model obtained by this method did improve upon the published model in terms of model size. However, although the actual RMS errors were not significantly better than those published, they were consistent and exhibited a lower standard deviation over the three QSAR sets compared to the original results. QSAR sets were also generated using a sphere exclusion technique [19]. Models generated using these QSAR sets did not show any significant improvement in terms of statistics or model size over the published results. When compared to the models generated using QSAR sets created by the SOM we also see that there is no significant improvement in the statistics of the models generated by the sphere exclusion methods. Furthermore, the RMS errors of the three QSAR sets generated by the sphere exclusion method are not as consistent as those generated by the SOM and are similar to the standard deviations of the RMS errors of the original QSAR sets obtained by activity binning. However, the SOM does lead to models that are significantly simpler than those generated using the sphere exclusion method or activity binning (the published results). Randomization tests indicate that the models generated did not arise due to chance correlations. Furthermore, the use of the $M_{(test,\ train)}$ diversity index provides an indication of the Dragon descriptor sets ability to generate good QSAR sets which in turn lead to QSAR models. However, though the study did lead to a better model than that published, it involved a number of arbitrary decisions such as the choice of initial descriptors to submit to the SOM as well as choosing a specific SOM split out of several runs. The algorithm could be substantially improved by implementing a method to optimize the threshold value so that classification of molecules in the SOM could be automated. Another improvement would be in the choice of initial descriptors. Since this study was exploratory in nature, we restricted ourselves to certain subsets of Dragon descriptors which we deemed to be holistic in nature. That is, we chose Dragon descriptor sets that appeared to characterize the whole molecule, rather than characterizing specific molecular features. In addition, the choice of Dragon descriptors was also guided by the fact that we did not want to use ADAPT descriptors during the initial classification process. However, there remains an element of arbitrariness in the selection of Dragon descriptor sets. This need not be the case and by including more or even all Dragon descriptors (followed by a PCA to obtain the main contributing components) the initial classification might be better. In addition, though the evaluation of $M_{(test,\ train)}$ does involve an arbitrary constant, it seems that looking at the trend rather than individual values (for fixed DL values) can be used to make a decision on which Dragon sets could be used for further study.

References

[1] T. Kohonen, Self organizing maps, in: T. Kohonen, T.S. Huang, M.R. Schroeder (Eds.), Springer Series in Information Sciences, Springer, Heidelberg, 1994.

[2] J.A. Janet, R. Gutierrez, T.A. Chase, M.W. White, J.C. Sutton, Autonomous mobile robot global self localization using Kohonen and region feature neural networks, J. Robot. Syst. 14 (1997) 263.

[3] A. Naim, K.U. Ratnatunga, R.E. Griffiths, Galaxy morphology without classification: self organizing maps, Astrophys. J. Suppl. Ser. 111 (1997) 357.

[4] J. Gasteiger, J. Zupan, Neural networks in chemistry, Angew. Chem. Int. Ed. Engl. 32 (1993) 503.

[5] N.W. Daniel, I.R. Lewis, P.R. Griffiths, Interpretation of Raman spectra of nitro containing explosive materials. Part II. The implementation of neural, fuzzy and statistical models for unsupervised pattern recognition, Appl. Spectrosc. 51 (1997) 1868.

[6] Y. Vander Heyden, P. Vankeerbergghen, M. Novic, J. Zupan, D.L. Massart, The application of Kohonen neural networks to diagnose calibration problems in atomic absorption spectroscopy, Talanta 51 (2000) 455.

[7] M. Novic, J. Zupan, Investigation of infrared spectra–structure correlation using Kohonen and counterpropagation neural networks, J. Chem. Inf. Comput. Sci. 35 (1995) 454.

[8] W. Wu, B. Walczak, D.L. Massart, S. Heuerding, F. Erni, I.R. Last, K.A. Prebble, Artificial neural networks in classification of NIR spectral data: design of the training set, Chemometr. Intell. Lab. Syst. 33 (1996) 35.

[9] J. Aires-de-Sousa, M.C. Hemmer, J. Gasteiger, Prediction of ^{1}H NMR chemical shifts using neural networks, Anal. Chem. 74 (2002) 80.

[10] H. Satoh, O. Sacher, T. Nakata, L. Chen, J. Gasteiger, K. Funatsu, Classification of organic reactions: similarity of reactions base on the electronic features of oxygen atoms at the reaction sites, J. Chem. Inf. Comput. Sci. 38 (1998) 210.

[11] L. Chen, J. Gasteiger, Knowledge discovery in reaction databases: landscaping organic reactions by a self organizing neural network, J. Am. Chem. Soc. 119 (1997) 4033.

[12] D.T. Manallack, D.J. Livingstone, Neural networks in drug discovery: have they lived up to their promise? Eur. J. Med. Chem. 34 (1999) 195.

[13] I.V. Tetko, V.V. Kovalishyn, D.J. Livingstone, Volume learning algorithm artificial neural networks for 3D QSAR studies, J. Med. Chem. 44 (2001) 2411.

