JOURNAL OF PHYSICAL ORGANIC CHEMISTRY
J. Phys. Org. Chem. 2003; **16:** 232–245
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/poc.597

Review Commentary

Neural networks as data mining tools in drug design

Johann Gasteiger,* Andreas Teckentrup, Lothar Terfloth and Simon Spycher

Computer-Chemie-Centrum und Institut für Organische Chemie, Universität Erlangen-Nürnberg, Nägelsbachstrasse 25, D-91052 Erlangen, Germany

Received 26 July 2002; revised 27 November 2002; accepted 28 November 2002

ABSTRACT: Neural networks are powerful data mining tools with a wide range of applications in drug design. This paper largely concentrates on self-organizing neural networks that can be used for investigating datasets both by unsupervised and by supervised learning. The representation of chemical structures is the key to success in establishing useful relationships. Applications are shown for exploring different structure representations, for establishing quantitative structure–activity relationships and for handling compounds having multicategory activities. The applications comprise the separation of compounds according to different biological activities, the location of biologically active compounds in large chemical spaces, the analysis of high-throughput screening data and the classification of compounds according to mode of toxic action. Copyright © 2003 John Wiley & Sons, Ltd.

KEYWORDS: self-organizing neural networks; Kohonen neural network; counterpropagation networks; chemical structure representation; 3D structure generation; library screening; biological activity prediction

INTRODUCTION

Data mining

In recent years, the term 'data mining' has come into widespread use. While different people might have different definitions, the overall objective of data mining is clear: extract knowledge from a large set of data in order to make predictions of new events. In this context, clear definitions of the terms 'data,' 'information' and 'knowledge' seem necessary. Again, different people might have different ideas on defining these terms; however, the following definitions have found widespread acceptance. Data become information when they are put into a context; generalization of information can lead to knowledge. Figure 1 illustrates this process for an important task in drug design: the establishment of relationships between chemical structure and biological activity.[1]

Biological activity data are in most cases useless as long as we do not know the chemical structure of the compound that exerts this biological activity. Only when we combine these two pieces of information, chemical structure and biological activity, do we obtain information. A set of such pairs of structure–activity data can be analyzed by some learning method to find the inherent relationship, to obtain knowledge on the structural requirements for biological activity.

Data mining is thus an inductive learning method because a series of experimental observations are used to arrive at some general insight. Data mining is nothing new; in fact, it is the predominant learning method in many scientific disciplines.

Chemistry in particular has built its scientific knowledge on inductive learning. With the advent of databases and the world wide web that contain data in electronic form, algorithmic learning methods have increasingly been used, and thus data mining in its more restricted sense by electronic means has gained increasing interest.

The range of inductive learning methods is wide: statistical and pattern recognition methods, or neural networks. In this discussion we will limit ourselves to neural networks.

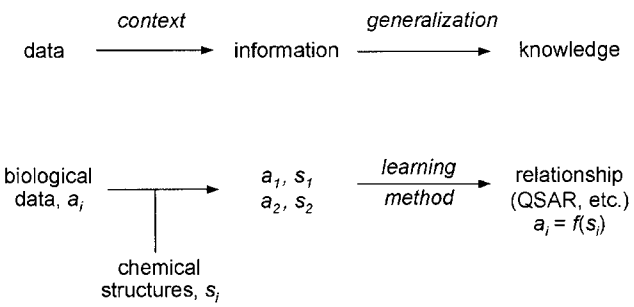

Figure 1. From data through information to knowledge

**Correspondence to:* J. Gasteiger, Computer-Chemie-Centrum und Institut für Organische Chemie, Universität Erlangen-Nürnberg, Nägelsbachstrasse 25, D-91052 Erlangen, Germany.
E-mail: gasteiger@chemie.uni-erlangen.de
Contract/grant sponsor: Bundesministerium für Bildung und Forschung (BMBF).
Contract/grant sponsor: Verband der Chemischen Industrie.

Drug design

A variety of tasks in drug design involve the analysis and processing of many individual data:

- comparison of combinatorial libraries
- search for new lead structures
- establishment of structure–activity relationships
- analysis of high-throughput screening data
- optimization of a lead structure
- exploration of conformational flexibility
- analysis of ADME (absorption, distribution, metabolism and excretion) data.

All these tasks can benefit from the use of data mining methods.

NEURAL NETWORKS

Artificial neural networks have been designed to model the information processing in the human brain. The high flexibility of the brain and the wide range of tasks performed by the brain have led to the development of numerous different artificial neural network models.[2] A variety of books deal with the different neural network methods and their application to chemistry and drug design.[3]

To highlight the importance of neural networks for analyzing chemical data, it may suffice to mention that each year more than 1000 publications appear on the use of neural networks in chemistry.

The following characteristics render neural networks so attractive for analyzing chemical data:

- Complex relationships are implicitly put down in the weights of the network; thus, no explicit mathematical form of such a relationship has to be given.
- Both linear and non-linear relationships can be modelled.
- Two steps are involved in modeling relationships, training and prediction. Training of a neural network is usually fairly rapid even with large data sets, while prediction is nearly instantaneous.
- By adding new data, a trained neural network can be further refined; training does not have to start from the very beginning again.

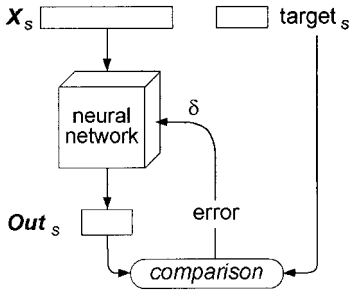

Figure 2. Outline of supervised training of a neural network. The output Out_s of the neural network for the object X_s is compared with the target values belonging to this object and influences the adjustment of the weights of the neural network

- Both unsupervised and supervised learning methods are available.

An essential decision in the use of neural networks is whether an unsupervised or a supervised learning method is chosen.[3] In fact, we recommend that any initial investigations of a data set should first be made with an unsupervised neural network. Before discussing this in detail, a brief general presentation of the two learning methods is given. In this discussion we will consider the neural network as a black box containing the weights to be adjusted without going into the details of the weight adjusting algorithm.

Figure 2 shows the essential outline of *supervised* learning. For supervised learning a series of data pairs consisting of objects and the corresponding target values of physical, chemical or biological data characterizing these objects should be given. An object represented by some descriptors is input into the neural network initially containing randomly assigned weights. With random weights the output of the neural network is necessarily wrong. Comparison of the obtained output with the desired target value of the input object provides the error of prediction, which is fed back to the neural network to adjust the weights in order to minimize the output error. This process is iteratively repeated with a series of objects and their target values until the output error has fallen below a predefined threshold.

Such a trained neural network can then be used to make predictions by inputting new objects. The descriptors of the new object are combined with the weights in

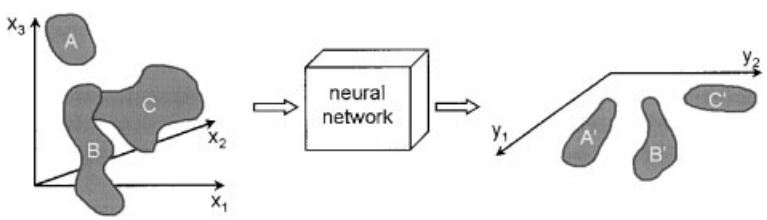

Figure 3. Essentials of unsupervised projection of a dataset. The objects from a high-dimensional space are projected to a lower-dimensional space, typically a 2D plane

the network to calculate an output value (or output values if more than one target value is used in the training of the network).

The most widely used supervised neural network learning algorithm is back-propagation learning, which is usually applied in a multilayer feed-forward network.[4]

In *unsupervised* learning, the representations of objects are investigated without using the property to be studied in the training of the neural network. One way of doing this is to use a neural network as a projection method. We can consider the descriptors used to represent the objects as coordinates of a space. An object, such as a molecule, is then a point in this multi-dimensional space. The objects may be found in various clusters in the high-dimensional space (Fig. 3).

