Applied Introduction to Multivariate Methods Used in Drug Discovery

Eugenia Migliavacca*

NCCR Molecular Oncology, Swiss Institute for Experimental Cancer Research, Bioinformatics Core Facility, Chemin des Boveresses 155, CH-1066 Epalinges, Switzerland

Abstract: The number of articles concerning optimization and applications of multivariate techniques in drug discovery testifies the growing importance attributed to these methods. This mini review focuses on some of the basic and most employed multivariate techniques in drug discovery research. Examples from the literature were selected to illustrate a number of potential applications.

Keywords: Multivariate data analysis, unsupervised methods, classification analysis, regression analysis, artificial neural networks, tutorial.

1. INTRODUCTION

The ample contribution of chemometrics to drug discovery is now well accepted and confirmed by the growing number of relevant publications. The advent of chemometrics techniques was motivated by the need to analyze and understand large volumes of biological and chemical data. However, at the same time, chemometrics as a discipline has advanced far beyond data analysis [1]. Novel concepts and methods are being developed to support many stages of chemical and pharmaceutical research; from target identification to lead optimization; from prediction of pharmacological compound characteristics to development of new formulations.

The focus of this review is on some of the basic and widely applied multivariate techniques in drug discovery research. Several demonstrative examples from the literature are presented to illustrate possible applications of multivariate techniques in drug design.

2. THEORETICAL BACKGROUND

The theoretical background is intended as a general introduction to the chemometrics techniques most commonly used in cheminformatics. This review is not exhaustive, its principal aim is to explain some of the currently used methods and to highlight their principal advantages.

The multivariate methods introduced in this review are extensively applied in different fields and are not intended exclusively for drug design, hence the use of the terminology universally adopted in statistics. Some of the key terms used in statistics are defined in Table 1. Table 2 shows the terms most likely to appear in the medicinal chemistry context.

Multivariate methods can be divided into two main groups, namely unsupervised and supervised. The unsupervised methods are largely used in exploratory data analysis. In these methods all objects are described as input vectors without the reference to the corresponding data target, no *a priori* knowledge of the class of the samples is required, Table 2. All objects can be utilized in the analyses, i.e., there is usually no distinction between the training and test sets, Fig. (**1**). The ensemble of input vectors forms the X data matrix. In the supervised methods, both input vectors (variables describing each object) and output vectors (responses or class attribution) are used in the analyses. The data set is subdivided into a training set and a test set, Fig. (**1**).

The training set data should be representative of the future population from which the new objects are drawn. The training set must extend over the x-space as widely as possible. The range of the training set defines application region of the derived model for future prediction. For example, if log P is among the chemical descriptors used to build a model to predict a biological activity, and the log P values range is limited by 2.0 to 8.0, it will be risky to estimate the activity for a compound whose log P value is 10.0. It is important to verify that the log P range is sufficiently broad. If this is not the case extrapolation is unsafe and can lead to erroneous results.

2.1. Unsupervised Methods

2.1.1. Principal Component Analysis

PCA is an excellent tool to provide an overview of the data, to detect trends, groupings and outliers (observations that are substantially different from the others and show extreme values that may strongly influence certain statistical analyses), to evaluate correlation among variables and their relative importance, and to reduce data dimensionality without a significant loss of information.

The objective of PCA is to find a way of condensing the information contained in a number of original variables into a smaller number of principal components (PCs) by decomposing the data matrix into a ‘structured’ part and a ‘noise’ part [2]. PCA aims to finds a new set of axes (PCs) such that most of the variability of the data is contained in the first few dimensions. The PCs are independent and uncorrelated variables that explain the observed variability. Each PC is a linear combination of the original variables. The size of contributions of the original variables depends on the relative orientation of PCs and the axes of the original variables, Fig. (**2**).

*Address correspondence to this author at the ISREC, Bioinformatics Core Facility, Chemin des Boveresses 155, CH-1066 Epalinges, Switzerland; E-mail: eugenia.migliavacca@isrec.unil.ch

1389-5575/03 $41.00+.00

Table 1. Basic Statistical Terms Used in This Mini Review

Name	Definition
Multivariate analysis	Analysis of multiple variables in a single relationship or set of relationships.
Chemometrics	Chemometrics is the science of relating measurements made on a chemical system or process to the state of the system via application of mathematical or statistical methods. Definition given by the International Chemometrics Society.
Parametric method	Parametric methods rely on the estimation of parameters (such as the mean or the standard deviation) describing the distribution of the variable of interest in the population.
Non-parametric methods	Nonparametric methods are used when the parameters of the distribution of the variable of interest in the population are not known (hence the name nonparametric). In other terms, nonparametric methods do not rely on the estimation of parameters (such as the mean or the standard deviation) describing the distribution of the variable of interest in the population.
Variance	The variance of a variable is a measure of the spread of the variable values. The squared root of the variance (standard deviation) expresses the measure of the spread in the same unit as the measurements.
Covariance	The covariance between two variables is a measured of their linear association. It depends strongly on the units of the variables.
Correlation	The correlation between two variables is a unitless, scaled covariance measured.
Categorical variable	The categorical variable assumes values that serve as a label; it is also referred as nominal or qualitative variable. ES
Continuous variable	The continuous variable can assume any numerical value. For any two values, there is another value between them that the variable may take on. ES
Training set	Set of objects used to derive the model.
Test set	Set of objects used to check predictive capacity of the model.
Cross-validation	The cross-validation procedure refers to the process of assessing the predictive accuracy of a model in a cross-validation set relative to its predictive accuracy in the training set from which the model was developed.
Outlier	An object that is substantially different from the other objects, that is atypical, i.e. it is represented by extreme values.

Table 2. Possible Correspondences of Generic Statistical Terms in Medicinal Chemistry

General term	Example
Object	Chemical compound.
Class	Therapeutic class; drug like compounds and non drug like compounds
Input vector	Set of parameters used to describe a compound.
Output vector	In regression analysis a biological activity or a physicochemical property that should be predicted. In classification analysis a distinct biological activity class label.

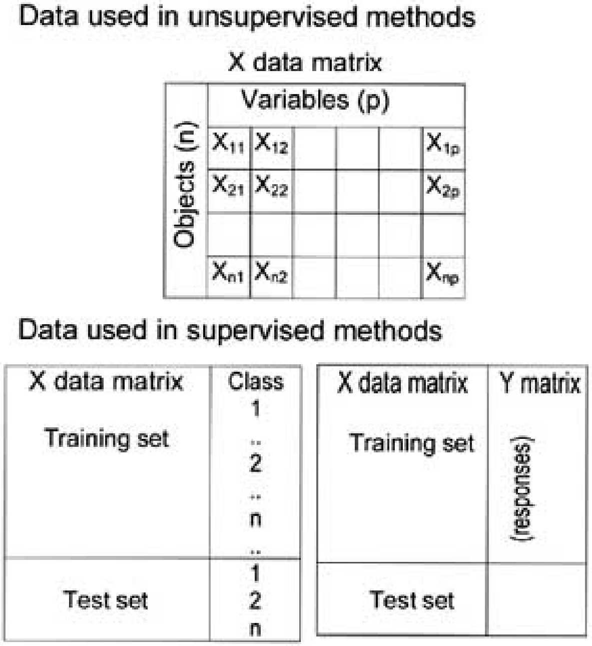

Fig. (1). Unsupervised methods employ only a X data matrix while supervised data analysis methods use information from the X data matrix as well as the data target that can be categorical (classification) or continuos (regression) variables.

