RSC Advances

ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/advances

Journal Name

ARTICLE

Multivariate statistical analysis methods in QSAR

SomayehPirhadi[a], Fereshteh Shiri[b*], Jahan B. Ghasemi[a,c]

[a]Drug Design in Silico Lab., Chemistry Faculty, K N Toosi University of Technology, Tehran, Iran[b]Department of Chemistry, University of Zabol, P.O. Box 98615-538, Zabol, Iran
[c]Drug Design in Silico Lab., Chemistry Faculty, University of Tehran, Tehran, Iran

Abstract

The emphasis of this review is particularly on multivariate statistical methods currently used in quantitative structure-activity relationship (QSAR)studies.The mathematical methods for constructing QSAR include linear and non-linear methods that solve regression and classification problems in data structure. The most widely used methods for the classification or pattern recognition; are principal component analysis (PCA) and hierarchical cluster analysis (HCA) as the exploratory data analysis methods. The regression analysis tools are artificial neural network (ANN), principal component regression (PCR), partial least squares (PLS) and classification and regression tree (CART). Also some pattern recognition approaches of k nearest neighbor (kNN), the soft independent modelling of class analogy (SIMCA) and support vector machines (SVM) have been described .Furthermore, different applications wererepresented for further characterization of these techniques.

Key words: Multivariate analysis, Regression, Classification, principal component analysis, KNN, PLS.

*Corresponding author:
E-mail address: fereshteh.shiri@gmail.com and Fereshteh.shiri@uoz.ac.ir , Department of Chemistry, University of Zabol, P.O. Box 98615-538, Zabol, Iran. Tel: 0985424822186, Fax: 0985424822180

Introduction

QSAR [1-4] attemptsto find a simple quantitative equation that can be applied to predict some property from the molecular structure of a compound. Results of measurements on a number of objects (descriptors and biological activity) are usually arranged in a matrix, which is called a two-way multivariate data table. Multivariate statistical methods are needed to understand of multidimensional data in its entirety. This equation relatesthe biological effects (i.e. the activity) and the chemistry (i.e. the structure) of each of the chemicals. A descriptor or molecular property is any number that describes the molecule. Each molecular descriptor is able to account a small part of the whole chemical information contained in the real molecule. Representing molecules in the space formed by these numbers (molecular descriptors or properties) result in the multivariate chemical space. There are a large number of molecular descriptors and structural fingerprints that have been widely used in substructure/similarity searching, clustering, classification, and other statistical learning approaches of toxicological and pharmacological properties of samples [5, 6]. The aim of applying variable selection methods is to optimally selecting a subset of molecular features that include necessary information, decrease noise and remove redundant descriptors which are not relevant to the activity prediction [7, 8]. Then, one would be able to develop more accurate and efficient computational tools and it can help clarifying the relationship between the structure of a compound and its biological activity. Statistical models attempts to find the equation parameters, which are weights of known descriptors. Molecular descriptors have a key role in chemistry, toxicology, risk assessment, ecotoxicology, pharmaceutical sciences, environmental protection policy, health research, and quality control. Any QSAR model development has an iterative nature, until enough information about a class of compounds has been achieved so as to either design compounds with the preferred activity profile, or to conclude that such a profile cannot be reached. The following steps can be regarded: First is to define the problem, i.e. selection of the biological activities of interest, choice of structural domain (structural class) and the choice of structural features to be varied. The second one is a quantitative description of the structural variation. The third is choosing the type of the model for the QSAR, i.e. a linear, quadratic polynomial, hyperbolic or exponential model, etc. It follows the selection of compounds (series design), and further stage is synthesising and biological testing. Data analysis, and

validation, can be the next that interpretation of results, and proposal of new compounds is situated at the end.

Effectiveness of a QSAR modelling and compound classification tool depends on several, sometimes conflicting, requirements in modern drug discovery and development processes. The complexity of these necessities has been quickly growing in recent years. Some of them are able to handle a vast number and diverse types of descriptors, and molecular diversity (e.g., multiple mechanisms of action). Also, they have high accuracy of prediction, model interpretability, and computational efficiency [9]. The major objective of QSAR [10] analyses is to develop predictive models which have applications in computer-aided drug discovery and many other fields. A prerequisite of applying a model in external predictions is to establish and validate its predictive power. Model internal validation is usually performed using leave one out (LOO) cross-validation which yields an overoptimistic estimate of predictive ability [11]. External validation is used now as a "must have" tool to evaluate the reliability of QSAR models [1]. In this method, typically the total data set is randomly split into a training set and a test set. Training set is used to develop QSAR models and then are used to predict activity of the test set. This approach has two faces. The advantageous part is that the test set compounds are "new" to the models as were excluded from the model development procedure, especially from the variable selection. The disadvantageous part is that random dividing the overall set has not any rationale for selecting test set chemicals. So, the rational division algorithms such as Kohonen Self- Organizing Map method [12], the Kennard-Stone method [13], have been created which may be able to "intelligently" split data sets into training and test sets to produce more predictive models. The rational splitting of data into training and test sets is more important for smaller data sets than large ones. Random selection works well for large data sets but causes significant instability for small data sets, especially if there are outliers. Previous studies have shown that the rational division algorithms does better to the simple random splitting and activity sorting methods [14, 15].Although it is not clear which rational division method is the most useful due to conflicting results [16, 17]. Interpretation of created models gives a viewpoint of the chemical space in proximity of the hit compound. In the drug discovery pipeline, precise QSAR models built on the basis of the lead series can aid in optimizing the lead structures [18].

In this review, we discuss several important and the most used methods of multivariate statistical analyses in QSAR. These approaches enclose linear and nonlinear regression methods, and supervised and unsupervised pattern recognition approaches in developing QSAR models together with presentation of useful practical applications.

Background

The appearance of QSAR is on the basis of the assumption expressed more than a century ago by Crum- Brown and Fraser (1868) that states that a substance acts as a function of its chemical structure. This resulted in the idea that similar structures show similar biological properties and a small change in a chemical structure is along with a proportionally small shift in biological activity [19].

Within the following hundred years, attempts of researchers were to formalize some of those relationships. Richardson (1868) expressed that the toxicities of ethers and alcohols has an inverse relation to their water solubilities. Richet (1893) established a relationship between the narcotic effect of alcohols and their molecular weight.Overton (1897) and Meyer (1899) independently demonstrated that the narcotic action of a lot of compounds was dependent on their oil/water partition coefficients [20].

In 1935, Louis Hammett tried to describe the relationship between structure and property. He proposed a correlation between reaction rates of para- and meta-substituted derivatives of benzoic acid in alkaline hydrolysis and the positive type of substituents (i.e. the changes in equilibrium constant by substitution) eq. 1.

$$\mathrm{Log}\frac{k}{K_0} = \sigma\rho (1)$$

Equation (1) is the Hammett free-energy relationship, where k is the reaction rate constant, σ is substituent constant, and ρ is the reaction constant. The above relation is applied as a reference and ρ=1, so the σ values for a host of substituents can be determined and later on, used to different reactions whereby ρ can be computed.

Later modifications and improvements to this approach were performed by Hansch, and Fujita et al. At the time of short collaboration of Hansch and Fujita in the early 1960s, they presented timeless guidelines as to how to translate differences in chemical structures into those features that relate to differences in their biological properties. In reality, the foundation of QSAR as a practical tool of drug design led by the pioneering works of Hansch and Fujita in the mid-1960s [21, 22]. Hansch, Fujita et al. [23] in 1962, published their study on the structure–activity relationships of plant growth regulators and their dependency on Hammett constants and hydrophobicity. The delineation of Hansch models resulted in explosive development in QSAR analysis and related methods. In the same years, Free and Wilson (Free & Wilson, 1964) developed a model of additive substituent contributions to biological activities, giving a further push to the development of QSAR strategies. At the end of 1960s, a lot of structure–property relationships were proposed both on substituent effects and indices demonstrating the whole molecular structure. Derivation of these theoretical indices was from

topological representation of molecule, mostly applying the graph theory meanings, and then usually called 2D descriptors. Balaban [24], Randic [25, 26], and Kier et al. [27] did fundamental works led to further significant developments of the QSAR approaches based on topological indices (TIs). Since then, along with the increasing knowledge in chemistry and the power enhancement of computers, researchers are developing and applying more and more QSAR/QSPR models. QSAR/QSPR development is now an important division of chemometrics that is the science of the application of mathematical or statistical methods to chemical data. Linear multivariate methods such as principal components analysis (PCA) have now wide applications in medicinal chemistry and related fields for developing structure-activity relationships (SAR). In addition to linear methods, non-linear methods can provide useful information about the relationships in the data. Some of non-linear methods available for multivariate data analysis, are very beneficial for the reduction of dimensionality and visualization of multivariate data [28].The aim of developing a QSAR equation is to realize a correlation between biological and chemical data attained by multiple linear regression (MLR) [29], sometimes also called ordinary least squares (OLS) [30]. Techniques such as principal component regression (PCR) [31] and partial least squares (PLS) [32] are called latent variable based techniques. MLR is considered as a "hard" model, while SIMCA (soft independent modelling of class analogy) and PLS are regarded so as to "soft" modelling techniques [33]. Generally, QSAR modelling tools provided from statistics, machine learning, etc. has traditionally satisfied demands. Some of familiar examples comprise decision tree (DT, or recursive partitioning) [34], artificial neural network (ANN) [35], PLS, k-nearest neighbours (kNN) [36], linear discriminant analysis (LDA) [37], and support vector machines (SVM) [38]. In general, linear methods such as PLS, MLR, and LDA may not be appropriate for dealing with multiple mechanisms of action.

Following the text, two general categories have been selected for the methods: pattern recognition, and regression methods. Although, several approaches are able to modelling both clustering and regression problems like PLS and ANN. Herein, pattern recognition methods include HCA, PCA, LDA and SIMCA. The other approaches can solve regression alone or plus pattern recognition problems.

Pattern recognition methods

Hierarchical cluster analysis

Depending on the existence of a training set, a pattern recognition technique can be either supervised or unsupervised. Exploratory data analysis (EDA) and unsupervised pattern recognition are methods commonly applied to make simpler and gain better overview of data structures. The major EDA technique is PCA. Other unsupervised pattern recognition approaches can be used for preliminary evaluation of the information contents in the data tables, such as cluster analysis (CA). Clustering or cluster analysis is used to natural grouping of samples in clusters that are not known beforehand, with a common property characterized by the values of a set of variables. It is therefore an alternative to PCA for describing the structure of a data table. In CA, samples are grouped based on similarities without regarding the information about the class membership. In QSAR modelling, it can be used to check out the homogeneity of data, identify some unusual data points, detect patterns, and represent potentially interesting relationships in the data. Because of the effect of different scales of the variables, a pre-processing of the data is required. CA groups objects consistent with a similarity metric, which can be distance, correlation or some hybrid of both. This method is on the basis of idea that the similarity has an inverse relation to the distance between samples. So, In CA, the distances (or correlation) between all samples are computed using a certain metric such as Euclidean distance, Manhattan distance, etc. Different clustering algorithms can be used to grouping of the samples, depending on the criteria considered to define the distance between two groups (linkage criterion): single (nearest neighbour), complete (furthest neighbour) or average linkages, centroid method, Ward's method, etc. There are two methods of representing the data by clustering [39]. First is depiction by the tree, in the form of a dendrogram, in a hierarchical clustering of objects, Figure 1. Its primary aim is to represent the data so as to emphasize its natural clusters and patterns. The calculated distances between samples are transformed into a similarity matrix which its elements are similarity indexes. The linkage rule specifies the distance between observations as a function of each two distances between samples. Cutting the tree is carried out at a level where a partitioning will result a clustering at a selected precision. The second approach of CA is to make a table including different clusterings. This table does not essentially yield a complete hierarchy thus, the indication is called non-hierarchical. There are two types of hierarchical clustering. In the agglomerative manner or "bottom up" approach, each object begins in its own cluster, and pairs of clusters are merged as one goes up the hierarchy. In divisive way, or "top down" approach, beginning of all observations is in one cluster, and divisions are carried out recursively as one goes down the hierarchy. If a CA is applied on a data matrix, a set of clusters can always be obtained, even without existing any actual grouping of the objects. Thus, a validation step seems to be necessary. There is a huge literature on validity of clusters. One is based on the permutation testing to see if there isreally a non-random tendency for the objects to be grouped together. In QSAR modelling, when data samples are diverse, HCA can be performed to manifest the natural clustering in the data set to build separate models [40].Overall, the main feature of CA can be regarded as grouping pattern depends on the

distance measure and linkage rule, and the result should be handled with care.

Principal Component Analysis

PCA is a technique of identifying patterns in data, and expressing data in such a way as to emphasize their similarities and differences. It is also likely to be the oldest and the most popular method in multivariate analysis. PCA is a useful data compression technique, by reducing the number of dimensions, without much loss of information that has found applications in fields such as outlier detection, regression and is a common technique for finding patterns in data of high dimension [41, 42]. Its aim is to represent of multivariate data into a low-dimensional space spanned by new orthogonal variables called principal components, PCs, which are obtained as linear combinations of the initial variables by maximizing the description of data variance. The first principal component, PC1, is oriented in the direction of maximum variance of the original data set. PC2 is defined orthogonal to the first PC to describe the remained maximum variance. All subsequent components are calculated orthogonal to the previously chosen ones and contain the maximum of leftover variance. Most of the information gathered in the data set is described by only the first few components and noise and redundancy is removed. For PCA to work properly, it needs to subtract the mean from each of the data dimensions. There are three most often used numerical approaches for PCA decomposition, besides of different names are fully equivalent: PCA, singular-value decomposition (SVD), and eigenvector–eigenvalue extraction. For any centred data matrix **X** (*m, n*), corresponding to m samples and n descriptors, the PCA decomposition can be presented as follows:

$$\mathbf{X} = \mathbf{TP}^{\mathbf{T}} \quad (2)$$

Orthogonal projection onto a specific PC results in a 'score' for each object (Figure 2). The matrix **T** (m, r), is the matrix containing the data scores, represents the position of the compounds in the new coordinate system formed by PCs in the axes. **P** is known as loading matrix of dimension (n, r) that r is the mathematical rank of the data that is equal to min (m, n). Columns of P describe formation of PCs from original variables (old axes).