An unsupervised neural network is then used to project the points from the high-dimensional space into a space with a smaller number of dimensions such as a two-dimensional plane. The purpose of this projection is to preserve as well as possible the topology of the high-dimensional space, such as the clustering of the objects, after projection into the low-dimensional space.

The clusters may be associated with different types of properties (different biological activities) which can then be identified after projection. It should be emphasized, however, that no knowledge on such properties (activities) is used in determining the projection by training the neural network. This is the essence of unsupervised learning.

It should also be mentioned that the human brain performs such projections by generating sensory maps of the environment in the visual, auditory, or somatosensory cortex. One such unsupervised learning method is the self-organizing neural network introduced by Kohonen.[5] This method will be explained in more detail in the next section.

Self-organizing neural network

The self-organizing neural network was introduced by Teuvo Kohonen nearly 20 years ago.[5] It is that neural network which probably has the closest analogy to some of the information processing in the brain, particularly as concerns the generation of sensory maps. The neurons of a Kohonen network are usually arranged in a two-dimensional layer, each neuron containing m weights. In fact, the neurons contain as many weights as the objects that are sent into a Kohonen network have descriptors. Figure 4 illustrates the architecture of a Kohonen network.

An object, a sample, s, represented by m descriptors, x_{si}, is sent into a two-dimensional network of neurons, each neuron, j, having m weights, w_{ji}. An object will be mapped into that neuron, c, that has weights most similar to the descriptors of the input object [Eqn. (1)]. This

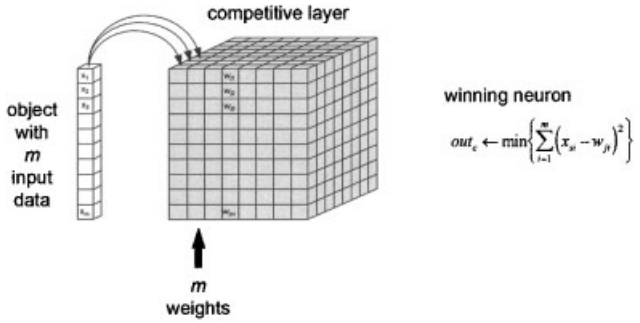

Figure 4. Architecture of a Kohonen network. Here, a network with 8×8 neurons is depicted. The first subscripted index refers to the number of the neuron. During training, the m-dimensional input patterns iteratively enter the network and the responses of all neurons are calculated. The winning neuron having the minimum distance to the input pattern is determined

neuron is called the central, or winning neuron.

$$out_c \leftarrow \min\left\{\sum_{i=1}^{m}(x_{si} - w_{ji})^2\right\} \qquad (1)$$

First, the weights of the network are randomly initialized. A weight adaption algorithm is then invoked as given by the equation

$$w_{ji}^{new} = w_{ji}^{old} + \eta(t) \cdot a(d_c - d_j)(x_i - w_{ji}^{old}) \qquad (2)$$

where w_{ji}^{new} and w_{ji}^{old} are the new and old weights, respectively, η is the learning rate, dependent on time or iteration cycle, t, $d_c - d_j$ is the topological distance between the central neuron c and the current neuron j, and $a(\cdot)$ is a topology-dependent scaling function. A new object input into a Kohonen network will be mapped into a neuron whose distance from the previously winning neuron is dependent on the similarity of the two objects. If the two objects have very similar descriptors they will be mapped into the same or closely adjacent neurons. Thus, by training a Kohonen network iteratively with a dataset of objects, a mapping of the objects into a two-dimensional space will be obtained that reflects the topology, the arrangement of the objects in the m-dimensional space.

One of the most important advantages of a Kohonen network is that it can generate maps and, thus, visualize relationships of objects. This can be used for similarity perception and for clustering. In this context, it is a major advantage that on expansion of a dataset by addition of new data training does not have to start from the very beginning. Rather, the trained network can be used, the new data are input and thus training is taken up a few times until the network has adjusted to the new data.

A two-dimensional arrangement of neurons can have two different types of topologies: a rectangular topology where the neurons in the center have eight immediate

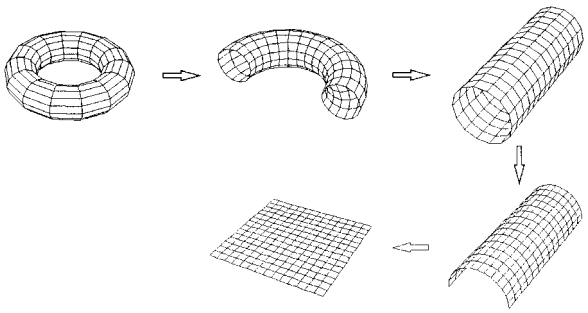

Figure 5. Conversion of the surface of a torus into a plane by disecting the torus by two perpendicular planes

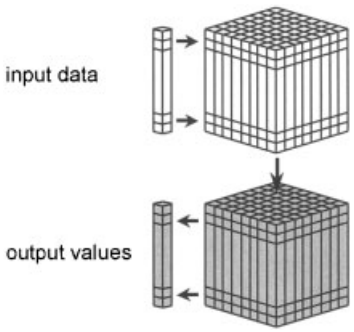

Figure 6. Architecture of a counterpropagation network. In addition to a Kohonen neural network, counterpropagation networks also have an output layer. To determine the winning neuron, only the input layer is taken into account. In the adaption step, the weights in the input and output layer are corrected

neighboring neurons, whereas neurons on the border have only five or even only three neighbors (for the neurons in the four corners). In order to provide each neuron with the same number of immediate neighbors, a toroidal topology has to be chosen. In this case, the neurons are arranged on the surface of a torus. For better visualization of the mapping obtained when using such a toroidal Kohonen network, the torus is cut along two arbitrary, perpendicular lines and the surface of the torus spread into a plane (Fig. 5). It should then be remembered that the neurons on the extreme left-hand side are to be taken as neighbors of the neurons on the extreme right-hand side. An analogous situation exists between the neurons on the uppermost and those of the lowermost line.

Here, we refer to two different types of applications of Kohonen networks: *similarity perception* or *interpolation*. Which type of application is invoked is mainly determined by the ratio of the size of the network as compared to the size of the dataset.

If the number of neurons is taken smaller than the number of objects in the dataset, several objects are forced into the same neuron, thus allowing one to perceive their *similarity*. The less neurons there are, the more objects are forced into the same neuron. With continuing decrease in the size of the network, however, the danger of collisions, of forcing objects belonging to different classes into the same neuron, increases.

If the number of neurons of the network is, however, chosen to be larger than the number of objects in the dataset, then the objects are distributed over the network with a number of neurons not receiving any objects. These neurons are, however, not really empty, but contain weights that correspond to (new) objects. Such networks can be chosen for interpolating data. For example, we have chosen such an approach for the simulation of infrared spectra, interpolating new infrared spectra from those that have been used for training the network.[6]

Counterpropagation networks

It has been emphasized that Kohonen networks are based on unsupervised learning; the property of interest is not used during the training of the network. The basic learning algorithm can, however, also be used for supervised learning, for finding a relationship between the objects and one or more properties of these objects. To do this, simply the architecture of the networks has to be extended; the Kohonen network shown in Fig. 4 has to be extended by as many layers as there are properties of the objects to be studied. The architecture then consists of an input block for representing the objects and an output block containing the properties of the objects (Fig. 6). This type of neural network is called a counterpropagation network.[7]

The properties of an object are stored in the lower block at exactly the same position as the object is stored in the upper block.