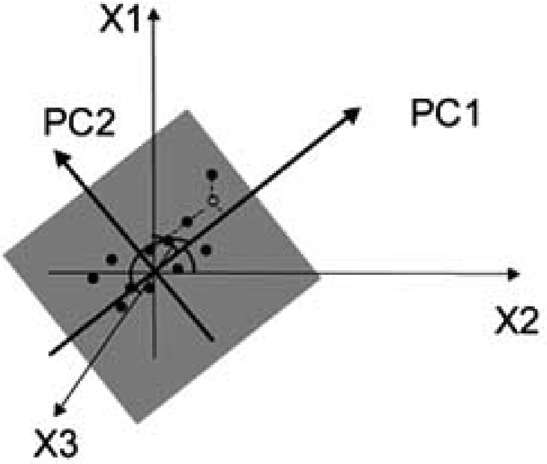

Fig. (2). Geometric interpretation of PCA. The PC1 represents the maximum variance direction in the data. Each observation may be projected onto this line in order to get a new coordinate value in the new coordinate system. PC1 and PC2 form a plane, which is a 2D window into the multidimensional X space. Each observation may be projected onto this plane in order to get new coordinate values in the new coordinate system. The new coordinate values are known as scores. Figure adapted from reference [3].

The first principal component (PC1) lies in the direction that explains the maximum amount of variation in the data matrix. The second principal component (PC2) describes the maximum amount of the remaining variation in the direction orthogonal to the PC1, Fig. (**2**). Successive principal components describe decreasing amounts of the remaining variation and are orthogonal to each other.

The procedure to determine the PCs consists firstly of the calculation of the appropriately scaled covariance (or correlation) matrix of the original data; secondly of the diagonalization of the covariance (or correlation) matrix to obtain the Eigenvalues (Eigenvalue matrix, Λ) and Eigenvectors (loading matrix, P). Finally, the original data is transformed by using the loading matrix as a rotation matrix.

Essentially the PCA works by decomposing the data matrix into the product of two matrices, the score matrix (T) and the transposed loading matrix (P'), and an additional residual matrix (E) [4], Eq. 1.

$$X(n,p) = T(n,a) \bullet P'(a,p) + E(n,p) \qquad \text{Eq. 1}$$

The first term, T • P', models the data structure while the second contains the part of data not explained by the PC model, the noise.

The score matrix (T) contains information about objects. Each object is described in terms of its coordinates with respect to the PCs. The plot representing the objects as projections onto the PC axes is known as the score plot, Fig. (**3**). The loading matrix (P) contains information about variables. The loading plot shows how the original variables are linearly combined to form the PCs, Fig. (**3**).

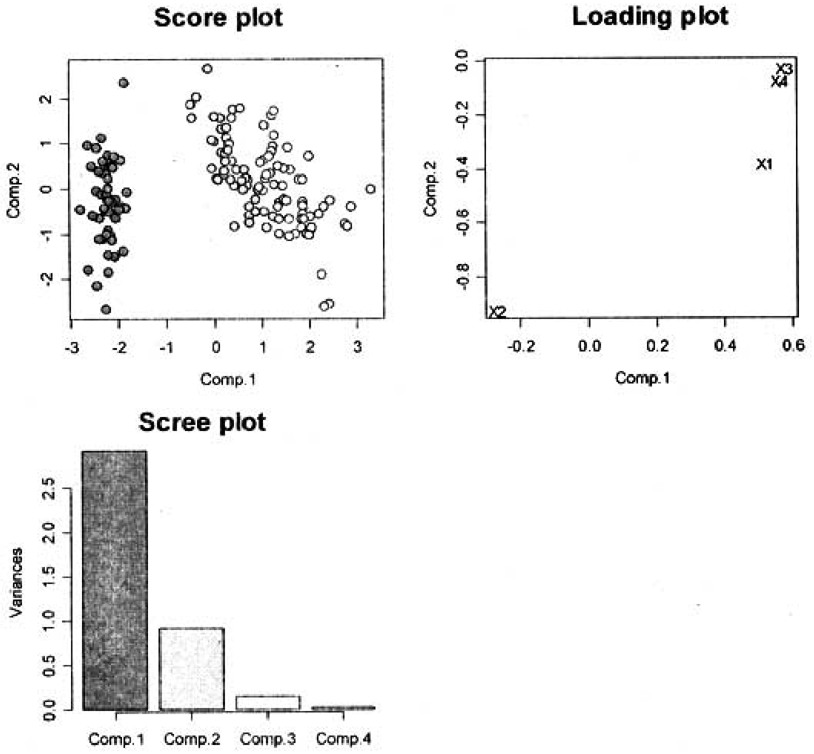

Fig. (3). Score, loading and scree plots. The score plot represents the objects in the new coordinate system, PC space. The loading plot represents original variables in the PC space, a variable with a high loading for a PC contributes a lot to this PC. For example variable X3 contributes a lot to PC1 while X2 contributes a lot to PC2. In the scree plot the Eigenvalues of the PCs are plotted against the number of PCs. This plot can be used to decide how many PCs are significant.

Different methods have been proposed to determine the number of significant PCs for a given data matrix. The most commonly used methods take into consideration either the percentage of variance explained by the PCs, or the Eigenvalues, i.e., the variance associated with each of the Eigenvectors [2, 5]. There are two main criteria based on the Eigenvalues: according to the first, only the PCs with an Eigenvalues greater than or very close to 1 are included, while the second is based on the analysis of the shape of the scree plot (Eigenvalues versus number of PCs), Fig. (**3**). The point at which the curve starts to straighten out indicates the maximum number of PCs to retain.

The PCA is frequently used for data description and exploratory data structure modeling [4]. It can be applied as an intermediate step in the more sophisticated data treatments [6].

PCA can lead to a better understanding of the data structure and improvement of the data interpretability.

Many softwares running under UNIX, Linux and Windows are available today to perform PCA calculation.

2.1.2. Cluster Analysis

Cluster analysis is a set of techniques for accomplishing the task of partitioning a series of objects into groups so that the objects within one group are more similar to each other than to those in other groups. Groups identified with these techniques are referred to as clusters.

There are two basic questions to answer in cluster analysis: The first one is how the similarity is measured. The second question is how the clusters are formed.

Similarity can be quantitatively measured using the concept of the correlation coefficient with higher positive correlation coefficient values representing greater similarity, or alternatively using the concept of distance, with smaller distances representing greater similarity [5].

The most frequently used similarity measure is the Euclidian distance. Essentially, it is a measure of the length of a straight line drawn between two objects, Fig. (**4**). For example the Euclidian distance between two compounds (s, t) described by their molecular weight (variable 1) and log P (variable 2) can easily be calculated by using the formula shown in Fig. (**4**). Since the Euclidian distance is quite sensitive to the scale of variables, the two descriptors should be properly standardized. Several other options are available. Some of the widely used alternatives are the Manhattan distance, based on the absolute differences of the coordinates of two objects, the Mahalanobis distance (a standardized form of the Euclidian distance), and the Pearson correlation coefficient. When objects are described using binary variables the most commonly used distance is the Tanimoto distance.

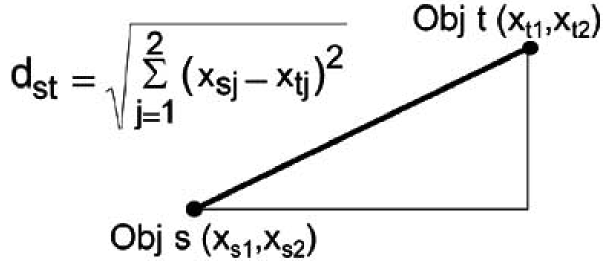
$d_{st} = \sqrt{\sum_{j=1}^{2} (x_{sj} - x_{tj})^2}$
Obj t (x_{t1},x_{t2})
Obj s (x_{s1},x_{s2})

Fig. (4). An example of Euclidian distance between two objects (s,t) measured on two variables.

As pointed out by Kubinyi [7], it is important to remark that the definition and quantitative description of chemical similarity is one of the critical issues in structure-activity relationship (SAR) studies as well as in combinatorial chemistry. Compound similarity can be safely defined only in the chemically closely related series that interact with a given biological target.

Numerous procedures for forming clusters have been developed. All clustering algorithms try to maximize the differences between clusters relative to the variation among objects within each cluster.

The most commonly used clustering algorithms can be classified into two general categories: hierarchical and nonhierarchical.