The correlation between a component and a variable estimates the information they share and is called loading. The number of original variables, n, is usually much more than significant components, which represent degrees of data compression. With the higher degree of correlation among the original variables the compression of the studied data set will be better, which outcomes in a smaller number of significant principal components. Cross-validation procedure such as the bootstrap and the jackknife can be applied to define the number of significant PCs [42]. Mathematically, PCA depends upon the eigen-decomposition of positive semi-definite matrices and upon the singular value decomposition (SVD) of rectangular matrices:

$$\mathbf{X} = \mathbf{USV}^{\mathbf{T}} \quad (3)$$

SVD decomposes X into three matrixes of **U**, **S,** and **V**, where **U** (m × r),and **V** (n × r) are orthogonal eigenvector square matrices and **S** (r × r) is a diagonal matrix containing the singular values (equal to the square root of the eigenvalues). **T** is generated by multiplying **U** and **S**, and **V** is the loading matrix **P**. Each eigenvalue represents the amount of variance in the initial data explained by the corresponding principle component. The total data variance is shown by total sum of eigenvalues. The eigenvector with the highest eigenvalue is the principle component of the data set. In QSAR studies, T matrix represents information about the objects, while the loading matrix, P, gives information about the original molecular descriptors. Outliers are observations that appear to depart from the bulk of the major part of data. Outliers can be accommodated or rejected in the modelling process. If accommodation is chosen, robust estimation methods are necessary for the model building. Robust estimation reduces the influence of outlying observations in the model. Least squares techniques are not robust against outliers. Most multivariate methods applied to chemical data are based on least squares (LS) techniques. PCA similar to any least square method is very sensitive to outliers. Outliers could change the direction of principal components and cause in the model inaccuracy (Figure 3). Outliers are observations that appear to depart from the bulk of the major part of data. Usually they do not fit the model, or are weakly predicted by it [43]. However, developing classic least squares models including outliers will likely produce instable models [44]. Presence of outliers from a QSAR could be related to many reasons. Typically, however, the deviating behaviour of such compounds can be real, such as a different mechanism of action from other compounds which are well modelled by the model, or due to errors in structure representation or biological activity annotation [45]. The robust PCA methods aim to obtain principal components that are not much affected by outliers. Many methods have been proposed to gain this goal. These methods are generally based on three different approaches: (i) techniques using a robust covariance matrix to take a set of robust eigenvectors and eigenvalues. (ii) methods generally based on projection pursuit that directly provide robust estimates of eigenvectors and eigenvalues and do not need to obtain the robust estimate of the covariance matrix, (iii) a hybrid of both.

Many techniques have been proposed based on projection pursuit (PP), [46]. The type of projection index used is the main difference among them. ROBPCA is a popular robust PCA technique that combines ideas of both projection pursuit and robust covariance estimation of data location and covariance in a low-dimensional space. This method is also very well suited for the analysis of high-dimensional data and has wide

applications to multivariate calibration and classification problems. (i) perform PCA for preliminary data dimensionality reduction; (ii) compute the being outlier measure (i.e., the projection index) for every object and construct the initial H-subset (H0) containing h objects with the smallest being outlier measure (the choice of h determines the robustness of the method and its efficiency; the default value of h is set to 75% of the total number of objects in the data); (iii) perform a further data dimensionality reduction by projecting the data onto k-dimensional subspace spanned by the first k eigenvectors of the empirical covariance matrix obtained for objects in H0; (iv) compute the robust data centre and covariance in the k-dimensional subspace and apply the re-weighted MCD estimator to the projected data.

Kernel PCA as a non-linear extension of PCA (using the kernel trick) has proven influential as a preprocessing step for classification algorithms. It can also be considered as a non linear feature selection. Obviously, besides of abilities of linear PCA, it cannot always perceive all structure in a given data set. However, by the use of useful nonlinear features, it is possible to extract more information. Kernel PCA is very well suited to extract interesting nonlinear structures in the data [47].Kernel PCA first maps the data into some feature space F via a (usually nonlinear) function ϕ and then performs linear PCA on the mapped data.

In a study a data set contains a set of 2548 compounds reported as P-glycoprotein (P-gp) inhibitors and non-inhibitors was gathered from two literature sources [48].Threshold values for inhibitors and non-inhibitors were set based on the IC_{50} values and on the percentage of inhibition. Compounds with an IC_{50} ≤15 μM, or >25−30% of inhibition were regarded as inhibitors while, compounds bearing IC_{50} and % of inhibition values of ≥100 μM or <10−12% were categorized as non-inhibitors. Also, the authors added 3D structures of 797 inhibitors and 476 non-inhibitors from a published data set. They used MDRR (multidrug-resistance ratio) values measured in adriamycin-resistant P388 murine leukemia cells for classification. With MDRR values greater than 0.5 compounds were assigned inhibitors, whereas molecules with lower or equal to 0.4 MDRR values were categorized as non-inhibitors. After some preprocessing on compounds such as elimination of duplicated structures, a data set of 1608 compounds, comprising 1076 inhibitors and 532 non-inhibitors was remained. A binary variable (1 for inhibitor, 0 for non-inhibitor) was used to indicate the activity of the compounds. The data set was divided into training and test set using D-optimal onion design (DOOD) that resulted in 1201 training (841 inhibitors, 360 non-inhibitors) and 407 test compounds (235 inhibitors, 172 non-inhibitors) (internal test set). The calculated descriptors were 62 2D descriptors implemented in MOE. They comprised physicochemical properties, atom and bond counts, and pharmacophoric features. In addition, a set of 166 MACCS fingerprints and a set of 307 substructure fingerprints were generated. A PCA on the whole data set was carried out using the software SIMCA-p. A set of representative machine learning methods such as SVM, kNN, DT, RF, and BQSAR was used for ligand-based classification. PCA was conducted to examine potential clusters and the coverage of the chemical space of the P-gp ligands. The first two principal components explained71.7% of the variance in the data set. A scatter plot is presented, in Figure 4a that represents the distribution of the compounds according the first two principal components. A distinct cluster of inhibitors at the right top corner could be seen, which mainly included cyclopeptolide derivatives, which are chemically different from the rest of compounds. Furthermore, there was quite a fine separation between inhibitors and non-inhibitors, which motivated for the development of classification models. In order to figure out the effect of the descriptors on the first two PCs, the loading plot was analyzed (Figure 4b). Majority of the inhibitors are highly influenced by the descriptors that hold hydrophobic information, e.g. the number of aromatic bonds or the partition coefficient (logPo/w). Additionally, the high contribution of LogS to non-inhibitors demonstrates that non-inhibitors have more hydrophilic properties than inhibitors. The hydrophobic requisite of P-gp inhibitors can be explained by the need of diffusing through the cell membrane in order to effectively bind to the hydrophobic active site of the protein. Applicability domain (AD) analysis was performed on the basis of the ED approach. Also an additional AD experiment was performed using a set of 986 FDA approved drugs extracted from DrugBank. The scoring plot of the first two principal components obtained by PCA of the FDA approved drugs, as well as the training and both test sets is provided in It had been observed that 973 compounds were in the domain of the training set and only a small amount (13 compounds) of the FDA drugs were situated outside. The scoring plot of the first two principal components obtained by PCA of the FDA approved drugs, as well as the training and both test sets is provided in Figure 5.

There are also many other applications of PCA in different fields from drug design to toxicity assessments. PCA also can perform variable reduction by retaining the variables containing the largest loadings in the initial components and or deleting variables with the largest loadings in the last components [49, 50].

Linear discriminant analysis (LDA)

Linear discriminant analysis (LDA), initially proposed by Fisher in 1936 [37], is a supervised pattern recognition method and one of the oldest and most studied ones. From a probabilistic point of view, it is a parametric method, because of its basic hypothesis that is in each category the data follow a multivariate normal distribution. An assumption in LDA is that the dispersion of the observations for all the classes is the

same, that is, the variance/covariance matrices of the different categories are equal. It mostly has applications in classification problems and also for dimensionality reduction. It works on data that has categorical target properties and molecular descriptors that are continuous variables. As a linear technique, its mission is to find the decision boundaries separating the classes of a target property in the multidimensional space of the variables with linear surfaces (hyperplanes). In LDA, while between-class variance is maximized the within-class variance is minimized. In the case of two classes where only two variables are measured, the best straight line in two dimensions separates the associated regions corresponding to each class. The best hyperplane is defined by a linear discriminant function which is a linear combination of molecular descriptors:

$$\boldsymbol{L} = \sum_{i=1}^{k} w_i x_i (4)$$

where L is the discriminant score (function) or canonical variate, x_i,. . .,x_k are the independent variables and their corresponding weights are w_i,. . ., w_k. As depicted in Figure 6, L represents the function where a set of data points is projected onto. This function is achieved by means of optimizing the weights, w_i, to maximize the ratio of the between-class to the within-class variance to get the largest class divisions. Equation 26evidences that there is a linear combination of the variables to define a latent vector, suchas in the case of PCA or PLS regression. However, while PCA selects directions with maximal structure among the data in a lower dimension, LDA recognizes directions which maximize the separation among the given classes. Determining the class of a test compound is performed by calculating the Mahalanobis distance of the chemical from the gravity center of each class [51, 52]. The Mahalanobis distance between an independent variable ($\mathbf{x}_i$) and the data center ($\bar{\mathbf{x}}$) is defined as

$$D(x_i) = [(x_i - \bar{x})^{\mathrm{T}} (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} (x_i - \bar{x})]^{0.5} (5)$$

wherei is the index of a specific chemical and $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ is the covariance matrix. The center is determined by the arithmetic mean vector. When the test compound is located nearest to the gravity center of its actual class, then it is properly classified. Otherwis, it would be incorrectly assigned to the other class with the smallest Mahalanobis distance.

Quadratic discriminant analysis (QDA) is similar to LDA and establishes parabolic boundaries in which for each class a different covariance matrix is possible. In QDA, distribution of objects in the space is less subjected to limitations compared to LDA. It is also possible to consider a method due to Jerry Friedman known as regularized discriminant analysis (RDA). RDA can be thought sort of a trade-off between LDA and QDA. RDA shrinks the separate covariances of QDA towards a common covariance as is in LDA. RDA can be considered as being the most general of the three methods.

There are two concepts to examine the performance of a QSAR-based classification model: sensitivity and specificity [53]. The ability of the model to detect known active and non-active compounds are called sensitivity and specificity respectively. The sensitivity represents the percent of the chemicals tested positive that are identified correctly as positive by the QSAR model. Thus, a model showing a high sensitivity has consequently a high true positive rate and a low false negative rate. The percent of the chemicals tested negative and are correctly identified as negative by the QSAR model is specificity. A high specificity of a model shows a high true negative rate and also a low false positive rate. To investigate the classifier model performance, several statistical tests have been used by scientists. Such tests involve plotting of receiver operating characteristic (ROC) curve and computation of classification accuracy, F-measure and other indices [54]. ROC curve is a tool that is used to compare the performance of different classification models. In this graph, the x-axis is false positive rate (specificity) and the Y-axis is true positive rate (sensitivity). A model in its best possible condition namely a high true positive rate and a low false positive rate would yield a point in the upper left corner of the ROC. On the other hand, when the model is not discriminating a straight line at an angle of 45 degrees from the horizontal line, i. e. equal rates of true and false positives would obtain [55]. F-measure mentions the harmonic mean of recall and precision. Recall assigns the accuracy of real prediction and precision refers to the accuracy of an obtained class. Recall and precision with higher values implicate higher F-measure value which consequently represents better classification ability of the model [56].

Soft independent modelling of class analogy (SIMCA)

This model, proposed by Wold et al. in 1976, was the first class-modelling method introduced in the literature [57]. It is a supervised soft modelling method. This method runs a whole principal component analysis or PLS regression on the entire dataset so as to recognize groups of samples. A SIMCA model contains a set of PCA models, one for each class in the data set. Local models are then determined for each class. Only the significant components are retained. The number of components in each class is determined by cross-validation. There may be a different number of principal components for each class. The number depends on the data in the class. The model is completed by defining boundary regions for each PCA model. This is shown graphically in Figure 7. Each PCA sub-model contains all the usual pieces of a PCA model: mean vector, scaling information, pre-processing such as smoothing and derivatizing, etc. In the construction of the SIMCA classification model, the standard deviation of residues is calculated for each class separately. In addition, variances of samples in each axis for the space described by the principal components are computed. These two measures are applied for the classification of new samples [58]. An object is put in a class on the basis of the residual distance, rsd^2, from the model which is representing the class itself:

$$r_{igi}^2 = (\hat{x}_{igi} - x_{igi})^2 \quad (6)$$

$$rsd^2 = \frac{\sum_j r_{igi}^2}{(p-M_j)} l \sqrt{\sum_{j=1}^{p} (x_{sj} - x_{tj})^2} (7)$$

where $\hat{x}_{igi}$is coordinates of the object's projections on the inner space of the mathematical model for the class, x_{igi} is object's coordinates, p is the number of variables, and M_j is the number of PCs significant for the j class [59]. The residuals from the model can be computed from the scores on the non-retained eigenvectors.Each new sample is classified to one of the built class models based on the best fit to the related model. The classes present in the training set should consist of as many objects as representing the diversity of the class. Any new object can belong to one of the classes, or may be an outlier to all the classes. Outliers are defined as observations outside the class boundaries. Aliens are objects inside the boundaries that do not fit in to the class. An object may be a member of more than one class if the classes are overlapping. For a specified class, the model then represents a line (for one PC), plane (for two PCs) or hyper-plane (for more than two PCs).

An attractive aspect of SIMCA is that a principal component mapping of the data has been done. Therefore, samples that may be described by many variables are mapped onto a much lower dimensional subspace to be classified. When an object is similar to the other objects in a class, it will locate near them in the PC map described by the samples forming that class. Another benefit of SIMCA is that a test sample is only belonged to the class for which, there is a high probability. While the residual variance of an observation exceeds the upper limit for any modelled class in the data set, the sample is not assigned to any of the classes owing to the fact that it is either an outlier or belongs to a class that is not represented in the data set. In addition, there would be a sensitivity for SIMCA to the quality of the data used to generate the PC models. So, there are recognitions to measure the quality of the data, for instance the modelling power and the discriminatory power. The modelling power delineates importance of a variable to incorporate the PCs to model variation, and discriminatory power describes how well the variable helps the PCs to classify the samples in the data set. If a variable contributes only noise to the PC models it would cause low modelling power and low discriminatory power then usually is deleted from the data. An important consideration in SIMCA is that the number of samples might be as few as 10 samples per class, and there is no limitation on the number of variables.As the number of variables may be more than the number of samples in chemical data sets. Many of the standard discrimination approaches would violate in these situations due to problems arising from the collinearity and chance classification. A main drawback of this algorithm is due to its sensitivity to data scaling and performance.

There are some applications of SIMCA scheme which has been used e.g. for modelling of in vitro hepatotoxicity by QSAR [60]. Also, in 3D-QSAR studies, SIMCA was used as a categorization tool [61, 62].

Among different classification approaches, e.g., LDA, QDA, and kNN, etc., SIMCA can be regarded as unique, in that it gives models of the classes. The resulted score plots present better indication of the data homogeneity in each class. For instance, they can be used to recognizing strong clusters in any of the score plots, resulting to such class that should be further split into subclasses.