The self-organizing neural network for information analysis: SONNIA

We have implemented a Kohonen neural network that particularly emphasizes the visualization aspect of such a network by having included several powerful graphic tools for visualizing chemical data. In order to show the power of a Kohonen network in visualizing the distribution of chemical objects we take a rather trivial dataset consisting of 11 aromatic hydrocarbons, 14 aliphatic hydrocarbons and 11 alcohols and phenols. These structures were initially represented by a 25-dimensional descriptor (the molecular dipole moment and an autocorrelation vector for both the partial atomic charges and the atom polarizability with 12 components each).[8]

This set of descriptors was reduced by a genetic algorithm to seven descriptors. The dataset of 36 compounds was sent into a Kohonen network of 6×3 neurons having toroidal topology. Different drop-down windows are provided for setting the topology and size of the network as well as controlling various parameters of

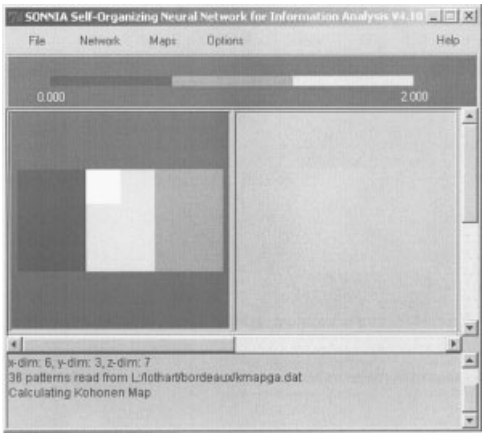

Figure 7. Kohonen network showing the distribution of aliphatic hydrocarbons (dark gray), alcohols and phenols (light gray) and aromatic hydrocarbons (gray)

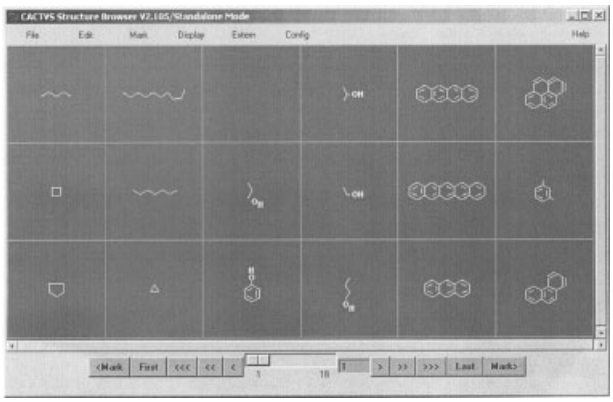

Figure 8. Visualization of the mapping of molecules into each neuron of the network shown in Fig. 7

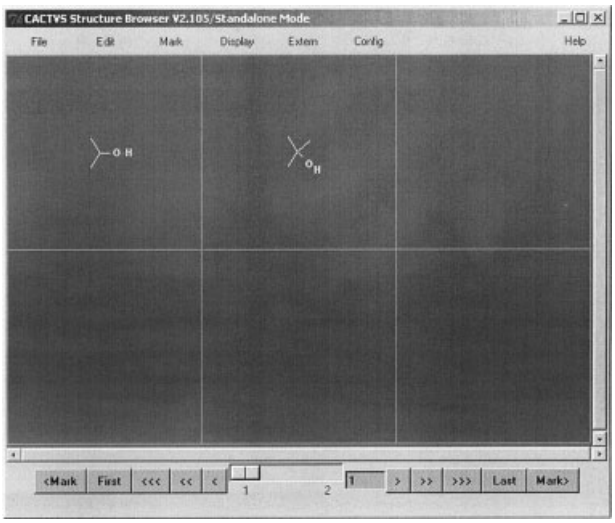

Figure 9. The two compounds mapped into neuron 0,4 of the network shown in Fig. 7

the learning procedure. After training of the network the results were visualized by assigning a dark gray color to the aliphatic hydrocarbons, a light gray color to the alcohols and phenols and a gray color to the aromatic hydrocarbons. The resulting map is shown in Fig. 7; one neuron did not receive any compound at all and is therefore colored white.

As can be seen, the compounds cluster according to their class memberships. (It should again be emphasized that the clustering of these structures according to compound class is an easy task; it was chosen for illustration purposes.)

In order to show the distribution of the compounds in the network, an option is incorporated into SONNIA that allows one to visualize that compound (the centroid) which most closely corresponds to the weights of each neuron. Figure 8 shows the result obtained.

Furthermore, all the compounds mapped into an individual neuron can also be shown. Thus, by clicking on the neuron at position 0,4 Fig. 9 is obtained, showing that both 2-propanol and *tert*-butanol are mapped into this neuron, emphasizing that the neural network has recognized the close similarity of these two compounds.

After this admittedly simple, illustrative case, we show the results obtained in a more relevant investigation. The dataset consisted of 299 compounds comprising 75 5-hydroxytryptamine 5-HT_{1a}-receptor agonists, 75 histamine H_2-receptor agonists, 75 thrombin inhibitors and 74 monoamine oxidase MAO_A inhibitors. The compounds were represented by a 128-dimensional vector obtained from an atomic radial distribution function $g(r)$ [see Eqn. (3)] using $a_i = a_j = 1$ in order to represent the molecular graph.[9] The 3D structures were obtained from the 3D structure generator CORINA.[10]

$$g(r) = \sum_{i=1}^{N-1} \sum_{j>i}^{N} a_i a_j \exp\left[-B(r - r_{ij})^2\right] \qquad (3)$$

where a_i and a_j are atomic properties (e.g. partial atomic charges obtained by the PEOE algorithm),[11] r_{ij} is the distance between atoms i and j in 3D space, r is a running variable, B is a so-called temperature factor and N is the number of atoms in the molecule. The dataset was sent into a Kohonen network consisting of 15×15 neurons having rectangular topology. The progress of training can be monitored by plotting the error after each presentation of a new object to the network (a cycle). Figure 10 shows the development of the error over 2990 cycles corresponding to sending the entire dataset of 299 compounds 10 times through the network (10 epochs).

After these 10 epochs, training was stopped and the result of mapping the 299 compounds into 225 neurons was visualized by assigning the compounds to the four different classes of agonists or inhibitors. It should again be emphasized that this information was not used during training, the training was unsupervised. Each neuron was

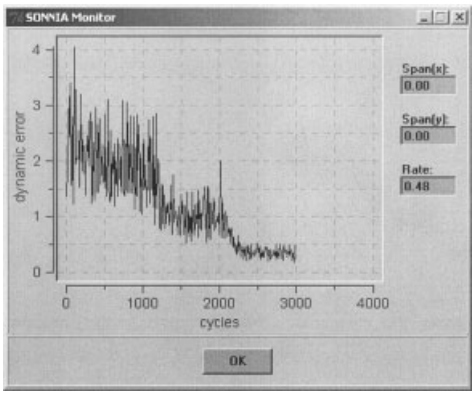

Figure 10. Development of the error during training of the network

assigned a symbol that corresponded to the majority of compounds mapped into this neuron: thrombin inhibitors with an ×, 5-HT_{1a}-receptor agonists with a circle, MAO_A inhibitors with a diamond and H_2-receptor agonists with a cross. Neurons that did not receive a compound at all were assigned a white color.

Figure 11 shows on the left-hand side the map that was obtained. An option is provided to indicate those neurons that contain compounds of different classes (collisions). This is indicated in the map in Fig. 11 on the right-hand side by coloring the corresponding neurons (e.g. in black).