There are essentially two hierarchical clustering procedures: agglomerative and divisive. In agglomerative methods each object starts out as a cluster in itself and subsequently the two closer clusters (or objects) are combined into a new cluster. In the divisive methods all objects initially belong to one large cluster and in succeeding steps the most dissimilar objects are split off and made into smaller clusters.

Among the hierarchical methods, agglomerative clustering algorithms are the most widely used. There are several variations of agglomerative clustering, which differ in how the distances are measured between clusters as they

Table 3. Principal Hierarchical Clustering Algorithms

Name	Definition of distance between two clusters	Comments
Single linkage	Distance between the closest objects in two clusters (minimum distance).	It can identify long, thin clusters.
Complete linkage	Distance between the most far-away objects in two clusters (maximum distance).	It finds tight, spherical clusters.
Average linkage	Distance between two clusters as the average of the distances of all objects in the two clusters.	It is less sensitive to outliers, tends to combine clusters with small variance.
Ward's method	Distance between two clusters as the sum of squares between two clusters summed over all variables. It minimizes the within-cluster variation.	It aims at finding compact, spherical clusters, of about equal size.
Centroid method	Distance between two clusters as the distance between the cluster centroids.	A centroid of a cluster is the average value of the objects contained in the cluster on each variable

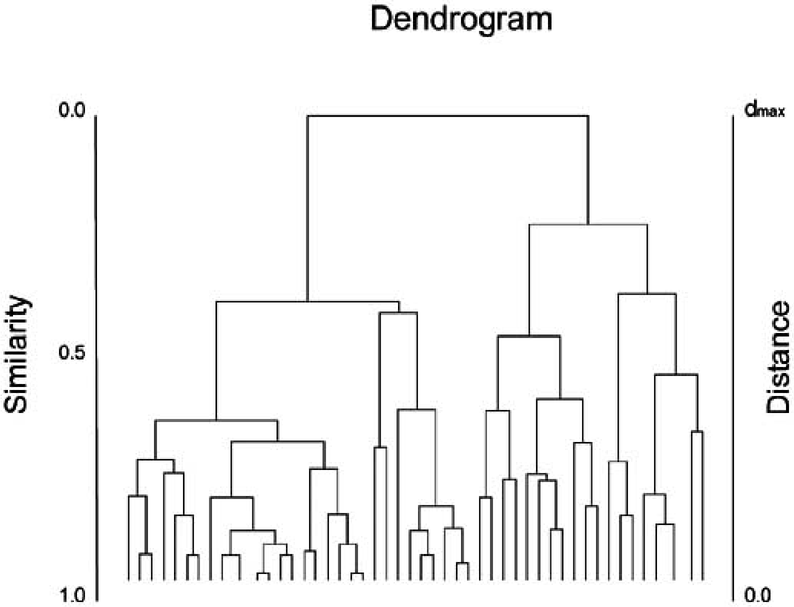

Fig. (5). A sample tree diagram (dendrogram) illustrating hierarchical clustering.

are constructed. Some of the most commonly used are single linkage, complete linkage, average linkage and Ward's method, see Table 3.

Each of these procedures may produce different results, as will any algorithm if different distance metrics are employed. Any cluster analysis is neither true nor false, it should be largely judged on the usefulness of results [8].

The results of hierarchical clustering are usually presented as dendrograms, in which the distance along the tree from one element to the next represents their relative degree of similarity, Fig. (**5**).

Among the nonhierarchical methods, the k-means clustering is the most commonly used algorithm [8]. This is a relocation algorithm based on the distance of each object from the centroid of each cluster. The number of k clusters is pre-fixed by the user. At first all objects are randomly assigned to one of the k clusters. Then the cluster centroid is calculated for each cluster. Subsequently, using an iterative method, objects are moved among clusters. Objects remain in the new cluster if they are closer to it than to the previous cluster. Partitioning continues until moving any object starts to increase the within-cluster variation and decrease the inter-cluster dissimilarity.

Determining the final number of clusters (stopping rule) is still a perplexing issue in cluster analysis. There is no standard and objective procedure, but there are many criteria and guidelines that have been developed to address this problem. The simplest and most common type of stopping rules consists of analyzing some measure of similarity or distance between clusters at successive steps, for example monitoring the average distance within a cluster. The final number of clusters is defined when the similarity measure exceeds a specified value or when the successive value between steps undergoes a large decrease. Other stopping rules attempt to apply statistical rules or adapt statistical tests, such as the cubic clustering criterion or the likelihood ratio. There is no solution that appears to be better in all situations, therefore the combination of theoretical foundations with the understanding of data is essential to make a good choice [8].

There is a wide range of applications of cluster analysis in computational chemistry. These applications may vary from the selection of representative compounds in a large chemical library to the evaluation of large amount of data accumulated in the course of a conformational analysis.

2.1.3. Kohonen Networks, Self-Organized Maps

The neural networks are analytic techniques modeled after the process of learning in cognitive systems, and the neurological function of the brain, see also 2.2.3. The neural network method developed by Kohonen is rather efficient in modeling the generation of sensory maps in the brain. This method is designed for clustering problems and operates in an unsupervised learning mode.

The Kohonen neural network, also termed the self-organizing map (SOM), is a technique that has been used for grouping in high dimensional space and projecting onto a lower dimensional space, usually the two-dimensional space for visualization purposes.

The aim of Kohonen learning is to map similar objects to similar neuron positions, identifying similarities between objects [9]. Training is performed in such a way that objects, represented by input vectors, with similar properties are mapped onto the same neurons (or nearest neighbors) in the two-dimensional space.

A Kohonen network is based on a single layer. This layer is usually arranged in a plane with its dimensions defined by the user. The dimensions of a Kohonen network are specified as x • y • n or as x • y, where x is the number of neurons in the first dimension of the active layer, y the number of neurons in the second dimension, and n the dimension of the input vector (number of variables representing each object), Fig. (**6**). A Kohonen network is characterized by the x • y • n weights. Each neuron has as many weights as there are input variables, Fig. (**6**). Before the training starts, random codes are assigned to all weights. During the training, all neurons receive the same input and each object is mapped to the neuron that contains the most similar weights compared to its input vector.

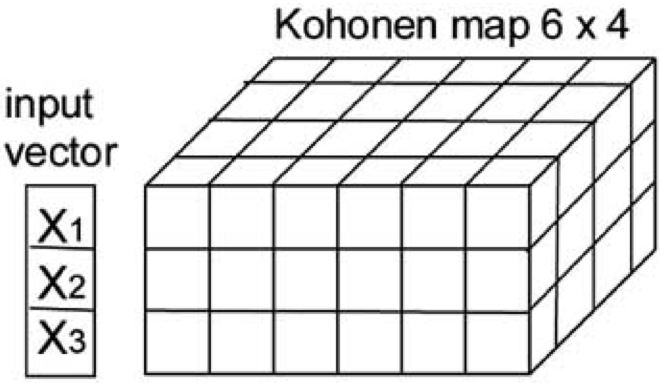

Fig. (6). A Kohonen network of 24 neurons (6 • 4) characterized by 72 weights (6 • 4 • 3).

A single cycle of the Kohonen algorithm can be briefly illustrated as follows. An object described by n variables enters the network, and the responses of all neurons (each having n weights) are calculated. The neuron whose output is the largest, or with the weights most similar to the input vector is selected. This is the central neuron and is denoted here as neuron 'c'. Subsequently the weights of the neuron 'c' are adjusted to make them more similar to the input vector and to improve response of the neuron 'c'. The weights of all the neurons in the neighborhood of the neuron 'c' are corrected in inverse proportion to the topological distance from the neuron 'c'. The next object is input and the process repeated [9].

All objects of the training set are iteratively fed to the network, the weights adjusted and the training is stopped when a pre-defined criterion (a measure of stability) is met. After training, all the objects of the training set are mapped. Similar objects are mapped onto the same or adjacent neurons.