Regression methods

Multiple linear regression

In a simple regression analysis, the relationship, called the regression function, is studied between a dependent variable, y and a single independent variable x, given a set of data that includes observations for both of these variables for a particular population. A regression function linear in the parameters (but not necessarily in the independent variables) will be referred to as a linear regression model. Otherwise, the model is called non-linear. Linear regression models with more than one independent variable are referred to as multiple linear models (MLRs). By fitting a straight line through the points, one can perform a simple linear regression analysis. The model is written in the following form;

$$y_i = b_0 + b_1 x_i + e_i \quad (8)$$

Regression function also includes a set of unknown parameters b_i, where $b_0$is the intercept and $b_1$is the slope of the line. Using the least squares criterion to estimate the equations, one can minimize the sum of squares of the differences between the actual and predicted values for each observation in the sample. That is, $\sum e_i^2$ will be minimized. In MLR, the relationship between the dependent variable and the explanatory variables is demonstrated by the following equation:

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_p x_{pi} + e_i (9)$$

where$b_0$is the constant term and $b_1$to $b_p$are the coefficients of the p predictor variables. The regression equation can be represented in matrix notation as follows:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e} (10)$$

where $\mathbf{y}$ is an n × 1 vector of response values, $\mathbf{X}$ is an n × (p + 1) matrix of independent variables, and bis a (p + 1) × 1 vector of regression coefficients [63].When $\mathbf{X}$ is of full rank the regression

coefficients can be resolved by the least-squares solution represented as:

$$\mathbf{b} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \quad (11)$$

Once **b**is determined, by using equation 11one can be able to estimate the dependent variable for other compounds.

MLR is a popular solution to regression problems in QSAR. There are at least three weaknesses in using MLR in the field of QSAR. Predictor variables, $\mathbf{x}$, is better to be mathematically independent (orthogonal), that is rank of $\mathbf{X}$ is equal to the number of $\mathbf{X}$-variables. Then MLR is sensitive to correlated variables. If MLR is performed to data sets containing significant correlations (collinearities) among $\mathbf{X}$-variables, the estimated regression coefficients get unstable. For instance, they may be much larger than expected, or may show wrong sign. So, there should be a minimal intercorrelation among the variables. A solution for this problem can be using long and lean data matrices so that the number of objects substantially is more than descriptors to decrease multicollinearity among variables. A minimum of 5 cases per predictor is usually recommended [14]. The usual approach taken with MLR and correlated X variables is to choose a subset of variables that are not so well correlated. The quality of a MLR can also be judged by looking at the standard error of the regression coefficients [28]. One should be aware of the pitfalls of using regression coefficients to argue the relative contribution of a descriptor to the measured activities which is just possible, after normalizing the equation [64]. MLR is on the basis of the assumption that each variable is important in the model, in other words, the model dimensionality is known a priori. Then the mission is to find out the “best subset of variables”, giving the optimal model in MLR. But testing all possible variable combinations is not practical when the number of variables is too large. A possible remedy for this problem can be selecting several orthogonal variables either using some preliminary knowledge or using a variable selection system. Several approaches for variable subset selection have been proposed in QSAR. Among them,the most widely applied are evolutionary and genetic algorithms, stepwise regressions, forward selection, backward elimination, and simulated annealing [65]. Consequently, three alternative procedures are commonly used, namely, forward selection, backward elimination and stepwise regression. These algorithms prevent all subset searches by following certain rules in conducting the search so are called “directed search” algorithms.

Genetic and evolutionary methods

Evolutionary algorithms have been used to relating structural information to activity or property information in both a quantitative [66, 67] and a qualitative manner especially in the QSAR model developments primarily as descriptor selection approaches. In these algorithms, a random process is usually used to generate an initial population of individual solutions. To investigate the fitness of each individual, a 'fitness function' is defined. It takes a candidate solution as input and yields a numeric score. Then, under application of selection criteria individuals are selected based on their fitness score for breeding. Lastly, breeding functions are used to return new solutions, which are replacements for the parent ones. The evaluation of the new solutions continues in a cycle that either a fixed number of cycles or a particular criterion is met [68]. There are growing reports of successful applications of evolutionary methods to docking, conformational analysis, drug dosing strategies, similarity searching, pharmacophore identification, feature selection, QSAR model buildings, combinatorial library design, and de novo design that anticipates a bright future for evolutionary methods in drug discovery[69, 70].

Among evolutionary algorithms, genetic algorithm is a stochastic optimization method mimicking the selection phenomenon in nature. It means, species with higher fitness can prevail in the next generation. Two tools of crossover together with random mutations of chromosomes are determining in the selection of surviving species. GA is governed by biological evolution rules.

As feature selection tools, GAs first applied in 1992[66]. Rogers and Hopfinger used GAs in a similar manner to choose functions of one or more features in a method they named the genetic function approximation (GFA) [71].Tests on the Selwood [72] and other datasets showed that GFA could produce multiple, high-quality QSAR models quickly. Some recent applications of GFA can be found here [73, 74].

Ridge regression

One of the various regression procedures which has been described for the highly correlated variables is ridge regression (RR). The regression coefficients in the RR procedure are obtained from equation 6;

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \mathrm{k}\mathbf{I})\mathrm{b} = \mathbf{X}^{\mathrm{T}}\mathbf{y} \quad (12)$$

Where $\mathbf{X}$ is the n×p matrix of the standardized $\mathbf{x}$ variables, k is a positive number (usually 0 < k < 1) and $\mathbf{I}$ is the p×p identity matrix. If compared with least square equation, it is revealed that a constant value ($\mathrm{k}\mathbf{I}$) is added to the diagonal elements of the $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ matrix of the normal equations. Note that biased estimates of the regression coefficients are obtained in RR as a constant k is added to the elements. In order to increase the stability of regression coefficients, it is needed to introduce some bias in RR. As k values increase, it results in an increase in bias estimates, but the variance decreases to a large degree. Also, as the residual sum of squares, SSRCS, with increasing of k value increases, R^2 decreases. Hoerl and Kennard [75] suggested an examination of a ridge trace which is a plot of the regression coefficients for different values of the bias parameter to determine k value. As the value of k is

determined, the regression coefficient should be stabilized and have the proper sign. Bear in mind that the reduction in R^2 should not be too large. This slight reduction can be evaluated from a plot of R^2 against different k values.The ridge regression is a means of regularising the regression (which is an "Ill-posed" problems (Tikhonov) into a "well posed" problem that is more stable. It allows a balance to be found between model complexity (that can lead to overfitting) and bias (where to model is too simple to explain the relationships in the data).

Ghasemi et al. [76] presented a QSPR studyfor estimating the incorporation of organic hazardous of 40 solutes [77] in cationic surfactant (CTAB) by application of the structural descriptors and MLR method. A total of 54 molecular descriptors were calculated to describe compound structural diversities. A genetic algorithm procedure was used for the variable selection and 27 molecular descriptors were selected. Then,a MLR analysis was performed by using stepwise method for model building between the selected molecular descriptors and micellar solubility (**Ks**). After regression analysis, the best equation wasselected on the basis of the highest multiple correlation coefficient (R^2) and simplicity of the model. As a result, a total of five molecular descriptors were obtained which is presented as the correlation matrix in Table 1. It is clear that the appeared descriptors in the MLR model are not highly correlated. To further evaluate the probable collinearity between selected descriptors, ridge traces using different lambda values were sketched using ridge function of the Statistics Toolbox 3.0 of MATLAB. Toward an expected conclusion, the ridge traces using different values of the lambda was constructed. The optimum value of the lambda was 0.01967 and the RR coefficients for DPLL, HomoE, logP, MP and RepE were −0.1009, 0.0020, 0.5695, 0.0640 and 0.0006, respectively. These results showed that the descriptors HomoE and RepE are collinear and have no considerable statistical impact on the final equation. The result obtained from the multivariate combinations is shown in eq. (13).

$$\log \mathbf{Ks} = -1.1522\ (\pm 0.2901) + 0.0070\ (\pm 0.0015)\mathrm{MP} + 0.8089\ (\pm 0.0897)\log \mathrm{P} - 0.1262\ (\pm 0.0454)\ \mathrm{DPLL} \qquad (13)$$

Table 1. Correlation matrix for selected variables by GA

	logKs	DPLLa	HomoEb	logPc	MPd	RepEe
log**Ks**	1					
DPLL	-0.19805	1				
HomoE	-0.10216	0.222757	1			
logP	0.860989	-0.23142	-0.02406	1		
MP	0.577207	0.477531	0.007778	0.355147	1	
RepE	0.76541	0.145376	-0.29033	0.650662	0.823401	1

a Dipole length
b Homo energy
c Octanol/water partition coefficient
d Melting point
e Repulsion energy

Tikhonov regularization

In every multivariate calibration model, as mentioned previously we commonly use the following equation:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e} \qquad (14)$$

Here the common approach for solving eq. 14for an estimation of b can be the least squares (LS) solution such as MLR, PLS, or PCR. This is normally delineated as a minimization problem in the 2-norm of [30, 78]

$$min(\|\mathbf{Xb} - \mathbf{y}\|_2^2) \qquad (15)$$

Generally, in cases where biased methods are used, they result in an overfitted model. However, in order to circumvent this problem Tikhonov regularization can be used with;

$$min(\|\mathbf{Xb} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{Lb}\|_2^2) \qquad (16)$$

where**L** is the representative matrix of values usually approximating a derivative operator such as the first or second order to generate a smoothed regression vector and k shows a scalar that must be optimized.Ithelps to control the shape of the regression vector with **L** [79, 80]

When minimizing the expression (16) with **L** = **I**, it is called Tikhonov regularization in standard form and is RR with ridge parameter λ. The point is that while eq. (16) represents RR when **L** = **I**, the standard approach in the chemistry literature for determining λ is not to use expression (10), but instead to use only a bias measure such as RMSEC, RMSECV, or RMSEV or using ridge trace plot [75].Ittries to show in two dimensions the effects of nonorthogonality and aims at portraying the information explicitly.Hence, it guides the user to a better estimate$\hat{\mathbf{b}}^*$. When **L** is not the identity matrix, then eq. (10) is said to represent Tikhonov regularization in general form. With QSAR data, it is expected that **L** ≠ **I** would not be beneficial because adjacent molecular descriptors (columns) in **X** are not related in a smoothed way as with spectroscopic data. Since, the eq.16minimizes the bias and variance, a harmonizedmodel is achieved, that is, the chance of obtaining an over or underfitted model is considerably reduced. Therefore, the appropriate model identified with eq. (10) forms an L- shaped curve by using a harmonious plot. The most desirable advantage obtained by RR models is that they are more harmonious and parsimonious than models obtained by PLS and principal component regression (PCR) when the data is mean-centered. RR is said to have the best bias/variance tradeoff, shown by the smallest RMSEC, RMSEV, and regression vector norms and the largest calibration and validation R^2 values [81].

J. Forrester and J.H. Kalivasanalyzed using leverages as a variance indicator. In all cases, the characteristic L-shaped curves were achieved and the same PCR, PLS, and RR models were identified as highly harmonious. For instance, points composing a PLS harmonious curve would correspond to PLS models formed with increasing number of PLS factors. Underfitted models which produce the curve parallel to the x-axis (bias axis) have low variance and high bias values. On the other hand, overfitted models have aharmonious curve positioned parallel to the y-axis (variance axis) resulting in high variance and low bias models. Ideally, the best model occurs at the corner of the plotted L-shaped curve and shows the more harmonious model providing a good balance between the minimization of variance and bias. It is ultimately up to the user to select the final desirable model.

John H. Kalivas et al. [81] carried out a QSAR study on a data set consisting of 142 inhibitors of carbonic anhydrase IV (CA IV) [82]. They used 63 molecular descriptors for modelling. The multivariate methods used were PLS, PCR and RR and the purpose was to compare modelling methods and select the best model for a given data set of compounds and descriptors. As shown in Figure 8a, all 63 molecular descriptors have been used for the prediction of CA IV by the harmonious RMSEC plot. The curve has increased in variance with the slight change in the RMSEC bias measuredat 6 or 7 factor PLS models. This occurs for PCR in 7 to 10 factor ranges and for RR between ridge parameter values 800 and 400. As demonstrated in Figure 8b the corresponding RMSEV values are given and the optimal models for PLS and PCR, namely, the six factor PLS and eight factor PCR models are clearly discerned. Figure 8b reveals that while the number of factors increases by one resulting in a little improvement for RMSEV values, the variance indicator increases. The plot in Figure 8b shows that a narrow range for the best ridge parameter value is achievable. Itwas mentioned before thatbecause the ridge parameter is continuous to infinite decimal places, a range is expected. From Figure 8, it is concluded that the RR model with a ridge value of 750 provides a good balance of bias and variance. Figure 8 and corresponding criteria values listed in Table 2 show that RR does the best and should be taken into account as the more harmonious model. As presented in the Tikhonov regularization, specification of the mostactual harmonious model needs to a more thorough variance expression for the predicted values.

Table 2. Model values for CA IV.

Modela	No.of descriptors	$\|\hat{b}\|_2$	RMSEC	RMSEV	R2cal	R2val	ER
RR(750)	63	0.0721	0.575	0.774	0.870	0.751	0.891
PLS(6)	63	0.0855	0.585	0.791	0.865	0.740	1.011
PCR(8)	63	0.0948	0.585	0.795	0.853	0.747	1.170
RR(6)	8	0.460	0.536	0.671	0.880	0.816	1.036
PLS(4)	8	0.463	0.546	0.676	0.875	0.813	1.034
PCR(5)	8	0.464	0.551	0.677	0.879	0.814	1.045
MLR	8	3.553	0.504	0.700	0.885	0.811	8

aParentheses contain ridge values for RR and the number of factors for PLS and PCR.

Principal Component Regression (PCR)

In multivariate analysis, the ordinary least squares regression coefficient estimators are generally adopted in fitting a MLR model, but estimation of the least squares is sometimes far from being perfect. It is well known that the quality of prediction by multiple regression is negatively affected by correlation between the **x**variables (multi collinearities). Therefore, the resulted estimate of the vector of regression coefficients, bmay have low probability of being close to the true value. PCR is a PCA based regression method that is quite popular in chemometrics. A heavy use of PCR and related procedures for predicting a dependent variable from a large number of highly correlated predictors has been made in QSAR. In PCR, regression of response variable is carried out on the principal components(PCs) instead of initial variables. Moreover, only a subset of all principal components is used to model building and the important features of original variables are retained. These PCs are chosen based on the order of variance within the data: eigenvectors related to the higher eigenvalues of the samplevariance-covariance matrix of the original variables are selected as predictor. In general, a PCA is performed on **X** and only just a subset of all PCs is retained as significant. Then, a multiple regression analysis of the response variable against the reduced set of principal components is performed using ordinary least squares estimation. Thus, the main advantage of PCR compared with multiple regression is that the number of variables is reduced to only a few uncorrelated ones (feature reduction).

A problem that one may encounter in PCR is discarding of minor **X**-components that probably have high correlation to **y**. One solution may be adding up PCs in the model until **y**is fitted well, if the resulted model also satisfies the (cross-) validation procedure. Another strategy is to enter PCs in a reverse order of common approach (PC1, PC2 ...). Another way is to calculate the correlation coefficient of each PC with **y**, and enter the PCs in descending order of correlation coefficient magnitude.