The maps show that a fairly reasonable separation of the four classes of compounds could be achieved—recall that a 128-dimensional space was projected into two dimensions!—with only a few (four) neurons with collisions. Again, clicking on a specific neuron shows

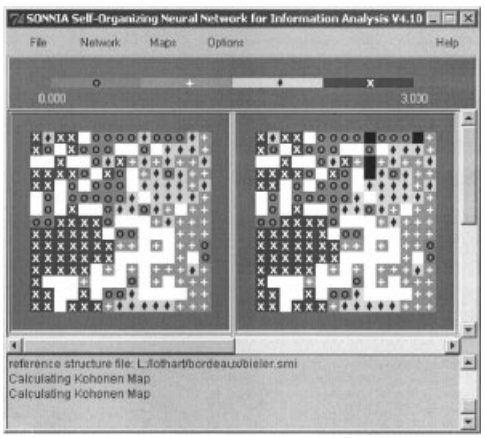

Figure 11. Mapping of a dataset of 299 compounds consisting of thrombin inhibitors (dark gray with a white ×), 5-HT_{1a}-receptor agonists (gray with a circle), MAO_A inhibitors (light gray with a diamond) and H_2-receptor agonists (gray with a cross). White neurons indicate those that did not receive any structure at all, black indicates neurons with collisions of molecules from different activity classes

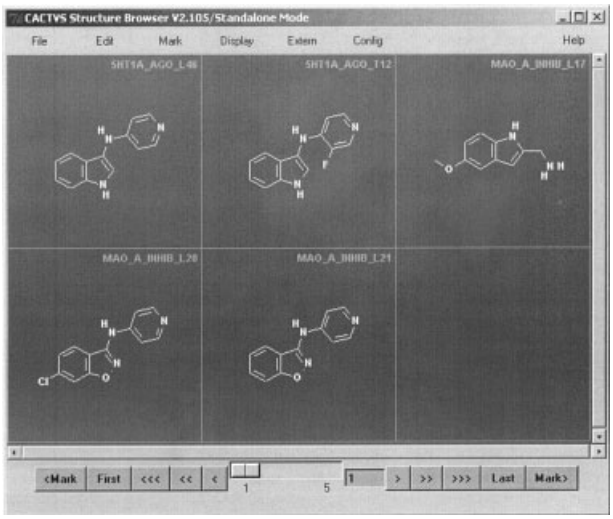

Figure 12. Compounds that were mapped into neuron 1,14 of the network shown in Fig. 11

all the compounds that are projected into this neuron. This is done in Fig. 12 with neuron 1,14 of Fig. 11 showing a neuron with collisions containing both MAO_A inhibitors and 5-HT_{1a}-receptor agonists.

STRUCTURE REPRESENTATION

The objects to be dealt with in drug design are many-fold: genes, proteins, small molecules, chemical reactions, metabolic pathways, etc. Here, we will focus only on small molecules and consider the problem of their representation in order to be able to input them into neural networks.

If a dataset of objects is investigated by a learning method such as a statistical or pattern recognition method, or a neural network, the objects have to be represented by the same number of descriptors or variables. This can clearly be seen from Fig. 4, where each object is represented by a vector of the same length, consisting of *m* descriptors.

In most cases, a dataset of compounds will consist of molecules of different sizes and of different numbers of atoms. Thus, if one wants to represent the structure of a molecule, a mathematical transformation has to be performed in order that molecules with different numbers of atoms end up with the same number of descriptors.

Furthermore, various degrees of sophistication can be chosen for structure representation, from the constitution through the 3D structure to molecular surfaces. We have developed a series of methods and corresponding software packages to automatically derive 3D structures or molecular surfaces from the constitution of a molecule as represented by a connection table.

Furthermore, several methods have been developed to calculate fundamental physicochemical properties of the

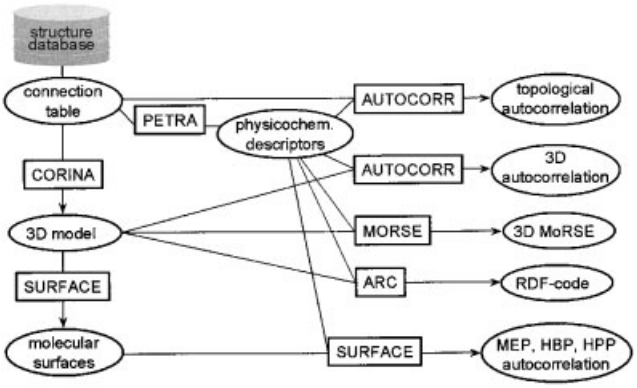

Figure 13. Overview of software packages and how they can be combined to end up with different structure representations

atoms and bonds in a molecule. These physicochemical properties can be combined either with the constitution, the 3D structure or molecular surfaces. Then, these structure representations are submitted to mathematical transformations such as autocorrelation or atom radial distribution functions to fulfill the requirement of a fixed number of descriptors irrespective of the size of a molecule.

Figure 13 gives an overview of these software tools and how they can be combined to end up with different structure representations.

We cannot go into the details of structure representation here, and have to restrict ourselves to throwing a few spotlights on the methods that we have developed and how they can be combined. Some more insight will be provided by the applications in the following sections.

The kind of structure representation to be chosen will be dictated by the problem that has to be studied. Usually, various structure representations will have to be explored before the one working best for the given problem will be found. Furthermore, the size of the dataset will be of influence: in order to keep computation times within reasonable limits, less sophisticated structure representations will have to be chosen for very large datasets. For datasets with many millions of structures one will probably have to stay with topological representations, leaving the representation of molecular surfaces to situations where the number of compounds is smaller. However, the methods that we have developed are so rapid that surface properties have been calculated with datasets of a few hundred thousand molecules.[8]

In the following we will only briefly mention a few of the systems indicated in the overview of Fig. 13.

3D Structure generation: CORINA

CORINA (COoRdINAtes) rapidly generates a 3D molecular model when given information on the constitution of a molecule as represented in a connection table such as an SDFile or a SMILES string. CORINA is a data and rule based system that is applicable to the entire range of organic chemistry and also to many organometallic structures.[10]

Table 1. Results of the conversion of the NCI database into 3D structures using CORINA

Conversion of the database of the National Cancer Institute:	
Number of structures (October 1999)	249 081
Conversion rate	99.5%
CPU time (Pentium III/600 MHz)	9600 s, 0.04 s/molecule

Table 1 gives the results for converting the entire open database of the National Cancer Institute in a single run into 3D models.

CORINA can be accessed on the Internet at http://www2.chemie.uni-erlangen.de/services/3d.html. Up to 1000 structure can be converted free of charge. For commercial applications, CORINA is distributed by Molecular Networks.[12]

Calculation of physicochemical effects: PETRA

PETRA (Parameter Estimation for the Treatment of Reactivity Applications) collects a series of methods for the calculation of all-important electronic and energy effects in organic molecules such as heats of formation, bond dissociation energies, charge distribution, quantitative measures of the inductive, resonance or polarizability effects, etc.[13] These methods are all empirical in nature in order to have them rapid enough to be able to calculate large datasets. Most of the methods have been published previously by our group. Table 2 gives an overview of the kind of properties that can be calculated for the atoms, bonds or entire molecule.

A molecular transformation: autocorrelation

The problem of representing molecules having different

Table 2. Atomic, bond and molecular properties that can be calculated by PETRA

Atomic properties—	
Charges	q_σ, q_π, q_{tot}
Electronegativities	χ_σ, χ_π, χ_{LP}
Polarizabilities	α_i
Bond properties—	
Charge differences	Δq
Electronegativity differences	$\Delta\chi$
Bond polarizabilities	α_b
Stabilization by delocalization	D^+, D^-
Bond dissociation energies	*BDE*
Molecular properties—	
Heats of formation	ΔH_f°
Mean molecular polarizabilities	$\bar{\alpha}$

numbers of atoms with the same number of descriptors has already been mentioned. One mathematical transformation that produces a fixed, preset number of descriptors for a molecule is autocorrelation, as given by the equations

$$A = [A(d_1),\ \ldots, A(d_l)] \tag{4}$$

where

$$A(d_k) = \sum_{i=1}^{n} \sum_{j=1}^{n} p(i)\ p(j)\delta\ (n_{ij}, d_k)$$

$$\text{and } \delta(n_{ij}, d_k) = \begin{cases} 1, if\ n_{ij} = d_k \\ 0, else \end{cases} \tag{5}$$

In Eqn. (4), l gives the dimension of the autocorrelation vector. In Eqn. (5), d_k gives the number of bonds for which the autocorrelation is calculated and n_{ij} is the number of bonds between atom i and atom j; p is a property such as the partial charge on atoms i and j or on two points of the molecular surface. The usage of different atomic properties p such as mass, charges, polarizability or electronegativities allows the consideration of a broad range of physicochemical effects. n is the number of atoms in the molecule. The equation for 3D autocorrelation is according to Eqns (4) and (5) but the distance is then the actual three-dimensional distance between two atoms. Accordingly, d_k is then an interval and not a discrete value. The use of autocorrelation in molecular structure representation has a long history.[8,14,15]

Having briefly presented methods to be used for the representation of molecular structures, we are now set to apply the powerful tools of Kohonen and Counterpropagation networks to problems encountered in drug design. As we can only give a few examples here, we will focus our discussion to typical problems and use them to give general guidelines.