[14] B. Bienfait, Applications of high resolution self organizing maps to retrosynthetic and QSAR analysis, J. Chem. Inf. Comput. Sci. 34 (1994) 890.

[15] V.S. Rose, I.F. Croall, H.J.H. Macfie, An application of unsupervised neural network methodology Kohonen topology-preserving mapping to QSAR analysis, Quant. Struct.–Act. Relat. 10 (1991) 6.

[16] S. Anzali, G. Barnickel, M. Krug, J. Sadowski, M. Wagener, J. Gasteiger, J. Polanski, The comparison of geometric and electronic properties of molecular surfaces by neural networks: application to the analysis of corticosteroid-binding globulin activity of steroids, J. Comput. Aided Mol. Des. 10 (1996) 521.

[17] P. Grammatica, V. Consonni, R. Todeschini, QSAR study on the tropospheric degradation of organic compounds, Chemosphere 38 (1999) 1371.

[18] G. Espinosa, A. Arenas, F. Giralt, An integrated SOM fuzzy ARTMAP neural system for the evaluation of toxicity, J. Chem. Inf. Comput. Sci. 42 (2002) 343.

[19] A. Golbraikh, A. Tropsha, Predictive QSAR modeling based on diversity sampling of experimental data sets for the test and training set selection, J. Comp. Aid. Molec. Des. 16 (2002) 356.

[20] R. Kocjancic, J. Zupan, Modelling of the river flowrate: the influence of the training set selection, Chemometr. Intell. Lab. Syst. 54 (2000) 21.

[21] D.B. Kirew, J.R. Chretien, P. Bernard, F. Ros, Application of Kohonen neural networks in classification of biologically active compounds, SAR QSAR Environ. Res. 8 (1998) 93.

[22] A. Yan, J. Gasteiger, Prediction of aqueous solubility of organic compounds based on 3D structure representation, J. Chem. Inf. Comput. Sci. 43 (2003) 429–434.

[23] P.M. Andersson, M. Sjostrom, S. Wold, T. Lundstedt, Strategies for subset selection of parts of in an in house chemical library, J. Chemometr. 15 (2001) 353–369.

[24] A. Linusson, J. Gottfries, T. Olsson, E. Ornskov, S. Folestad, B. Norden, S. Wold, Statistical molecular design, parallel synthesis and biological evaluation of a library of thrombin inhibitors, J. Med. Chem. 44 (2001) 3424.

[25] A. Linusson, J. Gottfries, F. Lindgren, S. Wold, Statistical molecular design of building blocks for combinatorial chemistry, J. Med. Chem. 43 (2000) 1320–1328.

[26] P.M. Andersson, T. Lundstedt, Hierarchical experimental design exemplified by QSAR evaluation of a chemical library directed towards the melanocortin 4 receptor, J. Chemometr. 16 (2002) 490–496.

[27] A. Golbraikh, Molecular data set diversity indices and their applications to comparison of chemical databases and QSAR analysis, J. Chem. Inf. Comput. Sci. 40 (2000) 414.

[28] P.C. Jurs, J.T. Chou, M. Yuan, Studies of chemical structure biological activity relations using pattern recognition, in: E.C. Olsen, R.E. Christoffersen (Eds.), Computer Assisted Drug Design, American Chemical Society, Washington, DC, 1979.

[29] A.J. Stuper, W.E. Brugger, P.C. Jurs, Computer Assisted Studies of Chemical Structure and Biological Function, Wiley, New York, 1979.

[30] R. Todeschini, V. Consonni, M. Pavan, Milano Chemometrics and QSAR Research Group, DRAGON, v. 2.1, Department of Environmental Sciences, P. a della Scienza, Milano, Italy.

[31] F.R. Burden, Molecular identification number for substructure searches, J. Chem. Inf. Comput. Sci. 29 (1989) 225–227.

[32] F.R. Burden, A chemically intuitive molecular index based on the eigenvalues of a modified adjacency matrix, Quant. Struct.–Act. Relat. 16 (1997) 309–314.

[33] R.S. Pearlman, K.M. Smith, Metric validation and the receptor-relevant subspace concept, J. Chem. Inf. Comput. Sci. 39 (1998) 28–35.

[34] D.T. Stanton, Evaluation and use of BCUT descriptors in QSAR and QSPR studies, J. Chem. Inf. Comput. Sci. 39 (1999) 11–20.

[35] P. Broto, G. Moreau, C. Vandicke, Eur. J. Med. Chem. 19 (1984) 66–70.

[36] G. Moreau, P. Broto, Nouv. J. Chim. 4 (1980) 359–360.

[37] G. Moreau, P. Broto, M. Fortin, C. Turpin, Computer conducted screening of molecular structures of potentially anxiolytic substances using an autocorrelation technique, Eur. J. Med. Chem. 23 (1988) 275–281.

[38] P.A.P. Moran, Notes on continuous stochastic phenomena, Biometrika 37 (1950) 17–23.