There are reports of application of PCR in different fields of QSAR [83, 84]. Xiao Li et al. correlated the toxicity of PAHs with physical and chemical properties QSAR descriptors by PCR Method [85].

Compared to multiple regression analysis, the advantage of PCR is that existing collinearities between variables are not a disturbing factor, and that the number of variables present in the analysis can be more than the number of observations. On the other hand, in PCR and similar methods, including too many variables in the PCA step may cause chemical interpretation increasingly difficult [28].

Partial Least Squares (PLS)

PLS is a generalization of MLR, and has been developed by Wold et al. [86-90]. It is occupying a middle ground between MLR and PCR, and all of them are special cases of continuum regression. Factors found by PCR capture the greatest amount of variance in the predictor (**x**) variables while MLR searches for a single factor that best correlates predictor (**x**) variables with predicted (**Y**) variables. In PLS regression, assumption is that both the independent matrix **X** and the dependent matrix **Y** can be projected onto a low-dimensional factor space and there is a linear relation between the scores of the two blocks. PLS tries to find factors which both include variance and achieve correlation. 3D-QSAR methods such as CoMFA, generate a large number of X variables often exceeding 10000, while the number of compounds is remained moderate, such as, between 10 and 100. So PLS is an appropriate tool for data analysis in this case. There are several algorithms for calculating the PLS regression model for different shapes of the data, such as Non-Iterative Partial Least Squares (NIPALS) as one of the most intuitive ways of computing PLS model parameters and SIMPLS [91]. In PLS regression, the **X** block of independent variables is correlated with the **y** vector (dependent variable) in such a way that the projected coordinates, **t**, are good predictors of **y**. In other words, the dependent variables are included in the decomposition procedure. When there are multiple **y**-variables, scores **U** and loadings **C** matrices are also computed for the **Y**-block. Also a vector named "inner-relationship" coefficients, **b**, which establishes relation the **X**- and **Y**-block scores, must also be computed. There are a number of methods to PLS development [30]. In NIPALS, scores **T** and loadings **P** (similar to those used in PCR) are calculated. It computes an additional set of vectors identified as weights, **w**. This addition of weights in PLS is necessary to keep scores orthogonal. The aim of SIMPLS algorithm for PLS is to maximize the covariance. When there is more than one predicted variable (**y**), both of the two algorithms of NIPALS and SIMPLS also work. The original NIPALS algorithm below, works with the original data matrices, **X** and **Y** (scaled and centered). Alternatively, so-called kernel algorithms work with the variance- covariance matrices, $\mathbf{X}^T\mathbf{X}$, $\mathbf{Y}^T\mathbf{Y}$, and $\mathbf{X}^T\mathbf{Y}$, or association matrices, $\mathbf{X}\mathbf{X}^T$ and $\mathbf{Y}\mathbf{Y}^T$, which is beneficial when the number of observations differ much from the number of variables. The linear PLS model finds "new" variables, called latent variables, which are linear combinations of the original variables also called **X** scores and which are also denoted by ta (a = 1, 2… A).So, one supposes that both **X** and **Y** is modelled by the same LV's, at least partly. The **X**-scores are orthogonal, with the numbers equal to A.

The ability of the regression model for future predictions is validated usually through an internal leave-one-out cross-validation procedure. To choose the optimal number of PLS factors, PRESS (predicted residual error sum of squares) value is monitored as the function of latent variables. The number of components related to the minimum PRESS is chosen.

$$PRESS = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (17)$$

where n is the number of compounds used in the cross validation procedure. The standard error of prediction (SEP; equation 20) can also be applied to measure the prediction ability of the model:

$$SEP = [\frac{PRESS}{n}]^{1/2} \quad (18)$$

The performance of PLS has been enhanced by combination with other methods to give better results in QSAR/QSPR analyses. Some of these approaches are genetic partial least squares (G/PLS) [92, 93], factor analysis partial least squares (FA-PLS) and orthogonal signal correction partial least squares (OSC-PLS) [94, 95].

PLS is more realistic than MLR to construct models of the often complicated relationships between chemical structure and biological activity. PLS provides diagnostics such as cross-validation and score plots with the corresponding loading plots, which inform about model complexity and the structure of X data that is not attainable with ordinary MLR. Moreover, a shortcoming of MLR is lacking the diagnostic tools for pointing out inhomogcncities in the data. In addition, PLS is able to address many collinear structure descriptor variables which make it easier to clarify the variation of compound structures [28].

Y-Randomization is commonly used as a tool in validation of QSPR/QSAR models. The goodness of the initial model in data description (R^2) is compared to that of models built using randomly shuffled response, based on the original feature pool and the original model building technique. Rucker and et al compared the original y-randomization approach and several variants thereof, by means of original response, permuted response, or random number pseudoresponse and original descriptors or random number pseudodescriptors, based on an original MLR model with descriptor selection [96]. Their investigations have been applied to several published MLR QSAR equations, and cases of failure have been recognized. Some progress also is shown toward the goal of getting the mean highest $R^2$of random pseudomodels by computation instead of tedious multiple simulations on random number variables. In the case of PLS models, two validation methods are widely used namely cross-validation and response randomization. However, for data sets that contain redundant observations bothmethods may be overly optimistic. Progressive scrambling [97] is a nonparametric method to perturbing models in the response space with not disturbing the underlying covariance structure of the data. After reordering the observations from largest to smallest response, they are then blocked following several rules. The responses within each block are then shuffled, and the modified pairings are submitted to PLS analysis to get relevant statistics. These statistics were shown to be robust for stable PLS models, in terms of the stochasticcomponent of their determination and of

their variation because of sampling effects available in training set selection.

Other extensions of PLSR

Several extensions of PLSR can be found in various directions, too such as non-linear modelling, hierarchical modelling when the variables are very numerous, multi-way data, PLS time series, and with many and collinear **X**-variables, a PLS version of LDA (PLS-DA) is helpful [98]. When dealing with biology, one should be aware of non-linear behaviour of the systems. Therefore the modelling of biology is fundamentally a non-linear event [45]. So, we cannot expect to model all biology with linear relationships, if so they will not to be predictable. There are, for example, QSARs for acute aquatic toxicity. To cure this situation, it might bepossible to model nonlinear data relations by means of PLS. The first group of approaches is to use a nonlinear model between the score vectors **t** and **u** based on reformulating the considered linear relation [99].

$$u = g(t) + e = g(X, w) + e \quad (19)$$

where g(.) indicates a continuous function modelling the existing nonlinear relation. e denotes a residual vector. Polynomial functions, smoothing splines, artificial neural networks or radial basis function networks have been applied to model g(.) [100, 101]. The assumption that the score vectors **t** and **u** are linear projections of the original variables is reserved. This results to the requirement of a linearization of the nonlinear mapping g(.) through Taylor series expansions and to the successive iterative update of the weight vectors **w** [102]. The second approach of considering nonlinear PLS is on the basis of a mapping of the original data through a nonlinear function to a new data space where linear PLS is performed. The theory of kernel-based learning has been also applied to PLS. The application of nonlinear kernel PLS (KPLS) methodology was to model relations between sets of observed variables, regression and classification problems [103, 104]. In kernel PLS, the original $\boldsymbol{\chi}$-space data is mapped into a high-dimensional feature space $\boldsymbol{\mathcal{F}}$ corresponding to a reproducing kernel Hilbert space [105, 106].

$$X \epsilon \boldsymbol{\chi} \rightarrow \Phi(X) \epsilon \boldsymbol{\mathcal{F}} \quad (20)$$

when the kernel trick is applied, the estimation of PLS in a feature space $\boldsymbol{\mathcal{F}}$ would be reduced to the use of linear algebra as simple as in linear PLS . KPLS method has also been shown to be competitive with the other state-of-the-art kernel-based approaches in regression and classification problems. The detailed description of kernel-based learning can be found elsewhere [103]. There are some examples in applications of KPLS method in QSAR/QSPR [107-109].

In a generalization of the PLS approach to multi-way data, a multilinear PLS algorithm (N-PLS) which is an extension of the traditional bilinear PLS has been developed. In the three-way mode of PLS, a trilinear model is obtained from decomposition of the three-way array of independent variables similar to the PARAFAC model. However, for N-PLS, least squares do not apply to fit the model, but in line with the philosophy of PLS, it searches to describe the covariance of the dependent and the independent variables. This is achieved by simultaneously fitting a multilinear model of the dependent variables, a (bi-) linear model of the independent variables, and a regression model relating the two decomposition models. The advantage is that the multiway (or higher order) structure of the data is retained. An application of this method in drug design was in the case of 3D-QSAR problems [110]. In a study on a series of azidothymidine (AZT) analogues, two methods of bilinear (traditional) PLS, applied to the unfolded dataset, and multilinear PLS (N-PLS), applied to the three-way array has been compared. The predictive abilities of the PLS- and N-PLS-based models were found to be nearly equivalent [111].

To represent one of the applications of PLS in the field of drug discovery, a work from Bacilieri et al. has been chosen [112]. In this paper, the SAR of hA_{2A} AR antagonists through an integrated structure- and ligand-based strategy has been investigated. A database of 751 hA_{2A} AR antagonists which can be grouped into 22 different scaffolds together with their pK_i values (all showing K_i< 1 μM) has been used. In each analysis, the compounds have been split into a training and test database randomly: 80% has been selected as the training set and 20% as the test set. Furthermore, a new small antagonist library has also been synthesized and pharmacologically characterized which comprises 29 pyrazolo-triazolo-pyrimidines (PTP) compounds. The complex with the high affinity antagonist ZM241385 (PDB code: 3EML) has provided from PDB site for docking simulations. Docking of all of the compounds into the TM binding site of the hA_{2A} AR crystal structure was carried out by using the docking tool of the GOLD suite. For each ligand, the best obtained docking pose as evaluated by the GoldScore scoring function was selected to form a first data set (named as best pose database). Moreover, the pose that best fits the crystallographic binding mode of ZM241385 was selected visually to form a second data set (referred to as selected pose database). Both two data sets were then regarded in the following QSAR studies. Ligands conformations needed to build pharmacophore models were generated by the docking simulations, and pharmacophore sites for each molecule were created with PHASE [113]. Consequently, to derive a common pharmacophore hypothesis consistent with all scaffolds, the crystal structure of ZM241385 and its docked conformation as an active subset was used, because at least two compounds are required to search for a common hypothesis. After definition of a consistent pharmacophore hypothesis, both pharmacophore-based and atom-based 3D-QSAR models, available in PHASE were built. In the atom-based model, representation of the ligands is done by a set of overlapping van der Waals spheres (one for each

atom in the ligand) classified as D, H, N, P, and W (where W refers to electron-withdrawing), while in the pharmacophore-based model representation of the ligands is by spheres of specified radius, to which a category reflecting their pharmacophore features is assigned. PLS analysis as the modelling tool is then used to correlate the above mentioned descriptors with the activities. The pharmacophore-based 3D-QSAR analysis has been derived considering a final training set of 599 molecules and a test set of 149 molecules. PLS statistical results for pharmacophore-based 3D-QSAR analysis are reported in Table 3. The authors also generated atom-based 3D-QSAR models for both the best pose and the selected pose data sets. The training set consisted of 601 molecules and the test set, 150 molecules, and the correlation coefficients of both models are about 0.8. The ability of the models to predict the K_i values of the 29 PTP compounds (external validation) is checked, and the correlation coefficients obtained are 0.6/0.5 with six and five PCs for the best pose and the selected pose database, respectively. The results show that the statistical quality of atom-based 3D-QSAR models enables to quantitative discrimination of active from inactive compounds, while pharmacophore-based 3D-QSAR models have ability to a qualitative prediction of the activity. As a consequence, in this specific case under study, the best approach to predict hA_{2A} AR antagonists activity is to apply molecular docking and to use the best pose conformations for the atom-based 3D-QSAR analysis. Bearing in mind that the applicability domain of pharmacophore based screening methods is more focused on the new hits identification rather than hit to lead optimization; the best pose-atom based 3D-QSAR model can be used as a query to screen large chemical databases.

Table 3. Pharmacophore based 3D-QSAR model results[a].

Pharmacophore based 3D-QSAR model	Best pose database	Selected pose database
Training set	599	599
Test set	149	149
PCs	6	7
r^2	0.49	0.55
q^2	0.42	0.44
RMSE	0.78	0.75
F	92.3	102

[a]PCs, principal components; r^2, squared correlation coefficient of calibration (training set); q^2, squared correlation coefficient of validation (test set); RMSE, root mean squared error; F, variance ratio (largest values correspond to better statistical significance).

Projection Pursuit Regression

The projection pursuit regression is a nonparametric statistical technique which applied to build anonlinear model. PPR seeks the ''interesting'' projections of data from a higher-dimensional to lower-dimensional space to attempt to find the intrinsic structure information hidden in the high dimensional data [114]. It can effectively overcome the curse of dimensionality. With the obtained interesting projection direction, it can be used for further study of visual pattern recognition and regression. For this reason, the PPR technique has fascinated more attention and gained extensive application in high dimensional data like the QSAR studies. Using of lower dimensional linear projection of the data, it is possible to visualize the data practically. Furthermore, PPR does not require specification of a metric in the predictor space. It does not split the sample, thereby allowing, when necessary, more complex models. In addition, interactions of predictors are directly considered. The PPR technique is based on an iterative two-stage process of projection and smoothing. Projection causesthe reduction of parameter space and smoothing causes establishing a non-linear relation. For application of smoothing, the reduction of the parameter space is essential; smoothing in high-dimensional spaces quickly becomes impossible because of data sparsity. More precisely, $\mathbf{X}1, \ldots, \mathbf{X}_n$, $\mathbf{X} \in Rp$ are p-dimensional data, then a k-dimensional (k < p) linear projection is $\mathbf{Z}1, \ldots, \mathbf{Z}n$, $\mathbf{Z} \in R_k$ where Zm=αTXm for some p × k matrix α such that αTα=Ik, the k-dimensional identity matrix. Such a matrix α is often orthonormal. There are a lot of infinite projections from a higher dimension to a lower dimension. So, to achieve the most interesting data structures, having an efficient technique to pursue a finite sequence of projections is required. Friedman and Stuetzle [115] successfully applied the projection pursuit (PP) which was combining projection and pursuit.

In a typical regression, several parameters should be given first: X, matrix of explanatory variables (n×p); n, the number of objects under investigation (Chemical structures); p, number of explanatory variables (Molecular descriptors); y, vector of response (n×1) (activity or property or). The projection process can be defined as zm=X αm. Where αm is the mth projection vector (length p); and zm is the vector of scores after projection of X on αm (length n). After the projection, the smooth functions (ridge functions) are used, which are as follows:

$$y = \bar{y} + \sum_{m=1}^{m_0} \emptyset_m (z_m) + \varepsilon M_0 \qquad (21)$$

Where M_0 is the number of incorporated smooths; εM0 is the residual error after fitting M0smooths. Then it can produce a non-linear regression model by the summation of a number of ridge functions.