SEPARATING MOLECULES OF DIFFERENT BIOLOGICAL ACTIVITY—COLLECTING DIFFERENT STRUCTURE REPRESENTATIONS

The task of the study presented here was to separate a dataset of 172 molecules into benzodiazepine agonists (60 compounds) and dopamine agonists (112 compounds).[16] Figure 14 shows some of the structures contained in the dataset emphasizing the problem mentioned in the previous sections: having to represent molecules with different numbers of atoms with the same numbers of descriptors.

The structures were represented by topological autocorrelation. Thus, d in Eqn. (4) was the number of bonds between the two atoms i and j; d was kept running from 2 to 8 (seven distances altogether). As a first shot, as no specific requirements of the receptor were available, we

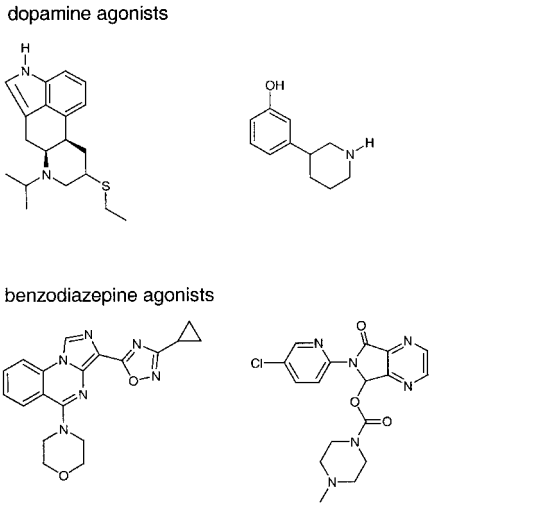

Figure 14. Representative 2D structures of dopamine and benzodiazepine agonists

decided to use a rather broad structure representation, including a variety of physicochemical effects in the autocorrelation vector. Separate seven-dimensional vectors (seven distance intervals!) were constructed with $\sigma-$ atomic charges, $(\sigma+\pi)$ atomic charges, $\sigma-$ electronegativity, $\pi-$ electronegativity, lone pair electronegativity, atomic polarizability and an atomic property of 1 (to just represent the molecular graph). These seven autocorrelation vectors were then concatenated to give a 49-dimensional representation of the 172 molecules in the dataset. Training a 10×7 Kohonen network with the entire dataset gave a map that was then marked by assigning colors to the network depending on whether a neuron contains a dopamine or a benzodiazepine agonist (Fig. 15).[16]

As can be seen, the two sets of molecules separate fairly well. This is even more remarkable as the class membership was not used in training the network but only in visualizing the results of training (unsupervised learning!). This attests to the relevance of the chosen structure representation for reproducing effects that are responsible for the different binding of dopamine and benzodiazepine agonists.

Next we turned our attention to the question on whether we still can see the separation of the two sets of

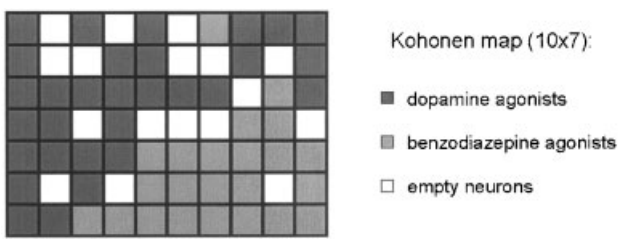

Figure 15. Kohonen map obtained from a dataset of 112 dopamine and 60 benzodiazepine agonists

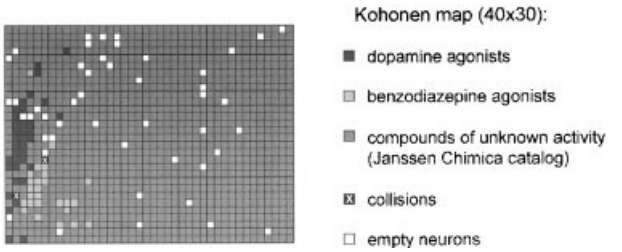

Figure 16. Kohonen map of a dataset consisting of the dopamine and benzodiazepine agonists of Fig. 17 and 8323 compounds in a chemical supplier's catalog

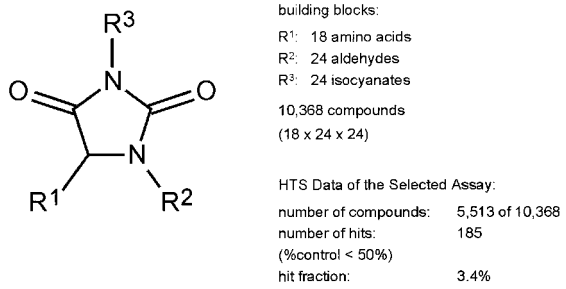

Figure 18. Hydantoin library and results from HTS assay

molecules when they are buried in a large dataset of diverse structures. For this purpose we added this dataset of 172 molecules to the entire catalog of 8223 compounds available from a chemical supplier (Janssen Chimica). Now having a larger dataset one has to also increase the size of the network, and a network of 40×30 neurons was chosen. Training this network with the same 49-dimensional structure representation as described, previously but now for all 8395 structures provided the map shown in Fig. 16.

Even in this fairly diverse dataset of structures, the dopamine and benzodiazepine agonists could be separated fairly well; only two neurons had collisions between these two types of compounds. What is even more important, however, is that we now know in which chemical space one would have to search for new lead structures for dopamine or for benzodiazepine agonists.

To illustrate this point, Fig. 17 shows the contents of the neuron at position 3,9. This neuron obtained two dopamine agonists and three compounds from the Janssen Chimica catalog of unknown biological activity which might be taken as lead structures for developing dopamine agonists.

The results presented here imply that a similar approach can be used for comparing two different libraries, for determining the degree of overlap between the compounds in these two libraries.

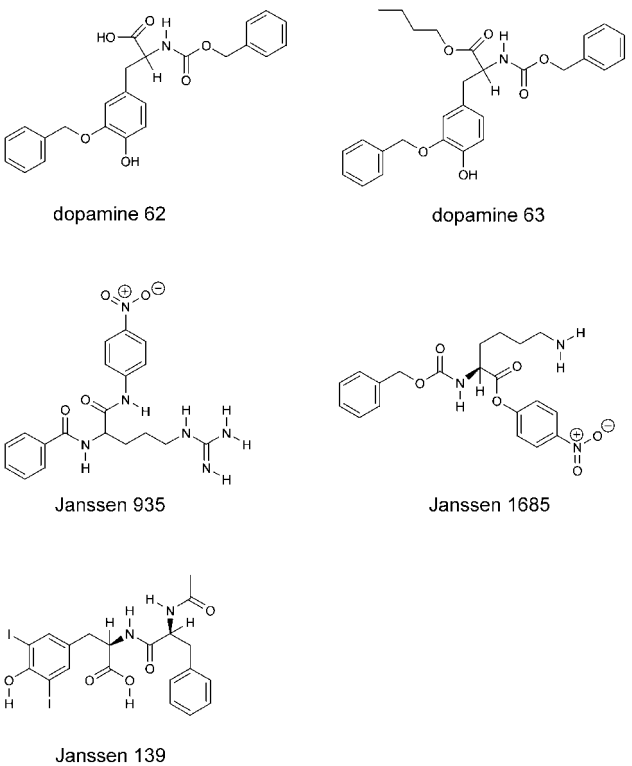

Figure 17. Structures that were mapped into neuron at position 3,9 of the Kohonen map of Fig. 18

DEVELOPING A SCREEN FOR A VIRTUAL LIBRARY—EXPLORING DIFFERENT STRUCTURE REPRESENTATIONS

The next study concerns the development of a screen to separate hits from non-hits of a combinatorial library. In order to achieve this, one needs the results of the testing of the compounds of a combinatorial chemistry experiment in an assay used in high-throughput screening. The library that was investigated consisted of hydantoins synthesized from 18 aldehydes, 24 amino acids and 24 isocyanates (Fig. 18). This produced $18 \times 24 \times 24 = 10368$ hydantoins, of which 5328 compounds were tested in a specific assay; 185 compounds (3.5%) turned out to be hits in this assay testing.