The Kohonen neural networks have been largely used in drug design for the mapping of molecular surface properties (see article by Gasteiger in this issue), such as the molecular electrostatic potential, into two dimensions [10,11] and also as a clustering procedure [12,13]. Applications of this method in QSAR studies were fully reviewed elsewhere [14].

Many software packages running under UNIX, Linux and Windows are today available to perform neural network calculation. Here are a few of the major software tools; **SNNS- Stuttgart Neural Network Simulator:** A complete simulator with graphical network editing and visualization tools. It is well documented, customizable and can run under several platforms (Unix+X), **Netlab Software:** A library of Matlab® functions and scripts based on the approach and techniques described in the book Neural Networks for Pattern Recognition by Bishop C.M. [6]. **MathWorks: Neural Network Toolbox** A neural network development environment that requires MATLAB. An extensive, annotated list can be found on the NEuroNet website http://www.kcl.ac.uk/neuronet/index.html at King's College, London, UK.

2.2. Supervised Methods

2.2.1. Classification Methods

The classification methods try to find a relation between X-variables (predictors) that describe objects and a qualitative variable that defines classes, Fig. (**1**). To apply classification methods, classes should be previously defined and each object of the training and test sets should be attributed to a class. The essential difference between classification methods and cluster analysis is that in the former the number of classes is known before the analysis and each object is attributed to a class, while in the latter we are looking at how objects group together, with the objects having no predefined labels and the unknown number of groups (clusters).

2.2.1.1. Discriminant Analysis

Discriminant analysis (DA) tries to find features that optimally separate objects in different classes. This method allows to identify boundaries between classes of objects. These boundaries are linear in the linear discriminant analysis (LDA), i.e., are represented by lines in two dimensions, planes in three dimensions and hyperplanes in higher dimensions. They appear as quadratic functions in the quadratic discriminant analysis (QDA). The discriminant function is determined in such a way as to minimize the classification errors, and is defined as a combination of the original variables that are able to discriminate the objects in the respective classes. The effectiveness of DA rests on the existence of independent variables that differ in the mean value from one class to another.

There are some key assumptions for the proper application of DA. The X-variables should correspond to a normal distribution and should be uncorrelated or moderately correlated; the variance of a given independent variable should be unchanged through the different classes, and the correlation matrices of the independent variables should be the equivalent for each class. Data not meeting these requirements can negatively affect the results and cause problems in the estimation of the discriminant functions. In these cases, the utilization of other classification methods such as the k-nearest neighbors or SIMCA should be considered.

2.2.1.2. k-Nearest Neighbors

The k-nearest neighbors is a nonparametric classification method, i.e., there are no assumptions on the variables distribution (Table 1), based on the analogy concept [15]. A distance matrix for all objects is calculated, usually utilizing the Euclidian distance, and an integer 'k number' is selected. The 'k number' is the number of the nearest neighbor objects considered in the class estimate of a new object. Indeed, an unknown object is classified on the basis of the class memberships of its k nearest neighbors. To determine the nearest neighbors, the distance matrix of all samples in the training set is checked for the k shortest distances to the new object. The new object is then assigned to the class that appears the most frequently within the k nearest neighbors, Fig. (**7**). If the nearest neighbors are equally distributed among different classes, the new object is assigned to the class for which the sum of the distances between the nearest neighbors belonging to this class and the new object is minimal.

The 'k number' should be large enough to minimize the probability of misclassification and small relative to the number of samples so that the nearest neighbors are close enough to the new object to give an accurate estimate of its true class. In practice various values of k are tested to find the best solution.

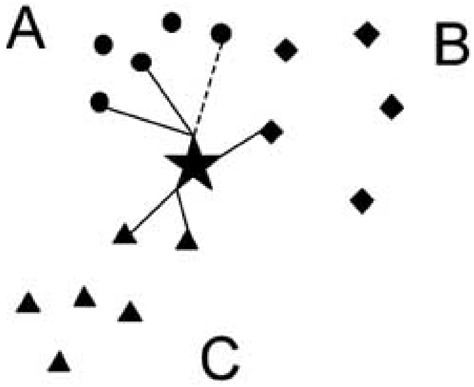

Fig. (7). A new objects is assigned to the class that appears most frequently within the k nearest neighbors. For k=5 the new object is assigned to class C, and for k=6 to class A.

This method usually gives good results, especially if the separation surface between classes is not linear.

2.2.1.3. SIMCA

The SIMCA (Soft Independent Modeling of Class Analogy) is based on the hypothesis stating that each class can be approximated by a PC model with few components, provided that most of the X-variables express a real similarity [16]. The philosophy behind this method is to allow objects to have individualities and to model only the common properties of the classes.

In the SIMCA approach each class of objects is modeled separately. Since each class has to support a PC model, this method is applicable if several objects populate every class. A complete classification consists of one PC model per class. A tolerance interval is introduced for each class and a new object is assigned to a class only if it fits inside the tolerance interval of that class.

The SIMCA method presents several advantages. It can be employed in the cases where the number of variables is larger than the number of objects, it can also easily handle the correlated variables and allow to represent results graphically [17].

2.2.2. Regression Analysis

The multivariate regression analysis relates two matrices, namely the X data matrix containing predictor variables and the Y response matrix consisting of criterion variables, by regression and provides a mathematical model, i.e., an equation describing the nature of the relationship between these two sets of variables. The multivariate model for X and Y is mostly used for prediction. We wish to estimate the response (dependent variable) for a new object described by a set of predictor variables.

2.2.2.1. Multiple Linear Regression

The multiple Linear Regression (MLR) is intended for the regression of a single dependent variable (Y-variable) on a set of independent variables (X-variables). The MLR formulation implicitly requires that all X-variables are not intercorrelated, i.e., columns in the data matrix X are linearly independent. This is a very important point since in most scientific applications the descriptors chosen to characterize objects are intercorrelated. Moreover MLR assumes that the X-values are not affected by errors, i.e., that they are noise free. In MLR the number of variable should exceed the number of samples.

MLR is the best method for the truly uncorrelated X-variables and X-values with insignificant errors. If this is not the case, the use of other approaches, e.g., PCR and PLS, is strongly recommended.

2.2.2.2. Principal Component Regression

The Principal Component Regression (PCR) can be viewed as a two-step procedure. First a PCA is conducted on the X data matrix to derive the T score matrix, and then on MLR is carried out on the T score matrix.

In contrast to MLR, the PCR has no problems with correlated X-variables. In PCR the score vectors are orthogonal and it can better cope with 'noisy' X-variables since the last PCs containing noise are discarded.

There is still one aspect of PCR that is not optimized. The decomposition of the X data matrix into PCs is conducted without taking into account the Y response matrix. This means that the decomposition is carried out in a way that does not guarantee the best results for the Y-variables prediction. This seems to be particularly true if we try to model more than one response at the time.

2.2.2.3. Partial Least Squares

The Partial Least Squares (PLS) [16,18] is a method for relating two matrices to each other by a linear multivariate model; it can be seen as a regression extension of PCA. PLS finds the linear relationship between a Y response matrix and X data matrix, Eq. 2.

$$Y = f(X) + E \qquad \text{Eq. 2}$$

PLS uses the variance of the Y response matrix to directly guide the decomposition of the X data matrix in such a way so as to obtain an optimal regression.

The main difference between PCR and PLS lies in the fact that PLS obtains the linear combinations of the original X-variables called Latent Variables (LVs) under two constraints: (1) LVs provide the best possible representation of the structure of the X data matrix and (2) LVs maximize the fitting between the X and Y matrices. Because of this direct involvement of the Y matrix in the decomposition of the X matrix, the PLS approach produces models with fewer components than PCR and superior interpretation possibilities.