PPR as an upgraded method applied in drug design. As it has been mentioned before, PPR was used to visual pattern recognition and regression. Du et al. [116] developed QSAR models for the data set of 39 ligands of derivatives of naphthalene, benzofuraneand indole with respect to their affinities to MT3/quinone reductase 2 (QR2) melatonin binding

site. They compared QSAR models based on linear regression and non-linear regression methods grid search-support vector machine (GS-SVM) and PPR. The physiological significance of the MT3/QR2 site is still unidentified and it is mostly interesting to design and synthesize new selective ligands, which will supply pharmacological tools to assess and better describe the role of this melatonin binding site. Therefore, finding an efficient way to get the affinity of the new compounds for melatoninergic binding sites MT3 in early ligand discovery is one of the major challenges facing the field. The optimized structures by AM1 method in MOPAC, were transferred into the CODESSA software to calculate the descriptors. Descriptors of CODESSA includeconstitutional, topological, geometrical, electrostatic, and quantum chemical; which can represent a variety of aspects in the compounds. Both the linear and non-linear models constructed. Compression of the results of three regression methods showed that PPR was proved to be a very useful tool in the prediction of the MT3/QR2 melatonin binding site, and it showed that to be a very promising machine learning method and would yield more extensive applications. Since the PPR analysis considers both linearity and nonlinearity in the model development, consequently it yielded more predictive models than linear regression. Also, the PPR method performs a flexible regression in a low-dimensional variable space, contrarily to the SVM regression which uses a fixed transfer function.In addition, it should provide facilities to the design and development of new selective MT3/QR2 ligands.

Furthermore, Yuan et al. [117] developed a PPR approachto predict binding affinity of a series of CCR5 receptor inhibitors. Results showed that PPR method is simple, practical and effective to predict the affinity of human CCR5 chemokine receptor inhibitors.

k-nearestneighbour method (kNN)

In supervised pattern recognition, (or discriminate analysis, or supervised learning) using the learning or training objects with known origin, one tries to derive a classification rule which allows classifying new objects with unknown origin in one of the known classes. This derivation is based on the values of the variables of the new object. In the first step of doing a supervised learning, training or learning set is selected. These are objects of known classification for which a certain number of features have been measured. In the second step, variables that are meaningful for the classification are chosen and those that have no discriminating (or, for certain approaches, no modelling power) are removed. Then, a classification rule using training set is derived. Finally, using an independent test set, validation of the classification rule is carried out. All the clustering methods can be applied in variable selection by using the transposed matrix of the original data, where descriptors can be located in rows and molecules in columns. Then, one or more representative descriptors are selected from each cluster. Nearest neighbours have very simple machine learning algorithm. In these methods, classifier is the distance between each object of the training set, and the function is only approximated locally based on neighbours [118]. Euclidean distance is commonly used but other distance measures can also be used. However, for strongly correlated variables, correlation based measures are favoured. For the training set of n objects, n distances related to a test sample are calculated and the lowest of them is chosen for the assignment of class membership. The k-nearest neighbour method (kNN) is a non-parametric unbiased approach with applications in classification and regression [15, 119-121]. In nonparametric modeling methods in general, derivation of the relationship between the descriptors and the activities is carried out directly from the data, instead of applying a functional form on the data a priori. A number of nonparametric methods applied to QSAR comprise gaussianprocess regression [122, 123], Nadaraya–Watson regression [124, 125], kernel discrimination classifiers [126, 127], and Kriging [128]. Typically, in these approaches a kernel is used to weight the effect of the training set molecules in such a way that similar compounds have the highest influence in the prediction.

In kNN classification, class membership is assigned while in the regression the average property value for the sample is calculated from the activity values of its kNNs in the training set, Figure 9. The k value giving the lowest classification error in cross-validation is chosen as the optimal one. In this procedure, each sample in the training set is removed and then classified using the remaining training set compounds. Formally, the upper level of k is the total number of compounds in the training dataset; though, the best values is determined by the class of its single nearest neighbour (k = 1) or by a vote of a small (generally odd) number of its nearest neighbours (k = 3, 5, …). This is repeated for different values of k. Each of the k nearest samples “votes” once for its class. The class with the highest number of votes is assigned to that sample. The kNN-QSAR methodology is implemented simply. First, distances between an unknown object (u), and all the objects in the training set is calculated. Then, the most similar (the least distances) k objects from the training set to object u are selected and the activity value of u is calculated as a weighted average of the activities of its kNNs. While a single observation is excluded from the training set, its activity value is predicted as a weighted average of the activity values of its nearest neighbours:

$$\hat{y}_i = \frac{\sum_{kNNs} y_i \exp(d_i)}{\sum_{kNNs} \exp(d_i)} \quad (22)$$

where d_i is the Euclidean distance of the compound from its kNNs and $\exp(d_i)$ is regarded as weight. The predicted power of the model can be conveniently estimated by an external Q^2 calculated as follows:

$$Q^2_{LOO} = 1 - \frac{\sum_{i=1}^{training}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{training}(y_i - \bar{y}_{tr})^2} \quad (23)$$

wherey_i are the observed and $\hat{y}_i$are the predicted values of the compounds in the training set and $\bar{y}_{tr}$is the mean of the experimental activity of the compounds in the training set. The advantages of this method are as follows: (i) beside mathematical simplicity, it achieves classification results as fine as (or even better than) other more complex pattern recognition approaches (ii) as a non-parametric approach, it does not need statistical assumptions, such as the normal distribution of the variables and (iii) its efficiency does not depend on the space distribution of the classes [129]. On the other hand, this technique is subject to some problems. kNN cannot work properly, if there are large differences in the number of samples in each class. A solution is that instead of a simple majority criterion an alternative criterion used then. For instance, another choice of criterion in kNN consists of weighting the importance as a neighbour of a known object to an unknown sample (inverse distance or inverse square distance). So, the nearest neighbours influence more the classification than the farther ones. Furthermore, in the case of large number of samples, the computation cost can become excessively much. Moreover, the information provided about the structure of the classes and the relative importance of each variable in the classification is not sufficient. It means that there is no information in the kNN model concerning what variables are useful in separating classes from each other while a for example a simple PCA model contains much of this information in the loadings. In addition, kNNs are not efficient in dealing with high dimensional data without dimension reduction or preselection of descriptors [9].

There are some applications of kNN scheme which has been used e.g. for prediction ofCOX-2 inhibition [130], modelling of anticonvulsant activity, dopamine D1 antagonists and aquatic toxicity [131]. In a study on P-glycoprotein transport activity [121], the result of kNN was comparable to decision tree, but it was worse than neural network and SVM. In an ecotoxicity QSAR study [59], k-NN showed better in comparison to some linear methods, but had a poorer operation to discriminate analysis and decision trees. In a recent study by Khashan et al. [132]kNN was applied as a variable selection approach to select among recently developed frequent subgraph (FSG) descriptors for QSAR models.

kNN and HCA in practice

As an example of how these two methods work, an article of work of Martin et al. about rational selection of training and test sets has been selected [17]. In this study, the authors examined the important problem of the QSAR model validation that is among rational or random division of a data set into training and test sets that which method is better. To approach this aim, four data sets were selected: a P. promelas LC_{50} data set consisting of 809 chemicals, a T. Pyriformis IGC_{50}data set consisting of 1085 chemicals, an oral rat LD_{50} data set of 7286 chemicals, and a bioaccumulation factor (BCF) data set of 643 chemicals. Each data set was randomly separated into a modelling and external evaluation set that incorporated 80% and 20% of the whole data set, respectively. The modelling sets were then split into training and test sets by rational and random orientations where training and test sets included 80% and 20% of the modelling sets, respectively. Calculations were performed separately by different researcher groups with abbreviate names drawn from US Environmental Protection Agency (USEPA), University of North Carolina (UNC), and UNC team in collaboration with A.V. Bogatsky Physical Chemical Institute NASU (UNC2). Each group carried out calculations for four different toxicity end points. The two research groups used different rational design approaches and QSAR methodologies with the purpose of fairly assess whether rational design methods accurately get better the external predictive ability of QSAR models. The USEPA group used the Kennard-Stone rational design method and the hierarchical clustering QSAR methodology. The UNC group used the sphere exclusion rational design method [15] and the random forest and kNN QSAR methods. Additionally, the minimal test set dissimilarity [133] and random forest were used by UNC2. Among them, rational division methods were Kennard-Stone algorithm, sphere exclusion algorithm, and minimal test set dissimilarity method. The QSAR approaches were hierarchical clustering, two implementations of random forest, and kNN QSAR. When using hierarchical clustering, the Kennard-Stone algorithm was applied to rationally divide a modelling set into training and test sets. For one of the random forest implementations, the sphere exclusion algorithm was selected to produce one couple of training and test sets. The minimal test set dissimilarity method was used for another random forest implementation. USEPA used 790 descriptors came from different classes, UNC calculated them using Dragon version 5.4 and also 2D Simplex descriptors generated by HiT QSAR Software [134] were used by the UNC team. As examples, the statistical results for the USEPAgroup (Kennard-Stone and hierarchical clustering methods) are shown in Tables 4 and 5.

Table 4. Splitting results in terms of the R^2 squared correlation coefficient (USEPA – Kennard-Stone and hierarchical clustering)

	R^2 test set		R^2 external evaluation set		R^2 external evaluation set (noapplicability domain, i.e. 100% coverage)	
end point	rational	random	rational	random	rational	random
LC_{50}	0.81	0.66	0.67	0.60	0.67	0.58

IGC_{50}	0.86	0.84	0.85	0.85	0.80	0.83
LD_{50}	0.76	0.53	0.55	0.53	0.47	0.47
BCF	0.89	0.80	0.73	0.81	0.71	0.68

Table 5. Splitting results on terms of the prediction coverage (USEPA – Kennard-Stone and hierarchical clustering)

	Coverage test set		Coverage external evaluation set	
end point	rational	random	rational	random
LC_{50}	100%	95%	99%	99%
IGC_{50}	100%	99%	98%	98%
LD_{50}	98%	82%	84%	83%
BCF	100%	96%	95%	94%

The three approaches to divide the modelling set into training and test sets did show no influence on model predictivity or coverage for the external evaluation sets for the QSAR methods that were used. The reason could be that both test and external evaluation sets were applied in a "blind prediction"way, as it is apparent from comparing their prediction performance. It was clear that the rational selection methods gained better prediction statistics for the test set but not for the external evaluation set. In the rational division methods and their corresponding QSAR methodologies, random selection provided a more accurate estimate of prediction ability as the statistics for the test set and external evaluation set are in essence equivalent. For kNN QSAR studies, division of a data set into training, test, and external evaluation sets is a common practice. In summary, it was found that roughly twice as many predictive models were obtained by the sphere exclusion algorithm than by using random division. The additional models may regard for the 5% (on average) boost in the prediction coverage for the external set. The R^2 values for the test and external sets showed comparable results, so the use of the sphere exclusion algorithm does not gain an unrealistic estimate of model performance. These results concern that in kNN QSAR studies sphere exclusion rather than random division should be used.

Artificial Neural Network

Artificial neural systems (NNs) have gained acceptance in various regions of chemistry, as delineated by the number of applications bring by Zupan and Gasteiger in their review [35]. The potential applications of ANN as modelling tools in multivariate calibration are broad. There are a lot of reports from ANNs methodology that successfully employed in QSAR.

The ANN is an appropriate technique that can be viewed as a general nonlinear modelling approach. Artificial neurons are simple computational devices that are highly interconnected and the connections between neurons determine the transfer function of the network [135]. Therefore, it is important for the user to have a good understanding of the science behind the underlying system to provide the appropriate input and, consequently, to support the identified relationship; however, ANNs are not interpretable. An artificial neural network determines an empirical relationship between the input variables called independent variables or descriptors (X) and output variables called dependent variables or responses(Y) without any prior knowledge in principle [136]. It can show the model of the form: Y = F(X) + e. Tasks of a neuron as a processing element are included to receive stimuli from other neurons through its dendrites and send stimuli to other neurons through its axon. Strength of connection between the neurons is stored as a weight-value which for the specific connection is called synapse. Information in an NN is distributed among multiple cells (nodes) and connections between the cells or synapse (weights). The activation signal is passed through a transform function to produce the output of the neuron, given by: y=f(a). The transform function can be linear, or non-linear, such as a threshold or sigmoid function. The process of determining the values for W on the basis of the data is called learning or training [137]. Learning used to change the connection weights solves a problem. In a biological system, learning involves adjustments to the synaptic connections between neurons the same for artificial neural networks (ANNs). Learning as a fundamental characteristic of ANNs has two basics types: supervised and unsupervised [138, 139]. In supervised method, there is a teacher but unsupervised is autonomous. Training of supervised networks is carried out by giving sets of input patterns and associated target patterns. During an iterative process, the internal representation of the data is tailored until the predicted results being closer as desired to the targets. In some applications, the target patterns are the same as input patterns and the network is in essence performing an identified mapping. Such networks are called self-supervised networks. It ought to be mentioned here that the supervised and unsupervised terms mean differently in the field of ANN from that is available in statistical pattern recognition approaches. In the former, supervised means any training which entails a priori targets, for instance, training a network is to map a data onto itself. In statistical pattern recognition terminology, supervised is used to mean training which involves a dependent variable which is used to derive a prediction rule e.g. a regression equation. On the contrary, when unsupervised learning is performed, a priori targets areabsent. Networks of this type may be applied to provide information on clusters of compounds which is just on the basis of the coordinates of the compounds located in the measurement space of its variables.

Many types of neural networks have been designed. The perceptron was first introduced by F. Rosenblatt in 1958 [140] with two layers including input and output layers. One of the most popular supervised networks is the multi-layer feed-forward type (also called multi-layer perceptron, MLP) with the error back-propagation learning rule that was introduced by M. Minsky and S. Papert in 1969 [141, 142]. Figure 10 shows an example of MLP with four descriptors and a single response y. The four descriptors are displayedx_1, x_2, x_3, x_4 at the input layer, the variables $w_{ij}^{"}$ and fh are the weight and transfer function between the input and hidden layer, $w_j^{"}$ and $f_o$are the weight between the hidden and output layer, b′ and b" are the bias vectors, respectively. In this kind of neural networks, neurons are ordered in input, hidden, and output layers. Input layer neurons get descriptor matrix, which have different weights and are passed to the hidden layer neurons (figure 10a). Aneuron activation function is then applied at each neuron to the sumof weighted inputs, and the results are obtained by output layer neurons, which compute activity values of compounds. During training process, adjusting of parameters of neuron functions and weights is carried out so that the overall error of predictions is minimized.There are network architectures with hidden layers (figure 10b). The output is the estimated response $\hat{y}$ that can be expressed as follow:

$$\hat{y} = f_o\left[b^{"} + \sum_{j=1}^{nh} w_{ij}^{"} f_h\left(\sum_{i=1}^{nd} w_{ij}^{'} x_i + b^{'}\right)\right] \quad (24)$$

where*nd* and *nh* are the number of input variables and hidden nodes.