The task was then to use this information to develop a filter that can separate hits from non-hits and, thus, could be employed in the screening of a virtual library of hydantoins. In order to achieve this, six different structure representations were explored: Daylight fingerprints of three different lengths (256, 512 and 1024 dimensions) and three autocorrelation vectors of molecular surface properties [molecular electrostatic potential (ESP), hydrogen bonding potential (HBP) and hydrophobicity potential (HPP)]. In this case, Eqn. (4) is used in such a way that a_i and a_j are properties of points on the molecular surface (ESP, HBP or HPP); d is now the distance of these two points on the molecular surface with the products of all distances in a certain range (e.g. between 1.5 and 2.0 Å) collected at the same position [in

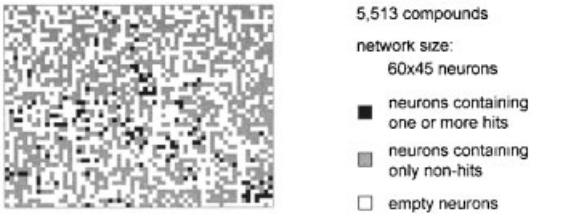

Figure 19. Clustering of the hydantoin library by a Kohonen network using 256 Daylight fingerprints: hits in neurons colored black and non-hits in neurons colored gray

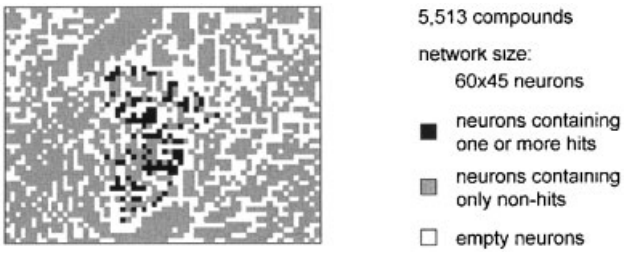

Figure 20. Clustering of the hydantoin library by a Kohonen network using a 26-dimensional autocorrelation vector obtained from the hydrogen binding potential on the molecular surface: hits in neurons colored black, non-hits in neurons colored gray

this case at A(3)] of the autocorrelation vector. A 26-dimensional autocorrelation vector was calculated in the range 0–13.5 A.

Separate Kohonen networks were trained with these six different structure representations, each network containing 60×45 neurons. The three different Daylight fingerprint representations all gave rather similar maps. As a typical result the Kohonen map obtained with 256 Daylight fingerprints is shown in Fig. 19.

As can be seen, the hits are spread all over the map; this representation is not useful for developing a screen to separate hits from non-hits.

From the autocorrelations of the three molecular surface properties, the results obtained from the hydrogen bonding potential appeared to be the most promising (Fig. 20). As can be seen, the hits collect in a cluster of the Kohonen map, although even in this cluster hits and non-hits are highly interdispersed. Is this a useful result? Remember that we want to separate hits from non-hits with an emphasis on making sure that we get as many of the hits as possible.

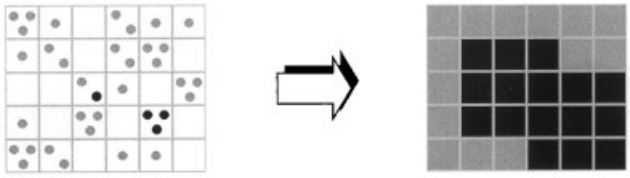

Figure 21. Developing a filter for finding hits: all neurons that obtain a hit in the training of a Kohonen network and their directly adjacent neurons are assumed to obtain hits in the prediction phase

To this end, we made the assumption that a neuron that has obtained a hit in the training phase is likely also to obtain hits from the virtual library. In order to make sure that we do not lose hits, we made the additional assumption that a hit might also end up in a neuron directly adjacent to a neuron that had obtained a hit. Figure 21 shows the development of such a screen by a recoloring of part of the Kohonen map.

After these preliminary investigations, we proceeded as follows: the dataset was split into two-thirds of the compound for training and one-third for testing making this split with the same ratio for both the hits and the non-hits. The structures were represented by autocorrelation of the hydrogen bonding potential and used for training a Kohonen network of 48×38 neurons. This gave the results shown in Figure 22 at the left-most side. After recoloring to obtain a classification filter, the map shown in the center of Fig. 22 was obtained. Sending the test set of 67 hits and 1761 non-hits through this network mapped 64 (96%) of the hits into the black area selected to

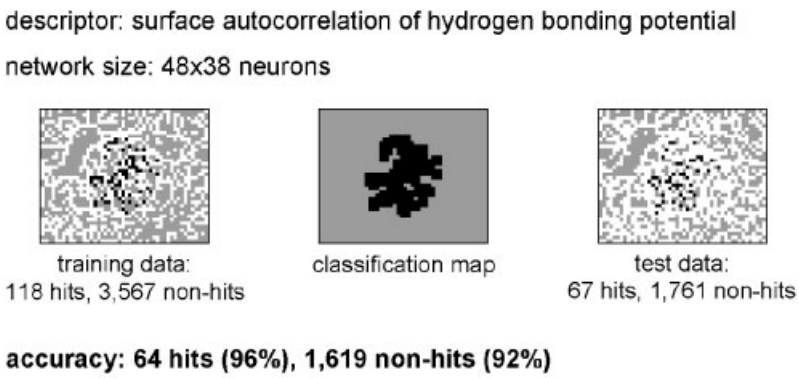

Figure 22. On the left-hand side the Kohonen map trained with two-thirds of the dataset (118 hits, 3567 non-hits); in the center the filter derived from this map (see Fig. 21); on the right-hand side the distribution of the test data (67 hits, 1761 non-hits). 64 of the hits (96%) fall in the black area of the filter, 1619 of the non-hits (92%) fall in the gray area of the filter shown in the center

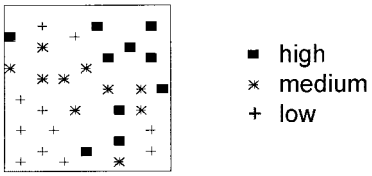

Figure 23. Distribution of the 31 steroids in a toroidal Kohonen map: in 10 neurons with highly active compounds, in 10 with compounds of intermediate activity and in 11 containing compounds of low activity

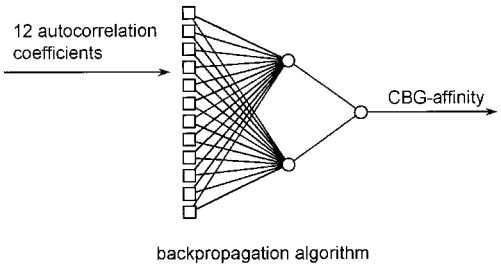

Figure 25. Architecture of the feed-forward network for the prediction of CBG binding affinity of the steroids

potentially contain hits and 1619 (92%) of the non-hits into the area around this cluster. Thus, 96% of the hits could be retrieved (i.e. 4% false negatives) with only 8% contamination by non-hits (false positives).

The essence of this study is that Kohonen networks, together with visualization tools, are very powerful for screening different structure representations to find the one most appropriate for the problem at hand.

ESTABLISHING QUANTITATIVE STRUCTURE–ACTIVITY RELATIONSHIPS—A COMBINED APPLICATION OF UNSUPERVISED AND SUPERVISED LEARNING

The power of a Kohonen network for rapidly testing different structure representations can also be utilized when a quantitative model between the structure of a compound and some of its physical, chemical or biological properties needs to be established.