PLS works by decomposing the X matrix into the product of two smaller matrices, similar to PCA, the loading matrix (P) and the score matrix (T), Eq. 3. The Y matrix is decomposed into the Y score matrix (U) and the Y weighting matrix (C), Eq. 4. X, and Y scores are connected and correlated by the inner relation, Eq. 5, giving Eq. 6.

$$X = T \bullet P' + E \qquad \text{Eq. 3}$$

$$Y = U \bullet C' + F \qquad \text{Eq. 4}$$

$$U = T + H \text{ (inner relation)} \qquad \text{Eq. 5}$$

$$Y = T \bullet C' + F \qquad \text{Eq. 6}$$

In the PLS algorithm, there is an additional loading matrix called matrix of weights (W) expressing the correlation between X and U. The matrix W is used to calculate T.

A PLS solution can be expressed by Eq. 7 where B is a matrix of PLS regression coefficients.

$$Y = XB + F \qquad \text{Eq. 7}$$

The best way to examine the information derived PLS analysis is by graphically plotting the matrices obtained in the analysis. Some plots are remarkably informative. The T/U score plot allows to check the correlation between X and Y obtained in the PLS model for each LV. The plot relative to the first LV considering the two score vectors t_1 and u_1 is the most instructive to visualize the correlation structure between X and Y. The (T) score plots represent objects in the space of the LVs, while the loading plots represent original variables in the space of the LVs. Since the loadings of a variable indicate how much this variable contributes to the LVs, variable with high loadings contribute a lot to the LVs. The weight plots represent the original X variables in the space of the weights. Since the weights represent how X variables combine to best fit the Y matrix, variables with high weights are highly correlated with Y variable(s). The coefficient plots offer a compact representation of a PLS model. There is one coefficient plot per Y variable, and it shows the influence of the X variables on each response, Y variable.

The predictive power of a model can be evaluated by cross-validation and/or using a test set. Cross-validation is an approach for assessing the best model in prediction. It is useful to establish the best level of complexity (number of LVs) for a model in order to distinguish between information and noise. Cross-validation assesses the probable predictive power of a model by attempting a prediction of all the objects in a set. Several models are derived by excluding one (leave-one-out) or more objects from the analysis until all objects are kept out at least once. In the most common leave-one-out cross-validation, every object is eliminated once. For n objects, n models are derived and n predictions are compared with the real output. The squared correlation coefficient R^2 is a quantitative measure of the goodness of fit, while the squared cross-validated correlation coefficient Q^2 is used as a quantitative measure of the goodness of prediction (Eq. 8). In Eq. 8, yi is the real output value for the object i, $\bar{y}$ is the average value and $\hat{y}i$ is the predicted output value derived from a model in which the object i was excluded. Most often, the highest Q^2 is taken as a criterion for selecting the optimum number of LVs.

$$Q^2 = 1 - \frac{\sum_i \left(y_i - \hat{y}_i\right)^2}{\sum_i \left(y_i - \bar{y}\right)^2} \qquad \text{Eq. 8}$$

PLS can also be used for classification, PLS-DA [19]. Instead of using a Y response matrix constituted by continuous variables, a Y matrix composed by dummy variables, which describe the class membership of each object, is used in the regression onto the X data matrix. The 'dummy' Y matrix has as many columns as classes filled with '1' and '0'. For an object of class k, the k^{th} column will assume a value of one while the other columns will be set at zero. Again, linear combinations of the original variables with good ability to distinguish between classes are extracted.

2.2.3. Artificial Neural Networks

The flexible nature of Artificial Neural Networks (ANNs) makes them adaptable to a wide range of problems, ranging from classification to non-linear regression and cluster analysis (see Kohonen neural networks).

The theory and general practice of ANNs and their applications in drug design have been reviewed in depth [9], therefore our discussion will include only the basic concepts of ANNs.

Among the different systems of ANNs that utilize the supervised learning method, the ANN with back-propagation of errors is applied most frequently in drug design.

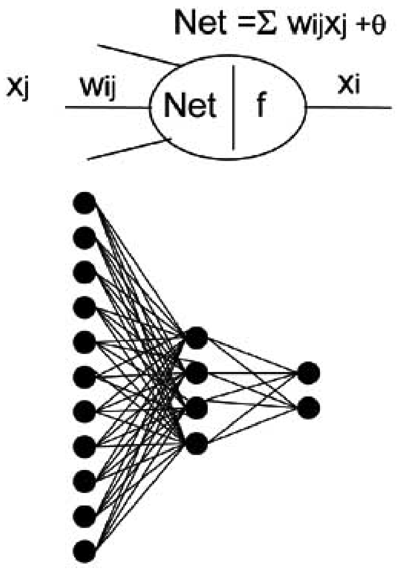

Fig. (8). Schematic representation of a node, which is the basic processing unit in a neural network, and a fully connected neural network with an input layer, one hidden layer and an output layer with two nodes.

The basic processing unit in ANN is a node or neuron, which receives input data, elaborates them and emits an output in analogy with neural cells, as shown in Fig. (**8**). Each connection with another node has an assigned weight defining the synaptic strength. The weights are adapted so that for any set of known objects the output values are as close as possible to the expected values. A node first processes the incoming data by summing each input value multiplied by its respective weight. At the next step this summed value is transformed by an activation function, also called transfer function, to generate an output value.

There are three basic types of nodes: input nodes, output nodes and hidden nodes. An input node receives a single input variable and transmits it to the network. An output node receives input and calculates an output or final value. In the multilayer networks there is a third type of node enclosed in the hidden layers, which receives input from different nodes of the previous layer, calculates output and sends it to other nodes, Fig. (**8**).

Three essential elements characterize a neural network: the architecture of the network, the arithmatic operation inside the node, and the learning method. We will briefly describe them.

(1) The network architecture is defined by the number of nodes, the number of layers and connectivity. The number of input variables determines the number of input nodes. The number of output nodes is equal to the number of predictive models (and is often equal to one), or to the number of classes in a classification problem. The number of hidden nodes and hidden layers is determined by trial and error and is limited indirectly by the number of samples for small data sets. Indeed it is preferable that the number of weights to be determined in a network, which depends also on the number of hidden nodes, is smaller than the number of data values (n objects multiplied by the p descriptors). In most cases neural networks consisting of one hidden layer are used. Unless differently specified on the basis of some previous knowledge, the layers of neurons are fully connected, Fig. (**8**).

(2) The mathematical function within the node that translates the summed score of the weighted input values should provide a final output value that is non-negative and continuous. Although there are several functions that satisfy these conditions, the most widely used is the sigmoid function, which is a nonlinear function with the S-shaped distribution.

(3) The most common form of learning is back-propagation of errors. Once the input vector for an object is processed through the system, the estimated output value is compared to the actual output value. If there is a difference between the two values, the system tries to improve the model in order to decrease this difference. In the back-propagation of error approach, the error in the estimated output value is calculated and then distributed backward through the system. The weights are changed throughout the layers, beginning with the weight correction in the last layer and progressing backwards towards the input layer.

For supervised learning the objects should be divided into three sets: a training set, a validation set, which allows to determine when to stop the training, and a test set for testing the predictive ability. If the number of objects is reduced and does not allow the subdivision in three sets, the test set is also used to establish when the training of a network is completed. In order to avoid overtraining it is critical to determine when the training of a network is completed. An overtrained network may well reproduce the output for the training set but is not robust when it has to estimate new data. This phenomenon is mainly due to parameter redundancy resulting from an overly complex network. Such network presents a large number of hidden nodes or multiple hidden layers that are unnecessary.

ANNs are largely used in applied and theoretical chemistry. There is an increasing number of publications using ANNs in drug design, different applications utilizing ANNs to predict properties, such as lipophilicity [20,21] available on the Internet.

2.2.4. Recursive Partitioning

Recursive partitioning algorithms (RP) can be used both as classification methods and as non-linear regression techniques. RP is a powerful approach allowing to uncover complex structure-activity relationships hidden in large chemical data sets, and yield interpretable modes [22,23]. RP techniques aim to detect the most statistically significant features that split the data set into smaller and more homogeneous subsets, correlating the object descriptors (X-variables) with dependent variables (Y-variables) or classes [24,25]. The result of a RP analysis is usually represented by a tree structure, Fig. (**9**). Initially, all objects reside in a node called root. Subsequently, for a binary tree all objects in a node are recursively split into two statistically distinct nodes. The statistically most significant split becomes a branch point in the RP tree and identifies the best rule to split the data into two subsets. The creation of a new branch point results in the generation of two subsets. Each subset has comparable features and similar responses. The process ends when there are no more statistically significant splits. Terminal nodes (leaves) appear at the end of each branch. These nodes correspond to small groups of objects for which relationships between descriptors and responses have been developed.