Self-organizing map (SOM) with unsupervised learning algorithm ANN is exposed to large amounts of data and tends to discover patterns and relationships in that data. Teuvo Kohonen introduced one of this NNs namely Kohonen network [143, 144].

In the beginning, the Euclidian distance $d_E(y_l,y_k)$ and topological distance $d_T(y_l,y_k)$ between output nodesy_l and y_k will not be related (Figure 11).

$$d_E(y_l,y_k) = \sqrt{\sum_{i=1}^{n}(w_{li} - w_{ki})^2} \quad (25)$$

But during the course of training, they will become positively correlated: Neighbour nodes in the topology will have similar weight vectors, and topologically distant nodes will have very different weight vectors.

In SOM, the network is provided with inputs but not with desired outputs. Making decision about what features it will use to group the input data is done by system itself. This is often called self-organization or adaptation. Training and topology adjustments are made iteratively until a sufficiently accurate map is obtained [145]. The self-organizing treatment may include competition between neurons, co-operation or both. Neurons are arranged into groups of layers. In competitive learning, neurons are grouped in such a way that when one neuron replies more powerfully to a particular input it represses or inhibits the output of the other neurons in the group. In cooperative training, the neurons within each group work together to strengthen their output. The learning mission is to make groups between patterns that are similar in some way, extract features of the independent variables and come up with its own classifications for inputs. ANNs consider the received data, find out some of the properties of the data and learn to display these properties in their output. The aim is to construct feature variables from which the variables, both input and output ones, can be predicted.

As far as the architecture (layout) and the learning strategy areconcerned, the CP-ANNs [12, 146] are very similar to the Kohonen ANNs [35, 147]. In fact, each CP-ANN is composed of one Kohonen layer of neurons and also an additional layer with exactly the same number and layout of neurons as the Kohonen. The two layers of neurons are located precisely above each other. Therefore, the Kohonen and output neurons are in one-to-one correspondence. The CP-ANN isthus an up-grade of the Kohonen ANN. The major purpose of the CP ANNs set up is to enable the Kohonen type of ANN to solve the supervised type of problems. In this additional (output) layer of neurons, with exactly the same layout of neurons, the target vectors, i.e. the responses Ys associated with each object, are processed. Thus, the counter-propagation network consists of two layers with the same number and the same layout of neurons. The only difference between both layers is the number of weights in the corresponding neurons. The first layer of neurons processes the objects Xs, therefore the neurons in this layer have as many weights as there are in input variables. During the learning period, the target vectors $\mathbf{y}_s$, are placed into the second layer of neurons where they are processed in exactly the same manner as the objects Xs in the Kohonen layer. Evidently, the neurons of the output layer have as many weights as there in the YS vector as responses. However, after the learning process is completed and in the prediction process, the second layer becomes the output from which the resulting predictions are output. The learning procedure, i.e. the correction of weights is carried out in such a way that for each input object only the weights of two groups of neurons are corrected, one in the Kohonen and the other in the output layer. Both groups of neurons are identical in size and positionexactly one above the other concentrically around the central neuron excited by a given input vector Xs. The diameter or size of both groups on which the correction has to be performed, depends on the time of training. Whereas, the groups at the end of the training shrink to the size of one, i.e. of the selected neuron, both groups expand over the entire corresponding layer at the beginning of the training process. The criteria for the selection of the most excited or the central neuron is either the maximal response or

the minimal difference between the weights and the input variables [148, 149]. The later criterion is the most widely used, thus:

$$\text{central neuron c} = \min\left\{\sum_{i=1}^{m}(x_i - w_{ji})^2\right\} \quad (26)$$

for all j belonging to the network, m being the number of input variables and

$$\Delta w_{ji}^1 = \eta(t)a(c, j_{max}, j)(x_i - w_{ji}^{1\,old}) \quad (27)$$

$$\Delta w_{ji}^2 = \eta(t)a(c, j_{max}, j)(x_i - w_{ji}^{2\,old}) \quad (28)$$

where Δw_{ji}^1 is the correction in the ith weight on the jth neuron from the ith layer (i = 1 and 2 stands for Kohonen and output layer, respectively); $\eta(t)$ is the learning rate which monotonically decreases during the training; a( ) is a correction depending on the topology and the position of the group of neurons to be corrected; c is the position of the neuron selected by Eq. (30); j_{max} is the size of the group of neurons to be corrected at a given training time; j is the neuron on which the weights are corrected; x_i is the ith input variable (i = 1, . . . , m); and y is a single target variable (EMA in our case). If there would be more responses in the target vector, $\mathbf{y}_s$, the target variable, y, would have an index.

In recent years, radial basis function (RBF) [150] neural networks have many applications due to the ability to perform nonlinear mapping of the physicochemical descriptors to the corresponding biological activity in the field of QSAR.

RBF networks (RBFNs) have some features: Learning in RBFN is supervised. The training is fast. The RBFNs are good for interpolation. The output nodes implement linear summation function as in an MLP. The hidden nodes implement a set of radial base functions (e.g. Gaussian functions). The RBFNs have three layers including input layer, hidden layer and output layer. The input layer only distributes the input signals to the hidden layer. The hidden layer includes RBF function that often uses a Gaussian function that is illustrated by a center (cj) and width (rj). The RBF acts by determining the Euclidean distance between input vector (x) and the radial basis function center (cj) and performs the nonlinear transformation with RBF in the hidden layer according to below:

$$H_j(x) = exp\left[-\left(\frac{\|x - c_j\|}{r_j}\right)^2\right] (29)$$

In which, H_j is the output of the *j*th RBF hidden unit. For the *j*thRBF, c_j and r_j are the center and width, respectively. The operation of the output layer is linear:

$$y_k(x) = \sum_{j=1}^{nh} w_{kj}h_j(x) + b_k (30)$$

where y_k is the kth output unit for the input vector **x**, w_{kj} is the weight connection between the kth output unit and the *j*th hidden layer unit and b_k is the bias.

There is a growing interest in the application of artificial neural networks in molecular modelling and QSAR. Neural network can apply to redundant descriptors well and to approximate any target function, but it is not worth using NN for linear functions. Also, it can handle partial lack of system understanding. Moreover, ANN has other advantages, such as creating adaptive models (models that can learn), adapting to unknown situations and it has powerful hybrid systems which are robust, flexible and easy to use. RBFs and BP neural network have been successfully applied in QSAR studies for prediction of different properties. For example, toxic effect on fathead minnows [151, 152], calcium channel antagonist activity [153], alpha adrenoreceptors agonists [154], air to water partitioning for organic pesticides [155], aldose reductase inhibitors [156], anti-nociceptive activity [157], and anti-HIV activity [158].

Here, we discuss how to apply SOM neural networks in virtual screening and diversity selection.

In 2006 [145], Paul Selzer and Peter Ertl, stated applications of Self-Organizing Neural Networks in virtual screening and diversity selection. A potent procedure for the analysis and modelling nonlinear relationships between molecular structures and biological activity are artificial neural networks (ANN) with the aim of recognizing which structural features are of pharmacological significance. In this study combination of neural networks and radial distribution function molecular descriptors helped to industrial pharmaceutical research. These applications contain the prediction of biological activity, compound selection for screening, and the extraction of representative subsets from large compound collections. They focus on Kohonen and counter-propagation artificial neural networks (CP ANN) and their role in the pharmaceutical development. They used molecular descriptors, computed from intramolecular atomic distances in 3D space, to illustrate the 3D shape of structures. 3D structural information used as the input needed for calibration the neural networks. The neural network clustered the compounds in a 2D map according to the similarity or diversity of their descriptors. For showing the input and output calibration data, an ANN vector is needed. The pharmacological properties of the molecule are assigned by spatial arrangement of pharmacophore features of structure. They used Radial Distribution Function (RDF) molecular descriptors, which state pharmacophore features by coding the arrangement of atomic properties in 3D space as a vector of real numbers. The RDF code of a molecule is defined as a curved histogram of all of the intramolecular atom distances that take place and can be translated as the probability distribution of discovering atom pairs at an R distance.

$$g(R) = \sum_{i=2}^{n} \sum_{j=1}^{i-1} a_i a_j \, e^{-B(R-r_{ij})^2} \quad (31)$$

where n is the total number of atoms, a_i and a_j are atomic properties of atoms $i$and j and r_{ij} is the distance between atoms i and j. B is a smoothing factor which can be interpreted as a temperature parameter defining the fuzziness of atom positions because of thermal movement. B value equal to 100 gave rational results. The 3D atomic coordinates were measured using the 3D structure creator CORINA. As an extension of the initial RDF code idea where the code is computed between all occurring atom pairs, the RDF code is calculated 3 times: first, where both atoms have negative charges, for them, second, where1 atom possess a negative and the other one a positive charge, and third, where both atoms own negative charges. These three codes were bonded to form the last structure descriptor. In order to compute time during network training in the standard type, the RDF code in bins of 0.3 Å was computed. CP ANN consist of a 2D arrangement of x×y connected neurons. Each neuron contains z weights. The count of weights z related to the dimensionality of the structures, illustration involving an input part (structure representation) and an output part (biological activity). During calibration, the network learns inductively about the correlation between input and output by analyzing a so-called training set. The training set is give the network several times. In each round, for each molecule, the most common neuron, the winning neuron, is defined by determining the Euclidian distance between the structure descriptor and the input parts of the neurons. Then, the neuron weights are corrected to become more similar to the calibration data. The winning neuron is adjusted to the highest level. Once trained, the ANN has the capacity to predict the property for test set compounds which are excluded during training. The most similar neuron for each test structure is assigned by calculating the Euclidian distance between the neuron weights and the molecular descriptor in a test step.

It permits the submission of a set of molecules as a SMILES or SD file, or as a generic set of descriptors in text format. After the training procedure, results are demonstrated as a colour coded map presenting the distribution of the molecules among the neurons. A significant benefit of this method is the ability to interactively analyze the results. The comparison of input and output layers permits an assessment of the importance of certain input variables with regard to the output.

Clearly, neural networks must compete with other statistical techniques such as partial-least-squares analysis, support vector machines, and nearest neighbour methods, which might be quicker in performing equally well or, in some cases, even superior. But, the importance preference of Kohonen and CP- ANN is that they give fast and intuitive feedback about the results of the cheminformatics research. This visual opinion is a key aspect for the acceptance of this technique because it meets the requirements of the researcher, like chemical structures, in a graphical manner. Neural networks can do things which would be very difficult to be done using traditional computing techniques.

Bayezian neural networks

Most of the regression methods are prone to over training like artificial neural networks are. Bayesian criteria can overcome these disadvantages as they control model complexity by providing an objective criterion to signal for stopping the training. Controlling the complexity of the neural network is carried out by setting up a penalty on the greatness of the network weights. In these methods, namely Bayesian regularized artificial neural networks with a Gaussian prior (BRANNGP),a Gaussian prior $\sum_{j=1}^{N_w} w_i^2$ is commonly applied to control the complexity of models [159-161].To achieving an optimum sparsity in these and linear models, they use an expectation maximization algorithm [162]. There are some successful application of these methods in QSAR modeling of diverse data sets [163-166]. Further improvement of these methods has been done by using a sparsity-inducing Laplacian prior$\sum_{j=1}^{N_w}|w_j|$ denoted as Bayesian regularized artificial neural networks with a Laplacian prior, BRANNLP) [162, 167] which makes the irrelevant weights in descriptor space to be set to zero, and just the meaningful remainder defines the model. Therefore, both the less relevant descriptors and the number of effective weights in the neural network model are set to the optimal level.

Support Vector Machine

Support vector machine (SVM) is a machine-learning technique used for resolving the classification and regressionproblems. In recent years, SVM as a relatively novel approach and a powerful modelling tool, showed a good performance compared to the old methods including neural networks. The foundations of SVM have been set up by Vapnik and his team [168, 169]. A least squares version for support vector machine modified by Suykens [170, 171] that its main advantage is to learn faster and easier. In fact, support vector machine is a binary classifier, which separate two classes by linear boundary. The basic idea of SVM was to construct an optimal separating hyperplane as decision level for determination of threshold separating between different data points or components. Using an optimization algorithm, the samples that constitute the boundaries of classes are called support vectors. This method is based on the supervised learning models. Thus a number of training data that are the least distance from decision boundary conceding as subset for decision boundary and support vectors. This method can be applied for linear and nonlinear analysis. In figure 12 two classes and their support vectors were shown. To separate the two classes, or in other words, to calculate the decision boundaries between two

classes, the margin optimization is used. This margin is defined as the distance of the closest data point from the separating hyper plane.

A linear decision boundary in general can be written as follows:

$wx + b = 0$ (32)

x is a point on the boundary decision, w is a n-dimensional vector the perpendicular to the boundary decision.

In this method, the boundary line between the two classes is calculated with these conditions:

1- All samples of class +1 are located on one side of the border, and all samples of class -1 are located on the other side of the border.
2- Decision boundary in such a way that the distance between the nearest training data points in the direction perpendicular to the boundary between the two classes of decisions to be maximized.
The first condition can be written as follows:
$\boldsymbol{w}^T x_i + b \leq -1 \;\; if y_i = -1$
$\boldsymbol{w}^T x_i + b \geq 1 \;\; if y_i = +1$
$y_i(\boldsymbol{w}^T x_i + b) \geq 1$

The distance of separator line from the origin is:

$r = \frac{y_s(w^T x_s + b)}{\|w\|}$(33)

So the distance between two margins is given by the following equation:
$\rho = 2r = \frac{2}{\|w\|}$(34)
The maximizing $\frac{2}{\|w\|}$ is given by minimized $\frac{1}{2} w^t w$
So, the separator line is obtained by solving the following equation:
Minimize $\frac{1}{2}\boldsymbol{w}^t \boldsymbol{w}$
Subject to $y_i(\boldsymbol{w}^T x_i + b) \geq 1$

Getting w and b from the solution of this equation require complex calculations.

To simplify it, the optimization problem could transform using the method of Lagrange's undetermined multipliers the below form. α is Lagrange multipliers.

Maximize: $W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j + \sum_{i=1}^{N} \alpha_i$

Subject to: $\sum_{i=1}^{N} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, \dots N$

After solving the optimization problem by Lagrange multipliers, w is calculated using the following equation:

$w = \sum_{i=1}^{n} \alpha_i y_i x_i$(35)

Due to existence of noise and error in measurement in real systems, the flexibility margin should be used so that the outlier would not cause wrong boundary. To this end, a parameter error is added to the equation:

Minimize $\frac{1}{2}\|w\|^2 + \gamma \sum_{i=1}^{N} e_i$

Subject to $y_i(w.x_i + b) \geq 1 - \xi_i, fori = 1, \dots, N$

$$\xi_i \geq 0, \quad fori = 1, \dots, N$$

γ parameter is related to the effect of error on margin, that should be optimized by user. When the classes are overlap, spearing of classes by a linear boundary always give an error. To overcome this problem data can be mapped the input space into a high dimensional feature space by choosing a suitable choice of kernel function that is a non-linear mapping (Figure 13). In new space the data have a less interference with each other. Nonlinear functions can be employed including polynomials, radial basis function and certain sigmoid function, that the most common of them is RBF function.