A feed-forward neural network trained with the backpropagation algorithm[4,17] is such a powerful modeling technique that it can come up with an apparently good quantitative model that has nevertheless a sub-optimal predictive power. In order to develop as good a model as possible, we recommend first investigating different structure representations by an unsupervised learning method such as a Kohonen network, before using it to build a quantitative relationship by a supervised learning method such as a backpropagation (BPG) network.

In order to illustrate this point we use the widely studied dataset of 31 steroids binding to the corticosteroid binding globulin (CBG) receptor.[18] Again, an autocorrelation vector was used; this time, the molecular electrostatic potential on the surface of the steroids was condensed into a 12-dimensional autocorrelation vector [cf. eqn (4)].[8]

A Kohonen network consisting of 9×9 neurons in a toroidal arrangement was used for mapping these 31 steroids. The binding affinity of these steroids to the CBG receptor was then split into three categories, high, intermediate and low, and this classification was used for visualizing the mapping as indicated in Figure 23.

Having a toroidal arrangement of neurons, one should remember that the neurons on the right-hand margin are direct neighbors of the neurons on the left-hand margin and a similar situation is given for the neurons on the top and the bottom lines (cf. Fig. 5). One way of visualizing this closed topology is to replicate such a map several times and put such identical maps together like tiles. The result of tiling four maps of Fig. 23 is shown in Fig. 24.

Figure 24 clearly shows that steroids separate fairly well into those having high, intermediate and low activity. This result is indicative that the chosen structure representation is appropriate for modeling CBG receptor binding activity. Therefore, the 12 autocorrelation vectors of the molecular electrostatic potential of these 31 steroids were taken to train a feed-forward network of the architecture shown in Fig. 25 by the backpropagation algorithm.[8]

Cross-validation by the leave-one-out method gave the result shown in Fig. 26 with a cross-validated $r^2 = 0.86$ and a standard deviation in pK value of the binding constant of 0.42. A CoMFA model based on 21 compounds of the same dataset had a fourfold cross-validated r^2 of 0.66.[18] Note that the cross-validated r^2 of these two studies is not directly comparable as fourfold cross-validation is a more stringent test than leave-one-out cross-validation. However, it shows that a relatively simple representation such as autocorrelation vectors of the molecular electrostatic potential are highly successful in predicting CPG affinity.

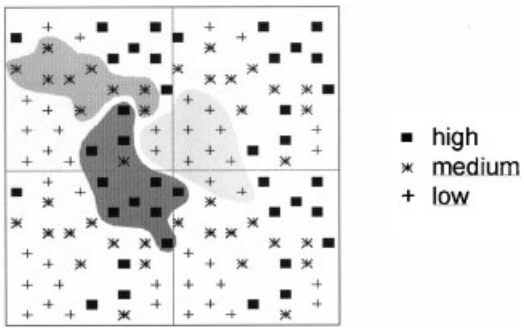

Figure 24. Assembly of four identical maps of Fig. 23 showing the closed topology of a toroidal map. As can be seen compounds separate fairly well according to activity

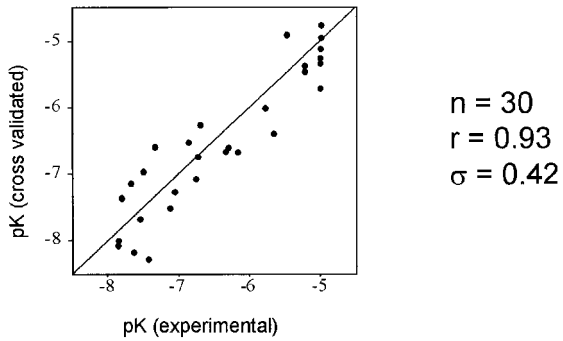

Figure 26. Cross-validated (leave-one-out) prediction of CBG binding affinity with the neural network shown in Fig. 25

Table 3. Dataset of compounds with different modes of action

1.	Non-polar non-specific compounds	14
2.	Polar non-specific compounds	18
3.	Uncouplers of oxidative phosphorylation	25
4.	Inhibitors of photosynthesis	15
5.	Inhibitors of acetylcholinesterase	14
6.	Inhibitors of respiration	3
7.	Thiol-alkylating agents	9
8.	Reactives	8
9.	Estrogenic compounds	9
	Total	115

MOLECULES WITH SEVERAL TYPES OF BIOLOGICAL ACTIVITIES—USING A COUNTERPROPAGATION NETWORK

Fairly often, molecules bind to several receptors and have several biological activities. Then, the problem might be to increase the selectivity, to maximize the binding to one activity while simultaneously minimizing the other activity, particularly if the other activity is toxicity. In the present study (unpublished work) we will investigate a dataset of 115 compounds having nine different biological activities.[19] Specifically, these compounds are all toxicants having nine different modes of toxic action. Table 3 shows these modes of toxic action (MoA) and the number of compounds for each MoA.

This dataset had already been studied by other statistical methods such as principal component analysis (PCA), linear discriminant analysis and the partial least-squares (PLS) method.[19]

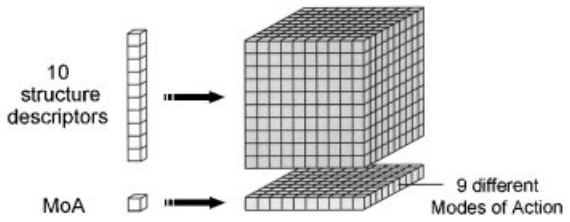

Figure 27. Architecture of the counterpropagation network for the classification of toxicants with one output layer

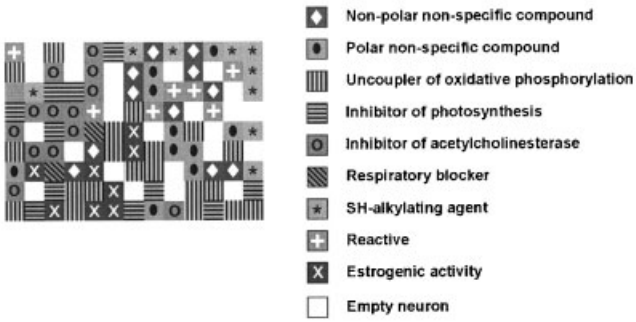

Figure 28. Projection of the toxicants of Table 3 into one output layer

Table 4. Variable selection methods applied (upper part) and 10 descriptors selected for representing the compounds of Table 3 (lower part)

24 Descriptors (geometric and physicochemical) → Principal Component Analysis → 13 Descriptors → Stepwise Discriminant Analysis → 10 Descriptors

MW	Molecular weight	$log\ K_{OW}$	Octanol-water distribution coefficient
MR	Molar refraction	ε_{HOMO}	Highest Occupied MO
D_{eff}	Effective diameter	V^+	Potential of possive atomic charges
SASA	Solvent-accessible surface area	Q_{AV}	Average of absolute atomic charges
SAVOL	Solvent-accessible volume	H^+_{MAX}	Maxium positive charge on hydrogen atom

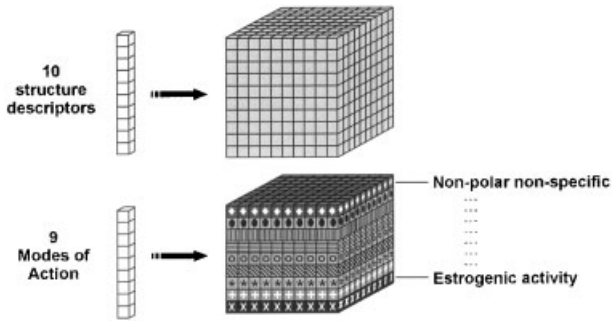

Figure 29. Architecture of the counterpropagation network for the classification of toxicants with nine output layers, one for each mode of action

We intentionally chose here the same structure representation as the one taken in the earlier study[19] in order to show some possibilities evolving from working with a neural network method. The emphasis in this example is placed on the comparison of different network architectures and not on quantitative results.

Table 4 gives the ten descriptors chosen for the representation of the 115 molecules of the dataset.