An example of application of recursive partitioning is given in the review by Alanas Petrauskas.

On the homepage of Chemometrics (http://www.acc.umu.se/%7Etnkjtg/chemometrics/softlinks.html), there is a list of major software packages and also some data sets. Most of the methods described here are also available in general statistical packages, for example in the ‘R’ package, which is free and can be downloaded at http://www.r-project.org/. The ‘mva’ package in R allows to perform multivariate analysis.

APPLICATION OF CHEMOMETRICS TECHNIQUES IN MEDICINAL CHEMISTRY

3.1. Problem 1: HCA to Evaluate Conformational Analysis

Chemical compounds are not rigid, and they exist at each moment in many different comformers. Conformational analysis is applied to identify low-energy comformers. Often a large number of conformers represent an ensamble of energetically accessible conformations for a particular

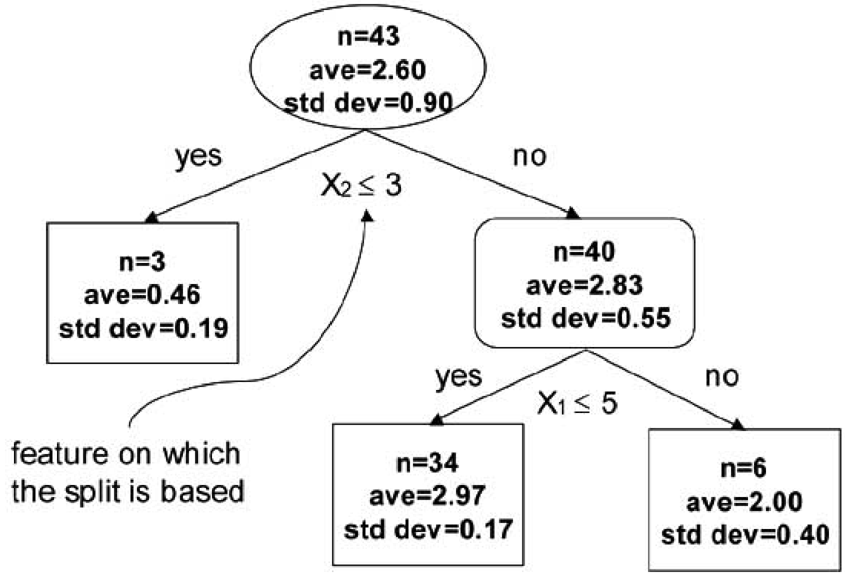

Fig. (9). A small binary decision tree.

compound. Many conformers are very strongly related and may be combined into common families. Several methods have been developed to group conformers into such conformational families [26], among them is cluster analysis. The parameters used for clustering are torsional angles or/and distances between important functional groups. Ermondi *et al.* [27] performed a conformational study of cetirizine and hydroxyzine at different protonation states by quenched molecular dynamics. Three structural parameters were analyzed for the conformers, expressed as distances between atoms and standardized by mean centering (new mean equal to 0) and scaling by the inverse of the standard deviation (new variance equal to 1). The Euclidian distance was adopted to calculate diversity between pairs of conformers. Conformational analysis combined with the HCA gave a good selection of conformers possibly present in aqueous solution. These results were confirmed by NMR measurements.

3.2. Problem 2: Analysis of Anticancer Activity Pattern Database

Hierarchical clustering algorithms are commonly used in compound selection and diversity analysis. Many such applications utilize binary representation of chemical structure, such as MACCS keys or Daylight fingerprints [28], and a distance measure, such as Tanimoto coefficients. These are by far the most common applications in drug design, although many other interesting applications are possible. Shi and his colleagues at the US National Cancer Institute (NCI) [29] analyzed with various statistical techniques the anticancer activity pattern database generated by the NCI anticancer drug discovery program. This database contains the activity values expressed as GI_{50} (the compound concentration required to inhibit cell growth by 50% compared with the controls) for about 70,000 compounds tested across 60 human cancer cell lines. The PCA, HCA and other techniques were applied. Analyzing the score plot of the first two PCs the authors were able to readily identify outliers and data entry errors. They clustered 25,023 compounds by their *in vitro* activity patterns across the 60 cell lines, i.e. utilizing 25,023 objects described by 60 variables or by a vector of 60 anticancer activity. As the distance measure, they used '(1-r)' where r is the Pearson correlation coefficient between the activity patterns of two compounds or clusters. The average linkage was selected as clustering algorithm after different algorithms were investigated. A data set of 131 agents of known mechanisms of action was analyzed. Based on the analysis of the dendrogram, three important observations were possible. Compounds with similar structure have a tendency to cluster together. Compounds with similar mechanism of action are likely to cluster together even when they are structurally dissimilar. Compounds similar in structure but different in the mechanism of action are distant from each other. Even this simple type of analysis can provide precious information to better understand the molecular pharmacology of cancer.

3.3. Problem 3: Blood-Brain Barrier Prediction

Recently, much attention in drug design has been focused on estimating the absorption, distribution, metabolism and excretion (ADME) properties (see article by Lombardo *et al.* in this issue). As part of this novel interest in ADME prediction, the determination of blood-brain barrier (BBB) penetration assumes great importance. For drugs targeted at the central nervous system, BBB penetration is a necessity, while for drugs targeted at other sites of action is an unwanted and possible dangerous feature. Therefore, it is of critical importance for the pharmaceutical industry to have models that can discriminate between compounds with high and low BBB permeability. Numerous publications [30-34] have addressed the problem of the BBB permeation modeling.

As an example of application of the PCA and PLS, we briefly describe the models obtained by Crivori [34]. The training set consisted of a data set of 44 compounds that was enlarged to 110 to take into consideration all the enantiomers of racemic drugs. Indeed, compounds were characterized by 72 descriptors derived from the 3D molecular fields (see article by Gasteiger in this issue) using VolSurf [35]. A set of 120 compounds (derived from 108 drugs) was used as an external test set. First a PCA was conducted on the 110 compounds and a model with three PCs was adopted. The first PC is able to discriminate between compounds that can penetrate the BBB (BBB+) and compounds that cannot (BBB−), as can be observed on the score plot of the first two PCs. The second and third PCs describe the chemical variability and spatial geometry. After that the predictive capacity of the PCA model was assessed using the external test set. Once the PCA model was obtained, PCA predictions for the test set were made by calculating the score vector T of the descriptors X for the new compounds using the loading P of the PCA model. Thus it was possible to plot the new compounds in the PCs space of the 'original' PCA model. A PLS discriminant analysis was carried out on the two combined data sets. A cross-validation procedure was conducted to select the number of significant LVs and to test the predictive capacity. The 'two-LV' model correctly predicts more than 90% of the enlarged data set (229 compounds since one was considered an outlier). Moreover the descriptors influencing the model are related to known molecular factors that affect the BBB penetration. It is possible to make this observation by looking at the coefficient plot, which shows the contribution of all descriptors to the model.

3.4. Problem 4: PLS in CoMFA

Since many of the chemical descriptors commonly used in computational drug design often have a certain degree of collinearity, MLR is not the most used technique in regression analysis. As mentioned above, PCR and PLS are the best candidates for solving collinearity problem in regression. Since it has been shown that PLS might be more efficient than PCR in extracting relevant information from the X data matrix and getting better results for the Y-variables prediction, PLS is the most utilized regression technique in computational chemistry. The main use of PLS is to model the relationships between theoretical descriptors and/or measured variables that characterize the structural variation of a data set and biological responses or physicochemical properties. As a result of the development of PLS, the 3D quantitative structure-activity relationships (3D-QSAR) and in particular the method called the comparative molecular field analysis (CoMFA) could be developed and became largely used [36-38].