$\varphi = \exp[-(\frac{\|x_i - x_j\|}{2\sigma^2})^2]$(36)

The advantage of the SVM is that, by use the kernel trick, the distance between a molecule and the hyperplane can be calculated in a transformed (nonlinear) feature space, but without requiring the explicit transformation of the original descriptors. A variety of kernels have been suggested, such as the polynomial and radial basis function (RBF).

In the nonlinear case we can one employs the "kernel trick" to express the dot products and the mappings Φ into the Hilbert space:

Φ:R d ⇛H (37)

in terms of some kernel function of the form

$K(x_i, x_j) = \Phi(x_i).\Phi(x_j)$ (38)

By this method, wherein the dot products are defined in the new space as a single function, it becomes unnecessary to determine Φ explicitly. This is especially useful in the case of the commonly-used radial basis function:

$K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$ (39)

(forγ>0), which renders H infinite-dimensional.

The advantage of the SVM is that, by use the kernel trick, the distance between a molecule and the hyperplane can be calculated in a transformed (nonlinear) feature space, but without requiring the explicit transformation of the original descriptors. A variety of kernels have been suggested, such as the polynomial and radial basis function (RBF).

Another application of SVMs is solving regression problems by introduction of an alternative loss function. Quadratic, Laplace, Huber and ε-insensitive are employed as loss functions that must be modified to include a distance measure. SVR works effectively on both linear and non-linear problems. In the same manner as the non-linear SVC approach, a non-linear mapping can be used to map the data into a high dimensional feature space where linear regression is performed using various kernel functions.

SVM regression relation is as follows:

$$Q = \frac{1}{2}\boldsymbol{W}^T\boldsymbol{W} + \frac{1}{2}\gamma\sum_{i=1}^{n} e_i^2 \qquad (40)$$

Subject to: $y_i = \boldsymbol{W}^T \emptyset x_i + b + e_i i = 1, \ldots, n$

y_i is the dependent variable, the best relation is the one that minimizes the cost function (Q) containing a penalized regression (e_i) error term.

Unlike excellent predictive power of SVMs, they are not interpreted simply, and little work has been carried out in this area in comparison to trees or neural networks. Performance of SVMs on various data sets is good so they are common in a variety of disciplines. The disadvantage of this approach is that the model builds time as a result of the quadratic programming step of the algorithm for creating a SVM. Their prediction ability is high due to natural avoiding local minima.

Now, SVM is established in the drug design field. For example, Briem&Günther [172] , to predict the likeness of a molecular compound to be a kinase inhibitor, developed support vector machine (SVM) models. Jorissen& Gilson [173] described the application of SVM models for virtual screenings.

Tomohiro Sato et al. [174] used support vector machine to 3D shape-based virtual screening using complete 3D molecular shape overlay with known inhibitors of 15 target proteins extracted from the ChEMBL database in order to progress the screening efficiency. The descriptor in this study was 3D similarity profile of a compound and was described as the array of 3D shape similarities with multiple known active compounds and was used as the explanatory variable of SVM. The prediction ability of the 3D shape similarity metrics presence in ROCS (rapid overlay of chemical structures), such as ShapeTanimoto and ScaledColor, were validated, using the inhibitors. The idea of empirical kernel map was used to calculate the descriptors, using reference structures which are known active compounds. In order to compute the input data for SVM, every compound in a database was overlaid on all known active compounds by ROCS and was exhibited by a vector involving of the resulting 3D shape similarities to the active ones. SVM models classify two classes of compounds nonlinearly, by mapping the data vectors to a high dimensional descriptor space and finding the hyperplane that decollated the two classes with the largest margin. “kernel trick” is the major variety between SVM and simple linear discrimination. In this study, a radial basis function (RBF) kernel was used to gain a complex nonlinear separating hyperplane. The gamma for the RBF kernel and the “C” value of the constant for the slacks variant were optimized by 5-fold cross-validation.

For obtaining the descriptors, first, a compound from the target database was aligned on all of the active compounds, and the similarity metrics, such as ShapeTanimoto or ScaledColor, with the active compounds were computed by ROCS. Then, the 3D similarity profile was obtained, by arraying the measured 3D shape similarity data (Figure 14). To assess the efficiencies of screening, active and decoy compounds in the test set were superposed to the active molecules in the training set, and the obtained 3D similarity profiles of the test set were entered into the SVM models, for predictions. Among the 3D similarity metrics, ScaledColor showed the best screening efficiencies, for both the enrichment factor at 1% (EF1) value and receiver operation characteristic curves (ROC) score, on average for the 15 target proteins. Totally, the pharmacophore-based metrics, such as ScaledColor and ColorTanimoto, showed better results than the shape-based metrics. The SVM models using ScaledColor for the 3D similarity profiles, for both the SVM models and usual similarity approaches indicated the best results. In general, the findings show that the machine learning approaches for the efficiency of virtual screening could handle plural information sources correctly.

Kinnings et al. [38] showed using support vector machines (SVMs) can improve predicted activities by the docking program eHiTS. They constructed two SVM models: Firstly, a regression support vector machine, which was derived using IC50 data from 80 Mycobacterium tuberculosis (M.tb) InhA inhibitors extracted from BindingDB (http://www.bindingdb.org/). Training the model was carried out using individual energy terms as features from the eHiTS docking software. The eHiTS give 20 energy terms contributed in the overall energy score. These scores were drawn from the output of the docking of 80 different InhA inhibitors into InhA. In order to resolve the optimal combination of energy terms (features) for regression modelling, different combinations were examined. Comparison of the relative accuracies of different combinations of features was done by Five-fold cross-validation. The combination of energy terms that gave the highest mean Spearman's rank correlation coefficient was

selected for the model. A classification model trained using 85 active molecules and 3035 decoy molecules from Directory of Useful Decoys (DUD) which includes actives and decoys in a ratio of approximately 1:36 and there is a strong bias to negative examples in the training samples. To cure this situation, a multiple-planar classification model was created (Figure 15). In this procedure, set of decoys was randomly partitioned into an n subsets and were constructed n model of combination n different negative subsets and positive set. In order to find the optimal value of n, 5-fold cross-validation was performed for various values of n from 5 to 36. In each of five iteration F-score was computed, and the mean F-score was assigned to that particular value of n. number of partitions of decoy set with highest F-score was selected. Each of models predicted a score for each molecule in the test set, and a weighted voting method was used for sum of the n scores that was assigned to each molecule. If the sum of the n scores was greater than zero, the molecule was considered active, and if the sum of the n scores was less than zero, the molecule was considered decoy. Performance of both regression and classification SVM models were better than original eHiTS scoring function .then they employ SVM to train a new scoring function for direct inhibitors of Mycobacterium tuberculosis (M.tb) InhA. and were able to increase the accuracy of virtual screening of direct InhA. inhibitors by this new function , and proposed that posphodiesterase inhibitors can bind potentially to target.

The Relevance Vector machine (RVM) proposed by Tipping[175] is a related but sparser classification and regression method, based on Bayesian statistics, which showed considerably better properties than SVM. Sparser models generated by such methods are able to make generalization to new data compared to models with less sparse properties.

Some of disadvantages of SVM methods are as follows[176]:

• SVMs are not optimally so while are relatively sparse.

• Predictions are not probabilistic. SVM provides a single value in regression while in the classification renders a deterministic binary decision. Ideally, it would be beneficial to capture the uncertainty of predictions.

• A cross validation procedure is necessarily used to estimate SVM parameters that is wasteful of time, data, and computation.

The RVM theory identifies how the algorithm overcomes the disadvantages of SVM, it is shown that for relatively diverse data sets RVM models provides models usually sparser, more predictive, or both compared to SVM models of the same data using a same data set of descriptors. Although, RVM is an iterative method, training times are minimal for the data set sizes employed here, and the increased sparsity, which generates benefits in terms of predictive powerand, arguably, interpretability, makes the small increase in computational effort worthwhile.

Classification and regression trees (CART)

CART is a non-parametric unbiased statistical strategy, which solves classification and regression problems. Classification and regression trees are machine-learning approaches to build prediction models from data, where trees are oriented graphs starting with one node and branching to many. In models based on classification trees, the dependent variable is categorical, while in the regression ones it is continuous. Indeed, each classification tree can be translated into a collection of predictive rules in Boolean logic. [177, 178]. CART is used to model the data space and fit a simple prediction model within each division. It is an alternative approach to nonlinear regression and sub-divide, or partition, the data space into smaller regions, where the feature interactions are more manageable. The sub-divisions then are partitioned again, this is recursive partitioning, until finally it would not possible to model the space. For classification and regression trees, the model in each region is just a constant estimate of Y. That is, if there are some points (x1; y1); (x2; y2) . . . (xc; yc) all belonging to a same leaf-node. Then the model for this is just the sample mean of the response variables in that cell. Classification or regression trees do not need to be binary, but most are. There are three type nodes in a decision tree: a root node, internal nodes, and terminal nodes. The root node in top holds the entire training samples and does not have any incoming branches. An internal node has one incoming branch containing a subset of the samples in the node directly above it and two or more outgoing branches. Furthermore, it includes the total of the samples in the nodes connected to and directly below it. Finally, the terminal or leaf nodes are, with one incoming branch and no outgoing branches. A node may assign to be a leaf if compounds directed to it fall into a single activity class or at least one class produces a clear majority. Typically, a single descriptor is applied as a condition to do a test in each node. Each leaf node would be assigned with a target property namely the activity class related to the leaf. However, assignment of each non leaf node (root or internal node) is done with a molecular descriptor. To select the most suitable descriptor for a split and its split value, an algorithm is used in which all descriptors and all split values are regarded. The split in which the best reduction in impurity between the mother group (tp) and the daughter groups (t_L and t_R) takes place is selected.

$$\Delta i(s, t_p) = i_p(t_p) - PL_i(t_L) - PR_i(t_R) \quad (41)$$

wherei is the impurity, s the candidate split value, and PL and PR are the portions of the objects in the left and the right daughter group, respectively. For regression trees, the impurity i is usually determined as the total sum of squares of

the deviations of the individual responses from the mean response of the group in which the considered molecule is assigned [179, 180].

$$i(t) = \sum_{n=1}^{n}(y_n - \bar{y}(t))^2 \qquad (42)$$

where $i(t)$ is the impurity of group t, y_n is the value of the response variable for sample x_n and $\overline{y_t}$(t) is the mean value of the response variable in group t.

Mathematically this is expressed as: The result of the test, determines the direction of the algorithm to one of the child nodes branching from the parent. By repeating the procedure, traversal of the tree towards the leaves is done. The training of the model is carried out by incremental addition of nodes. The CART procedure consists of three major steps. First is that in a forward stepwise procedure a complete decision tree is built from data. It means each parent node is divided into two child nodes by a best splitter. This is done by labelling the interior nodes with questions, and edges or branches by the answers (yes or no in classic versions) to determine the left and right regions: the split for continuous variables is defined by "$x_i<a_j$" where $x_i$is the selected explanatory variable and $a_j$its split value. Every recursive algorithm needs to have a stopping criterion. A more typical criterion may be that halt when each child would contain less than a predefined value (five) data points, or when addition of information by splitting would be less than some threshold. Picking the criterion is important to obtain a good tree. A case is that the tree will grow and splitting continues until sum of squared differences of all y values of objects in that node reaches to less than a predetermined threshold (fully grown tree).

Tree-pruning

Addition of more variables produces a chance of over fitting, resulting in poor predictions for unknown samples. Then, the second step consists of 'pruning' the tree that is via removing extra variables. Size of the tree is determined by counting the terminal nodes. A complexity value for instance the sum of squares of differences can be used for pruning the tree. Within the pruning procedure, terminal branches are cut successively to obtain a series of smaller subtrees from the maximal tree. Then a comparison is made between different subtrees to find the optimal one. This comparison is on the basis of a cost-complexity measure, in which both tree accuracy and complexity are regarded.

Selection of the optimal tree

Now that the optimal tree has to be selected among the obtained sequence of sub trees, the final job is to make a balance between the performance of the tree (the right size) and its complexity. It is often based on the evaluation of the predictive error by performing the cross-validation algorithm or by using an external test set [39]. In an n-fold cross-validation (CV), 1/n parts of the data from the training samples, one after another, are removed till all parts removed once and just only once. These are used as a test set to assess the predictive power of the tree, build with the left over data [181]. Then the tree with the smallest mean CV error is accounted as the most accurate tree, defined by the RMSECV:

$$RMSECV = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}} \qquad (43)$$

wherey_i is the response value of sample i, $\hat{y}_i$ is the predicted response value for object i and n is the total number of objects.

Decision Trees have been tested in some studies [182, 183]. In several datasetsrelated to ecotoxicity, decision trees usually achieved lower error than LDA or kNN methods [59]. There are also applications of decision trees, in anti-HIV activity [184], toxicity [185] and oral absorption [186].

Ensemble learning based methods

Random Forests

There is usually relatively low prediction accuracy for decision tree (DT). This downside may hamper usefulness in applications such as virtual screening of compound libraries. Many efforts have been done by many research groups to improve its prediction accuracy. The outputs of these attempts led to a large number of various tree-based algorithms [187]. Account for the fact that one of the best ways to improve the performance of DT based algorithms is to use ensembles of trees; RF was presented [9, 188]. This perspective concerns that the performance of an ensemble of not-exactly-tuned diverse trees as regressor would be better than that of a well-tuned single tree. RF is a powerful tool which is able to present performance and is among the most accurate methods that grows many classification trees. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). RF is an ensemble of B trees {T1(x), ..., TB(x)}, where **x** = {x_1, ..., x_p} is a p-dimensional vector of molecular features or properties related to a molecule. The ensemble produces B outputs {$\hat{y}_1 = t_1$(X), ...,$\hat{y}_B$=t_B(X)} where$\hat{y}_b$, b = 1, ..., B, is the prediction provided by the bth tree for a molecule. Outputs of all trees are aggregated to produce one final prediction,$\hat{\mathbf{y}}$. For classification problems, $\hat{\boldsymbol{y}}$is the class predicted by the majority of trees. In regression it is the average of the individual tree predictions. Some of recently applications of RF method in QSAR can be found here [189, 190].