[39] R.C. Geary, The contiguity ratio and statistical mapping, Incorp. Stat. 5 (1954) 115–145.

[40] J. Galvez, R. Garcia, M.T. Salabert, R. Soler, Charge indexes. New topological descriptors, J. Chem. Inf. Comput. Sci. 34 (1994) 520–525.

[41] J. Galvez, R. Garcia-Domenech, C. de Gregorio Alapont, V. De Julian Ortiz, L. Popa, Pharmacological distribution diagrams: a tool for de novo drug design, J. Mol. Graph. Model. 14 (1996) 272–276.

[42] J. Galvez, R. Garcia-Domenech, V. De Julian Ortiz, R. Soler, Topological approach to drug design, J. Chem. Inf. Comput. Sci. 35 (1995) 272–284.

[43] V. Consonni, R. Todeschini, in: H.D. Holtje, W. Sippl (Eds.), Rational Approaches to Drug Design, Prous Science, Barcelona, 2001.

[44] V. Consonni, R. Todeschini, M. Pavan, Structure/response correlations and similarity/diversity analysis by GETAWAY descriptors. 1. Theory of the novel 3D molecular descriptors, J. Chem. Inf. Comput. Sci. 42 (2002) 682–692.

[45] V. Consonni, R. Todeschini, M. Pavan, P. Grammatica, Structure/response correlations and similarity/diversity analysis by GETAWAY descriptors. 2. Application of the novel 3D molecular descriptors to QSAR/QSPR studies, J. Chem. Inf. Comput. Sci. 42 (2002) 693–705.

[46] J.H. Schuur, P. Selzer, J. Gasteiger, The coding of the three-dimensional structure of molecules by molecular transforms and its application to structure-spectra correlations and studies of biological activity, J. Chem. Inf. Comput. Sci. 36 (1996) 334–344.

[47] J. Gasteiger, J. Sadowski, P. Selzer, L. Steinhauer, V. Steinhauer, Chemical information in 3D space, J. Chem. Inf. Comput. Sci. 36 (1996) 1030–1037.

[48] R. Todeschini, M. Lasagni, E. Marengo, New molecular descriptors for 2D and 3D structures—theory, J. Chemometr. 8 (1994) 263–273.

[49] R. Todeschini, P. Grammatica, E. Marengo, R. Provenzani, J. Chemosphere 33 (1995) 71–79.

[50] R. Todeschini, P. Gramatica, 3D-modelling and prediction by WHIM descriptors. 5. Theory development and chemical meaning of WHIM descriptors, Quant. Struct.–Act. Relat. 16 (1997) 113–119.

[51] R. Todeschini, P. Grammatica, Quant. Struct.–Act. Relat. 16 (1997) 120–125.

[52] P. Grammatica, M. Corradi, V. Consonni, Chemosphere 41 (2000) 763–777.

[53] R. Todeschini, P. Grammatica, E. Marengo, R. Provenzani, Modeling and prediction by using WHIM descriptors in QSAR studies: submitochondrial particles (SMP) as toxicity biosensors of chlorophenols, Chemosphere 33 (1996) 71–79.

[54] R. Todeschini, M. Vighi, R. Provenzani, A. Finzio, P. Grammatica, Modeling and prediction by using whim descriptors in QSAR studies: toxicity of heterogeneous chemicals on Daphnia Magna, Chemosphere 32 (1996) 1527–1545.

[55] B.E. Mattioni, P.C. Jurs, Prediction of dihydrofolate reductase inhibition and selectivity using computational neural networks and linear discriminant analysis, J. Mol. Graph. Model. 21 (2003) 391–419.

[56] L.B. Kier, L.H. Hall, Molecular connectivity. VII. Specific treatment to heteroatoms, J. Pharm. Sci. 65 (1976) 1806–1809.

[57] L.B. Kier, L.H. Hall, Molecular Connectivity in Structure–Activity Analysis, Research Studies Press Ltd., Wiley, London, 1986.

[58] L.B. Kier, L.W. Hall, Molecular connectivity. I. Relationship to local anaesthesia, J. Pharm. Sci. 64 (1975) 1971–1974.

[59] L.B. Kier, L.H. Hall, Molecular Connectivity in Chemistry and Drug Research, Academic Press, New York, 1976.

[60] A.T. Balaban, Highly discriminating distance based topological index, Chem. Phys. Lett. 89 (1982) 399–404.

[61] D.T. Stanton, P.C. Jurs, Development and use of charged partial surface area structural descriptors in computer assisted quantitative structure property relationship studies, Anal. Chem. 62 (1990) 2323–2329.

[62] S. Liu, C. Cao, Z. Li, Approach to estimation and prediction for normal boiling point (NBP) of alkanes based on a novel molecular distance edge (MDE) vector, lambda, J. Chem. Inf. Comput. Sci. 38 (1998) 387–394.

[63] M. Randic, On molecular identification numbers, J. Chem. Inf. Comput. Sci. 24 (1984) 164–275.