The dataset was then sent into a counterpropagation (CPG) network consisting of 13×9 neurons with 10 layers (one for each descriptor) in the input block and one layer in the output block (Fig. 27), with the output values having nine different values corresponding to the nine different MoA. The resulting distribution of modes of action in the output layer after training the network is shown in Fig. 28.

Clearly, no pronounced clustering of the compounds according to MoA can be discerned. What is the problem? Is the chosen structure representation not appropriate for this specific problem?

Rather than making this statement we want to show that a counterpropagation network can offer an architecture that is well suited to these multicategory datasets, to datasets for which a battery of biological activity data is available. For such problems, one layer for each biological activity should be chosen. In the given case, the architecture of a counterpropagation network shown in Fig. 29 was selected.

With this CPG network, interesting results were obtained: for modes of toxic action that correspond to toxicities associated with receptor binding a clustering of the compounds could be observed. For example, the layer in the output block corresponding to estrogenic compounds (layer 9 in the output block) showed a clear clustering of the active compounds (Fig. 30).

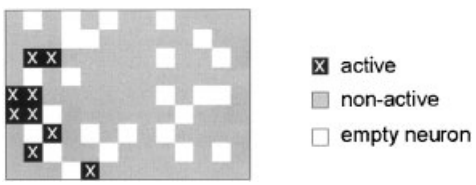

Figure 30. Distribution of compounds in the layer of estrogenic compounds; neurons colored in black and marked with × contain estrogenic compounds, neurons in light gray contain other compounds

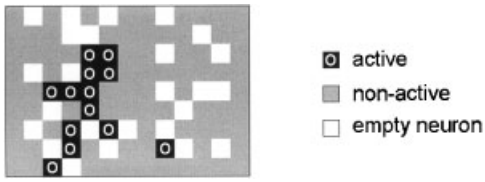

Figure 31. Distribution of compounds in the layer of inhibitors of acetylcholinesterase

In a similar manner, compounds that are inhibitors of acetylcholinesterase cluster in the corresponding layer of the output block (Fig. 31).

On the other hand, compounds corresponding to rather general, unspecific modes of toxic action are distributed over a broad area in the respective layer, as shown for polar non-specific toxicants in Fig. 32.

For these kind of toxicities, not quite specific structural prerequisites are required leading to some spread in structural space.

These results should have illustrated that the use of a counterpropagation network can lead to new insights when a battery of biological activities is given. Furthermore, a CPG network is the tool of choice for optimizing selectivity in different biological activities.

SUMMARY AND CONCLUSIONS

Learning from data has always been and still is the most important method for obtaining knowledge in chemistry. Powerful computerized learning methods have become available for assisting in this knowledge acquisition process.

Artificial neural networks are some of the most powerful data mining tools for combining data from diverse sources and finding connections between these data. Drug design in particular has a strong need for such data mining tools as, first, sometimes enormous amounts of data have to be processed, and second, complex relationships have to be studied and modeled.

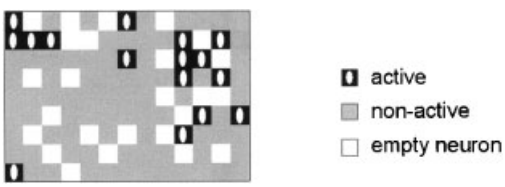

Figure 32. Distribution of compounds in the layer of polar non-specific toxicants

In this paper, we have concentrated on self-organizing neural networks and have shown a wide range of applications in the drug design process. We have particularly emphasized the importance of unsupervised learning. However, we do know that supervised learning is also of much importance but it should be the second step in data analysis.

We strongly believe that the architecture of a Kohonen or counterpropagation network is particularly suited for studying chemical data and the relationship between structural information and physical, chemical or biological properties.

Important as the particular method chosen for data analysis is, it should not be forgotten that the real secret to success in analyzing chemical data lies in an appropriately chosen structure representation. We have to strive to find the best structure representation for a given problem. Different levels of structure resolution—constitution, 3D structure, molecular surfaces—and different physicochemical properties have to be selected, conformational flexibility has often to be accounted for and chirality might be of influence. A wide range of structure coding methods have been developed—and to come back to where we started: self-organizing neural networks and visualization techniques associated with them offer a way of rapidly scanning different structure representations.

Acknowledgements

We thank the Bundesministerium für Bildung und Forschung (BMBF) and the Verband der Chemischen Industrie for funding our research. We want to thank Boehringer Ingelheim, Germany for providing us with the dataset of the hydantoin library and Dr. M. Nendza for making her dataset of modes of toxic action accessible to us. Further, we thank our co-workers mentioned in the references for their contributions to the research reported here.

REFERENCES

1. Gasteiger J. In *Proceedings of the 13th European Symposium on QSAR: Rational Approaches to Drug Design*, Höltje HD, Sippl W (eds). Prous Science: Barcelona, 2001; 459–474.
2. Gasteiger J, Zupan J. *Angew. Chem.* 1993; **105**: 510–536; *Angew. Chem., Int. Ed. Engl.* 1993; **32**: 503–527.
3. Zupan J, Gasteiger J. *Neural Networks in Chemistry and Drug Design* (2nd edn). Wiley-VCH: Weinheim, 1999.
4. Rumelhart DE, Hinton GE, Williams RJ. In *Parallel Distributed Processing: Explorations into the Microstructures of Cognition*, vol. **I**, Rumelhart DE, McClelland JL (eds). MIT Press: Cambridge, MA, 1986; 318–362.
5. (a) Kohonen T. *Biol. Cybern.* 1982; **43**: 59–69; (b) Kohonen T. *Self-Organization and Associative Memory* (3rd edn). Springer: Berlin, 1989.
6. Schuur JH, Selzer P, Gasteiger J. *J. Chem. Inf. Comput. Sci.* 1996; **36**: 334–344.
7. Hecht-Nielsen R. *Appl. Opt.* 1987; **26**: 4979–4984.
8. Wagener M, Sadowski J, Gasteiger J. *J. Am. Chem. Soc.* 1995; **117**: 7769–7775.
9. Hemmer MC, Steinhauer V, Gasteiger J. *Vibr. Spectrosc.* 1999; **19**: 151–164.
10. (a) Gasteiger J, Rudolph C, Sadowski J. *Tetrahedron Comput. Method.* 1992; **3**: 537–547; (b) Sadowski J, Gasteiger J. *Chem. Rev.* 1993; **93**: 2567–2581; (c) Sadowski J, Gasteiger J, Klebe G. *J. Chem. Inf. Comput. Sci.* 1994; **34**: 1000–1008.
11. (a) Gasteiger J, Marsili M. *Tetrahedron* 1980; **36**: 3219–3228; (b) Gasteiger J, Saller H. *Angew. Chem., Int. Ed. Engl.* 1985; **24**: 687–689.
12. http://www.mol-net.de.
13. http://www2.chemie.uni-erlangen.de/software/petra.
14. Moreau G, Broto P. *Nouv. J. Chim.* 1980; **4**: 757–764.
15. Broto P, Devillers J. In *Practical Applications of Quantitative Structure–Activity Relationships (QSAR) in Environmental Chemistry and Toxicology*, Karcher W, Devillers J (eds). Kluwer: Dordrecht, 1990; 105–127.
16. Bauknecht H, Zell A, Bayer H, Levi P, Wagener M, Sadowski J, Gasteiger J. *J. Chem. Inf. Comput. Sci.* 1996; **36**: 1205–1213.
17. Werbos P. In *System Modeling and Optimization: Proceedings of the International Federation for Information Processes*, Drenick R, Kozin F (eds). Springer: New York, 1982; 762–770.
18. Cramer RD III, Patterson DE, Bunce JD. *J. Am. Chem. Soc.* 1988; **110**: 5959–5967. The dataset can be downloaded at: http://www2.chemie.uni-erlangen.de/services/steroids/index.html.
19. Nendza M, Müller M. *Quant. Struct.–Act. Relat.* 2000; **19**: 581–598.