Lopez-Rodriguez and her colleagues [39] have conducted a 3D-QSAR study, using the CoMFA method for the 5-HT(4) receptor of a series of benzimidazole-4-carboxamides and carboxylates derivatives. A subsequent computational simulation of ligand recognition has been successfully applied to explain the binding affinities. The CoMFA model shows high predictive ability. Steric and electrostatic fields and solvation energy of this novel class of 5-HT(4) receptor antagonists constitute the relevant descriptors for structure-activity relationships. Computational simulation of the complexes between a benzimidazole-4-carboxamide and a carboxylate derivative and a 3D model of the transmembrane domain of the 5-HT(4)R, constructed using the reported crystal structure of rhodopsin, has allowed to define the molecular details of the ligand-receptor interaction. Both the derived computational models have facilitated the identification of the structural elements of the ligands that are key to high 5-HT(4) receptor affinity. The combination of these two computational approaches provides the tools for predicting the affinity of new related compounds and for guiding the design of new ligands

3.5. Problem 5: Prediction of Drug Likeness

Recently, several groups have attempted to define drug likeness [40-42]. Indeed, there is much interest in the development and application of computational methods for predicting drug likeness. These methods can be applied to virtual compounds allowing an early elimination of poor candidates even before synthesis. I have chosen this topic to illustrate how some methods outlined in the theoretical background can be applied.

3.5.1. Solution 1

Sadowski and Kubinyi developed a feedforward neural network system with back-propagation of errors for discriminating between drugs and nondrugs [42]. Compounds were chosen from the WDI (World Drug Index) database and from the ACD (Available Chemical Directory) database as collection of drugs and non-drugs, respectively. Both databases were preprocessed to remove unwanted compounds [42]. The WDI and ACD compounds were assigned a drug likeness score of 1 for drugs and 0 for non-drugs. For training, two subsets of 5000 compounds were randomly extracted from both databases.

To be used as input for ANN, each compound should be translated into a suitable set of descriptors. Ghose and Crippen have developed a system of 120 atom type descriptors for predicting octanol/water partition coefficient [42]. In the study by Sadowski and Kubinyi, the counts of 92 Ghose and Crippen atom types within a molecule were used as molecular descriptors. This set of 92 descriptors is similar to a molecular fingerprint describing each compound and forming an appropriate input vector for an ANN.

A 92 • 5 • 1 feedforward neural network (92 input units, 5 hidden neurons and 1 output neuron) was trained with a training set of 10000 compounds. All layers were totally connected resulting in 465 (92 • 5 + 5 • 1) weights.

One of the possible classification techniques, a two-category or binary classification can be achieved either by two output neurons one for each class, or by one output neuron, which is set to '1' for one class (e.g., drugs) and to '0' for the other (e.g., non-drugs). The authors decide to work with one output neuron.

The output neuron produces an output score between 0 and 1. Drugs and non-drugs were separated according to a borderline that was set at the scoring value of 0.5. Once trained the neural network system was able to correctly classify 83% of the ACD compounds and 77% of the WDI compounds. The method is very fast and allows classifying hundreds of thousands of compounds in a few hours. For the screening purposes the threshold value separating drugs and non-drugs was set to 0.3 since the major concern was not to reject a potentially valuable compound.

3.5.2. Solution 2

Wagener and van Geerrestein applied a different classification technique yielding comparable results [41]. They developed a decision tree to distinguish between drugs and non-drugs using the same databases (WDI and ACD) and the same chemical descriptors (Ghose and Crippen data types) as Sadowski and Kubinyi. While offering the same accuracy of prediction (17.4% error rate in prediction), this method provides a list of structural features that are fundamental to the discrimination between drugs and non-drugs. The outcome of the RPA is a decision tree that consists of a set of nodes. Each step from the root (first node) to the leaf (terminal node) corresponds to a rule defining explicating the presence or absence of a specific Ghose and Crippen atom type. During training a decision tree is created. A single descriptor is identified that splits the entire training set into two similar subsets. Testing all possible partitioning and choosing the one that provides the best enrichment of the known classes achieved this. The resulting subsets are again split into subsets using different descriptors. At each node, a single feature of the chemical structure (e.g., presence of a Ghose and Crippen atom type: oxygen connected to an hydrogen) is tested. The procedure is continued until no further significant splits can be found. To improve the accuracy of the classification system, the authors decided to combine the prediction of several decision trees in a special voting procedure termed boosting.

Based on the analysis of the decision trees, Wagener and van Geerrestein were able to identify a few simple features that can explain the most significant differences between drugs and non-drugs. 75% of drugs can be correctly classified by testing the presence of hydroxyl, tertiary or secondary amino, carboxyl, phenol, or enol groups, while non-drugs are characterized by their aromatic nature and a low number of functional groups besides halogens.

3.5.3. Solution 3

Brüstle and colleagues have investigated three different techniques to address the problem of drug likeness, namely PCA, RPA and the Kohonen neural networks [12].

The drug data set was derived from the WDI database using a procedure designed to select only real drugs. The Maybridge database was selected as non-drug data set, although it was assumed to contain a subset of compounds that could make good drugs. All compounds were characterized by a set of molecular descriptors derived from AM1 semiempirical calculations. Based on the results of previous studies, 26 descriptors that were judged to provide a good description of physical property space were selected.

A PCA was performed on the compounds from Maybridge database to investigate the dimensionality of the physical property space. The number of PCs was selected based on two tests. According to the first test the number of PCs with Eigenvalues larger than one should be retained. The second test consists of the analysis of the scree plot (Eigenvalues of the PCs plotted against the number of PCs). In the scree plot, the Eigenvalues of the PCs can be plotted on a line graph. Since the first PCs account for more variance than the last PCs, the plotted line has a negative slope. The scree test stipulates that contribution of the significant PCs should stop when this line flattens out. Based on these criteria, it was possible to conclude that the space described by the 26 semiempirical descriptors for the Maybridge database has 7-8 dimensions.

Looking at the loadings, it was possible to define the nature of the PCs. The first PC separates compounds on the basis of their size and shape; the second PC constitutes an electrostatic description of the positive zones of compounds, while the third PC represents an electrostatic description of the negative areas of molecules. The PC4 and PC5 discriminate compounds according to the hydrogen bond donor and acceptor properties, respectively. The PC6 reveals the polarity of compounds.

Using histograms of percentage of frequencies of each PC for the Maybridge and drug data sets it was possible to identify which PCs can discriminate between drugs and non-drugs. Only the third PC of the 26 descriptors can distinguish between drugs and non-drugs and thus can be used as a numerical index of drug likeness.

It is important to define the nature of all the significant PCs because it is possible that the first PC does not represented the property we are interested in, but it a PC that still explains a certain amount of variance should represented it.

The set of theoretical parameters was extended from 26 to 66 and RPA was used to identify the descriptors that are more relevant in the discrimination between drugs and non-drugs. Following this procedure, it was possible to detect 2 descriptors that were strongly present in the PC3, and a third one descriptor that was not part of the first 26 parameters. This parameter is defined as the difference between the energy of the lowest unoccupied molecular orbital and the highest occupied molecular orbital and can be related to the characteristics of the strongest hydrogen bond acceptor. The authors did not apply a supervised method to discriminate between drugs and non-drugs since they considered the non-drug data set to be contaminated by drugs. Instead they used RPA to select suitable descriptors for the Kohonen neural network approach. The three descriptors retained from RPA procedure were used to train a Kohonen network.

Different network architectures were tested and the 100 • 100 • 3 architecture was adopted. As a result of training of the Kohonen network, a given compound can be qualitatively classified as drug or non-drug by determining which neuron the compound is assigned to. Indeed, drugs form a cluster in the Kohonen map suggesting that this approach can efficiently recognize potential drugs. As shown by the results of this study, the Kohonen neural networks are useful tools for the analysis and visualization of large multivariate data sets.

4. CONCLUSIONS

The techniques described in this review can provide useful tools to establish relationships between chemical descriptors and physicochemical properties or biological activities. New methods in computational chemistry and chemometrics are in continuous development to better understand data structure, improve data classification and obtain more robust data models.

There is no single correct way to analyze any data set, as application of different techniques reveal different aspects of the data. Unfortunately, there can be incorrect ways to analyze certain data sets, therefore it is important to consider the hypotheses assumed by different methods in relation to the data set we want to analyze. The results of any data analysis have to be evaluated in their chemical and biological context. In drug design the results from multivariate data analysis should be considered as a working hypothesis to be proven or rejected by the design and testing of new compounds.

Clearly, development, implementation and correct application of multivariate methods provide a major contribution to the long and difficult process of drug discovery.

5. ACKNOWLEDGEMENTS

I thank Dr. Alexei A. Adzhubei for his helpful contribution and useful discussions.

6. REFERENCES

[1] Winkler, D.A. *Briefings in Bioinformatics,* **2002**, *3*, 73

[2] Jolliffe, I.T. *Principal Component Analysis,* Springer-Verlag: New York, **1986**.

[3] Eriksson, L., Johansson, E., Kettaneh-Wold, N., Wold, S. *Introduction to Multi- and Megavariate Data Analysis using Projection Methods (PCA & PLS),* Umea, Sweden, Umetrics AB, **1999**.

[4] Wold, S., Esbensen, K., Geladi, P. *Chemometrics and Intelligent Laboratory Systems,* **1987**, *2*, 37.

[5] Hair, J.F., Anderson, R.E., Tatham, R.L., Black, W.C. *Multivariate Data Analysis,* Prentice Hall: New Jersey, **1998**.

[6] Bishop, C.M. *Neural Network for Pattern Recognition,* Oxford University Press, **1995**.

[7] Kubinyi, H. In *Computer-Assisted Lead Finding and Optimization: Current Tools for Medicinal Chemistry*; van de Waterbeemd, H., Testa, B., Folkers, Ed.; Wiley-VCH, **1997**, Ch. 1, pp 7-28.

[8] Everitt, B.S., Landau, S., Leese, M. *Cluster Analysis,* Edward Arnold, **2001**.

[9] Zupan, J., Gasteiger, J. *Neural Networks in Chemistry and Drug Design,* Wiley-VCH: Weinheim, **1999**.

[10] Gasteiger, J., Li, X., Rudolph, C., Sadowzki, J., Zupan, J. *J. Am. Chem. Soc.,* **1994**, *116*, 4608.

[11] Gasteiger, J., Li, X. *Angew. Chem. Int. Ed. Engl.,* **1994**, *33*, 643.

[12] Brustle, M., Beck, B., Schindler, T., King, W., Mitchell, T., Clark, T. *J. Med. Chem.,* **2002**, *45*, 3345.

[13] Tetko, I.V., Kovalishyn, V.V., Livingstone, D.J. *J. Med. Chem.,* **2001**, *44*, 2411.

[14] Anzali, S., Gasteiger, J., Holzgrabe, U., Polanski, J., Sadowzki, J., Teckentrup, A., Wagener, M. *Perspect. Drug Discovery Des.,* **1998**, *273*, 9-11.

[15] Sharaf, M.A., Illman, D.L., Kowalski, B.R. *Chemometrics,* John Wiley & Sons: New York, CA, **1986**.

[16] Wold, S. In *Chemometrics: Mathematics and Statistics in Chemistry*; Kowalski, B.R., Ed.; D. Reidel Publishing Company, Dordrecht, Holland, **1984**, pp 7-28.

[17] Coomans, D., Broeckaert, I., Derde, M.P., Tassin, A., Massart, D.L., Wold, S. *Comp. Biomed. Res.,* **1984**, *17*, 1.

[18] Wold, S., Eriksson, L., Sjöström, M. In *Encyclopedia of Computational Chemistry*; Von R. Schleyer, P., Ed.; Wiley: New York, **1998**, pp 7-28.

[19] Ståhle, L., Wold, S. *J. Chemometrics.,* **1987**, *1*, 185.

[20] Tetko, I.V., Tanchuk, V.Y., Kasheva, T.N., Villa, A.E. *J. Chem. Inf. Comput. Sci.*, **2001**, *41*, 1488.

[21] Tetko, I.V., Tanchuk, V.Y. *J. Chem. Inf. Comput. Sci.*, **2002**, *42*, 1136.

[22] Hawkins, D.M., Young, S.S., Rusinko, A. *Quant.Struct.-Act.Relat.,* **1997**, *16*, 296.

[23] Chen, X., Rusinko, A, Young, S.S. *J. Chem. Inf. Comput. Sci.,* **1998**, *38*, 1054.

[24] Breiman, L., Friedman, J.H., Olshen, R.A., Stone C.J. *Classification and Regression Trees,* Wadsworth Int. Group: Belmont, CA, **1984**.

[25] Young, S.S., Hawkins, D.M. *SAR QSAR in Environ. Res.,* **1998**, *8*, 183.

[26] Taylor, R., Mullier, G.W., Sexton, G.J. *J. Mol. Graphics*, **1992**, *10*, 152.

[27] Ermondi, G., Caron, G., Bouchard, G., Plemper van Balen, G., Pagliara, A., Grandi, T., Carrupt, P.-A., Fruttero, R., Testa, B. *Helv. Chim. Acta,* **2001**, *84*, 360.

[28] James, C.A., Weininger, D., Delany, J. Y.C. *Daylight Theory Manual Daylight 4.71* Daylight Chemical Information Systems, Inc. of Mission Viejo, CA, **2000**.

[29] Shi, L.M., Fan, Y, Lee, J.K., Waltham, M., Andrews, D.T., Scherf, U., Paull, K.D., Weistein, J.N. *J. Chem. Inf. Comput. Sci.,* **2000**, *40*, 367.

[30] Lombardo, F., Blake, J.F., Curatolo, W.J. *J. Med. Chem.,* **1996**, *39*, 4750.

[31] Norinder, U., Sjöberg, P., Osterberg, T. W.J. *J. Pharm. Sci.,* **1998**, *87*, 952.

[32] Kelder,J., Grootenhuis, P.D.J., Bayada, D.M., Delbressine, L.P.C., Ploeman, J.-P. *Pharm. Res.,* **1999**, *16*, 1514.

[33] Luco J.M. *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 396.

[34] Crivori, P., Cruciani, G., Carrupt, P.-A., Testa, B. *J. Med. Chem.,* **2000**, *43*, 2204.

[35] Cruciani, G., Crivori, P., Carrupt, P.-A., Testa, B. *Theochem.,* **2000**, *17*, 503.

[36] Kubinyi, H. *3D QSAR in Drug Design. Theory, Methods and Applications*, ESCOM Science Publisher, Leiden, **1993**.

[37] Kubinyi, H., Folkers, G., Martin, Y.C. 3D QSAR in Drug Design. Volume 2: Ligand-Protein Interactions and Molecular Similarity, Kluwer Academic Publishers, **1997**.

[38] Kubinyi, H., Folkers, G., Martin, Y.C. *3D QSAR in Drug Design. Volume 3: Recent Advances,* Kluwer Academic Publishers, **1998**.

[39] Lopez-Rodriguez, M.L., Murcia, M., Benhamu, B., Viso, A., Campillo M., Pardo L. *J. Med. Chem.*, **2002**, *45*, 4806.

[40] Sadowski, J., Kubinyi, H. *J. Med. Chem.,* **1998**, *41*, 3325.

[41] Wagener, M., van Geerestein, V.J. *J. Chem. Inf. Comput. Sci.,* **2000**, *40*, 280.

[42] Viswanadhan, V.N., Ghose, A.K., Revankar, G.R., Robins, R.K. *J. Chem. Inf. Comput. Sci.,* **1989**, *29*, 163.