Training Procedure

The tree growing algorithm used in RF is unpruned CART, although other alternatives could be considered as well. Suppose a data set with n molecules for training, in which D =

$\{(x_1, y_1), ..., (x_n, y_n)\}$, and $\mathbf{x}_i$, i = 1, ..., n, is a vector of descriptors and $\mathbf{y}_i$can be either the corresponding class label (e.g., active/inactive) or activity value (e.g., -logIC_{50}), the steps of training algorithm can be as follows: (1) A bootstrap set (i.e., randomly selected samples, with replacement, n molecules) is drawn from the training set of n molecules. (2) In each bootstrap sample, a tree with the following properties is grown: at each node, instead of testing the performance of all variables, the best division among a randomly selected subset of m_{try} descriptors is chosen. Here m_{try} is essentially the only tuning parameter in the algorithm. Each tree grows to reach the maximum size (i.e., until no further splits would be possible) and not pruned back. (3) Repetition of the above steps is carried out until (a sufficiently large number) B such trees are obtained. When m_{try} = p, i.e., the best split at each node is selected among all descriptors, the RF algorithm would be the same as Bagging [191]. The RF algorithm can be very efficient, especially when the number of descriptors is very large. The reasons of this efficiency compared to that of growing a single decision tree, may be related to differences between the two algorithms. The first is that in a usual tree growing algorithm, splitting performance of all descriptors is tested at each node, while in RF only m_{try} of the descriptors is tested. Since m_{try} has typically a very small value (the default in the software is the square root of the number of descriptors for classification), the search would be very fast. Second, to attain the optimal prediction strength in a right model complexity some pruning is usually required for a single DT. Cross validation does the job which can consume a major portion of the computations. RF, on the other hand, does not perform any pruning at all. It is found that when there are an excessively large number of variables; training of RF can be done in less time other than a single decision tree.

Estimation of performance of RF by out-of-Bag

An ideal way of doing assessment of performance for a prediction algorithm is via using a large independent test data set. In practice, when the number of samples is limited, some type of cross-validation [192] is usually used, which, in some cases, would be computationally cumbersome. Cross-validation in RF is performed in parallel with the training step by using the so-called Out-Of-Bag (OOB) samples [193]. Specifically, in the process of training, each tree is grown using a particular bootstrap sample. Because bootstrapping is part of sampling with replacement from the training data, some of the objects will be "left out" of the sample, while some others will be repeated in the set. So, the training samples are bootstrapped randomly with replacement to about two thirds of the original training set as the in-bag set and the remaining one third of the samples, the "left out" molecules (DOOB), as the OOB samples [194]. Because of not using OOB molecules in the tree construction, there can be a possibility to use them to estimate the ensemble prediction performance in the following method. If $D_b^{OOB}$is the OOB part of the data for the bth tree, it will then be possible to use the bth tree to predictD_b^{OOB}. As each training molecule, x_i, is in an OOB object, in average, about 1/3 of the time, it would be possible to calculate an ensemble prediction $\hat{\mathbf{y}}^{OOBB}(x_i)$by aggregating only its OOB predictions. An estimation of the error rate for classification or mean square error (MSE) for regression is carried out by

$$\text{Error rate} \approx (Errorrate)^{OOB} = n^{-1}\sum_{i=1}^{n} I\left(\hat{Y}^{OOB}(Xx_i) \neq y_i\right) \quad (44)$$

$$MSE \approx MSE^{OOB} = n^{-1}\sum_{i=1}^{n}\{\hat{\mathbf{y}}^{OOB}(x_i) - y_i\}2 \quad (45)$$

where I(.) is the indicator function. In reality, OOB performance compared with n-fold cross-validation shows that they are in reasonably good agreement, so that the assessment of RF performance indeed does not necessitate additional cross-validation.

Other ensemble learning based methods

There are some other well-known ensemble learning methods in addition to RF, namely bagging [191] and boosting [194, 195]. Both these two methods have also been proven to significantly get better performance over that of a single tree. Bagging can be regarded as a RF when m_{try} = p. In boosting, there is a sequence of trees. While all the data have been used to train each of them, data points have been reweighted in each tree consistent with whether they were misclassified by the previous tree in the sequence, for classification. On the other hand in regression, growing of each tree is based on the residuals of the previous trees. For prediction, weighted vote (in classification) or weighted average (in regression) of the ensemble outputs is carried out. Moreover, there has been proposed actually several implementations of boosting, such as Freund and Schapire's Friedman's stochastic gradient boosting (SGB) [196], and many others [197, 198]. There are several evidences that boosting and RF usually do better than bagging [199]. It ought to be mentioned two other ensemble methods because of having been used in QSAR modelling: random FIRM [200] and decision forest [187]. In random FIRM, each ensemble of trees is built on all the training data, but at each split, a variable is randomly chosen consistent with probabilities related to the descriptor's significance from an appropriate statistical test defining the split. While random FIRM surely has ability to be used for prediction, using one of the tree aggregation procedures, its application is basically for model interpretation. Decision forest is another ensemble based method built such that descriptors used by each tree arenot available to the other trees although such that the prediction accuracy of each tree does hold above a specified threshold. In a comparison, made between decision forest and RF, it is shown that they have similar performance on one data set. In recursion forest [201], an ensemble of trees all is grown on the same training data, while the tree growing parameters are

systematically varied to produce different trees. A consensus selection procedure is then applied to calculate a boolean intersection of the outputs. Adaboost [202] is the first applicable approach of boosting, and it has been appointed as one of the top ten data mining algorithms [203]. It is well known that it reduces bias (besides from variance) [204], and like to SVMs boosts the margins [205]. AdaBoost uses the whole data-set to train each classifier serially, but after each round, it pays more attention to difficult cases, with the aim of correctly classifying examples in the next round that were incorrectly classified during the current iteration. Hence, the more focus is given to instances that are harder to classify. The quantity of focus is determined by a weight, which at first is equal for all samples. At the end of each round, the weights of misclassified instances are increased while the weights of correctly classified instances are decreased. In addition, another weight is assigned to each individual classifier depending on its overall accuracy which is then used in the test phase; more confidence is given to more accurate classifiers. Finally, when a new instance is submitted, each classifier gives a weighted vote, and the class label is chosen by majority. Lastly, there is a current work underway on building ensembles of trees grown using evolutionary programming rather than recursive partitioning [185]. In QSAR, the boosting method has been found useful in modelling the COX-2 inhibition, estrogen and dopamine receptor binding, multidrug resistance reversal, CDK-2 antagonist activity, BBB permeability, logD and P-glycoprotein transport activity [183]. In comparison with RF, it showed better results for several datasets but worked worse for the others. Bagging and other similar ensembles were used in QSAR [206-208]. In another case, the kNN and decision trees were used as base methods in bagging for prediction of drug likeness [209].

CART and LDA in practice

To illustrate how CART and LDA work, a study from Zang et al. has been selected [210]. The ToxCast and Tox21 programs have tested ~8200 chemicals in a broad screening panel of in vitro high-throughput screening (HTS) assays for estrogen receptor (ER) agonist and antagonist activity. In this study, the authors exploited the HTS data got from ToxCast and Tox21 programs for 8000 environmental chemicals including pesticide active and inert ingredients; industrial chemicals such as solvents, surfactants, and plastics; cosmetics and personal care ingredients; food additives; and pharmaceuticals. They developed binary QSAR classification models that related chemical structures to estrogenic activity by the application of three machine learning methods, LDA, CART, and SVM. Training compounds from the ToxCast project were categorized as active or inactive (binding or nonbinding) classes based on a composite ER interaction score derived from a collection of 13 ER in vitro assays. A total of 1537 chemicals from ToxCast were used to derive and optimize the binary classification models while 5073 additional chemicals from the Tox21 project, evaluated in 2 of the 13 in vitro assays, were used to externally validate the model performance. A total of 51 molecular descriptors were calculated using the QikProp software (Schrodinger version 3.2). All chemicals with available structures were also fingerprinted using publicly available SMARTS sets FP3, FP4, MACCS from OpenBabel, PADEL and PubChem. A total of 4328 bits of structural fingerprints were generated.

Table 6. Data sets used for classification study

Data set	Total chemicals	Active chemicals	Inactive chemicals	Active/inactive
Tox21	6610	435	6175	1:14.2
ToxCast	1537	264	1273	1:4.82
Training set (I)	1025	176	849	1:4.82
Internal test set (II)	512	88	424	1:4.82
External test set (III)	5073	171	4902	1:28.7

It is well known that a common problem in machine learning (ML) model building occurs when the training HTS data are highly imbalanced with only a small number of active chemicals compared to the number of inactive chemicals. Therea new approach was proposed to tackle the problem of class imbalance using what they term a “target-independent” clustering method. The limitations of the model’s predictive capability outside the training set were also examined. Feature selection was applied by using random forest, which ranks the importance of each descriptor in the classification process, and was useful in eliminating the unrelated and redundant descriptors to ER activities and improved the model’s prediction performance. The molecular descriptors captured important information and were more discriminative than fingerprints in thebinary classification. The models employing descriptors presented significantly superior results than those employing fingerprints. To assess the performance of these machine learning methods, it is useful to examine whether their prediction ability is at a similar level in terms of overall accuracy, sensitivity, specificity, and G-mean. The best model was derived from SVM with the optimal settings of the RBF kernel function and the set of descriptors selected by RF method. When compared with LDA and CART, the SVM classification model presented better statistics and produced improved results, not only in cross validation, but also in the prediction of two independent test sets, giving the highest sensitivity of 76.1% and specificity of 82.8% for the internal validation set. Although, CART achieved lower classification accuracy than SVM and LDA, and it is the simplest model with the best interpretability. These models were developed by using ToxCast data set with about 1500 known active and inactive chemicals, thereby, covers a small portion of the chemical space of the Tox21. The reliability of the prediction model was strongly dependent on the structural similarity

between training compounds and test compounds. Although, satisfactory results were achieved in both cross validation and internal tests, and the predictive power of the built model for a test set that is beyond the training chemical space, such as the Tox21 data set, should not be anticipated without caution. Overall, this research work suggests that the binary QSAR classification models are useful for *in silico* prediction of the estrogenic activity and for characterizing the molecular features of environmental chemicals. Also, highly accurate predictive models can be built based on chemical descriptors. These models can be applied in virtual screening of large databases for identifying compounds with potential risk at a reduced cost.

Nd-QSAR methods

With introduction of comparative molecular field analysis (CoMFA) [211] in 1988 for the first time, structure–activity relationships were presented based on the three-dimensional structure of the molecules (3D-QSAR) [212, 213]. In this way, a series of compounds superimposed in 3D space are placed onto a surface or grid which mimics the binding site of the true biological receptor. Ligands' interaction with chemical probes is then gathered as descriptors. However, in traditional 2D-QSAR models descriptors are derived from a two dimensional graph representation of a molecule. The determining factor to the quality of a 3D-QSAR model is the correct alignment of the ligands, the identification of which is almost impossible in the absence of structural information for the target protein. Though, some 3D-QSAR methods are alignment independent such as a method that uses GRid-INdependent Descriptors (GRIND). GRIND are among field based features that are obtained starting from a set of molecular interaction fields [213, 214]. Many other 3D-QSAR methods have been developed thereof that a comprehensive explanation of them can be found in literature [215, 216]. While the alignment problem has long been recognized, 4D-QSAR approaches would seem to provide decent solutions [217, 218]. This concept approaches the alignment problem by incorporating molecular and spatial variety by representing each compound in differentconformations, orientations, tautomers, stereoisomers or protonation states. The true binding mode (or the bioactive conformation) is then identified by the algorithm underlying the QSAR concept. Furthermore, it can have fundamental biological relevance, in the case of multi-mode binding targets [219]. While this method profoundly reduces the bias with choosing a bioactive conformation, orientation, or protonation state, it still needs to a "sophisticated guess" about appearance and importance of the associated local induced fits, the adaptation of the receptor binding pocket to the individual molecule topology. The fifth dimensions allowing for a multiple representation of the topology of the quasi-atomistic receptor surrogate is considered in 5D-QSAR approach [220, 221]. Extension of dimensions to six in 6D-QSAR allowed researchers to simultaneously consider different solvation models. This can be obtained explicitly by mapping parts of the surface area with solvent features which genetic algorithm was used to optimize the position and size, or implicitly. 6D-QSAR has been applied on 106 diverse molecules binding to the estrogen receptor and the results suggested that this approach is appropriate for the identification of an endocrine-disrupting potential related to drugs and chemicals [222].

Conclusion

QSAR has been applied successfully over several decades to find predictive models to solve problems in the fields of drug design, toxicity, risk assessment and etc. The scientific community is displaying more and more notice in the QSAR field. QSAR models should be seen together with their components not alone. One part is to choosinga right model developing method. Hereby, we have discussed several multivariate analyses methods to establish regression models from linear to nonlinear and supervised to unsupervised regression methods. Nowadays, the need to deal with systems biology and complex systems pushes further toward creatingnew borders where mathematics, and statistics is applied to produce new effective useful knowledge in the field of multivariate methods. Meanwhile, we observed very efficient tools and methods for generating molecular descriptors and different validation methods.These are two of the most remained challenging parts of QSAR modelling by multivariate methods. Then, it is mandatory to generate suitable chemical descriptors to account useful and focused chemical information corresponds to an understudy problem and to develop statistical tools to approve the accuracy and reliability of the obtained QSAR models.

References

.

1. L. Eriksson, J. Jaworska, A. P. Worth, M. T. Cronin, R. M. McDowell and P. Gramatica, *Environmental health perspectives*, 2003, **111**, 1361.
2. C. Hansch, A. Leo, D. Hoekman and A. Leo, *Exploring Qsar*, American Chemical Society Washington, DC, 1995.
3. R. D. Cramer, J. D. Bunce, D. E. Patterson and I. E. Frank, *Quantitative Structure-Activity Relationships*, 1988, **7**, 18-25.
4. M. Karelson, *Molecular descriptors in QSAR/QSPR*, Wiley-Interscience, 2000.
5. T. Cheng, Q. Li, Y. Wang and S. H. Bryant, *Journal of chemical information and modeling*, 2011, **51**, 229-236.
6. P. Vasanthanathan, O. Taboureau, C. Oostenbrink, N. P. Vermeulen, L. Olsen and F. S. Jørgensen, *Drug Metabolism and Disposition*, 2009, **37**, 658-664.
7. M. Carbon-Mangels and M. C. Hutter, *Molecular Informatics*, 2011, **30**, 885-895.
8. Y. Xue, Z.-R. Li, C. W. Yap, L. Z. Sun, X. Chen and Y. Z. Chen, *Journal of chemical information and computer sciences*, 2004, **44**, 1630-1638.
9. V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan and B. P. Feuston, *Journal of chemical information and computer sciences*, 2003, **43**, 1947-1958.

10. A. Tropsha, P. Gramatica and V. K. Gombar, *QSAR & Combinatorial Science*, 2003, **22**, 69-77.
11. D. M. Hawkins, *Journal of chemical information and computer sciences*, 2004, **44**, 1-12.
12. J. Zupan, M. Novič and I. Ruisánchez, *Chemometrics and Intelligent Laboratory Systems*, 1997, **38**, 1-23.
13. R. W. Kennard and L. A. Stone, *Technometrics*, 1969, **11**, 137-148.
14. A. Golbraikh and A. Tropsha, *Molecular diversity*, 2000, **5**, 231-243.
15. A. Golbraikh, M. Shen, Z. Xiao, Y.-D. Xiao, K.-H. Lee and A. Tropsha, *Journal of computer-aided molecular design*, 2003, **17**, 241-253.
16. W. Wu, B. Walczak, D. Massart, S. Heuerding, F. Erni, I. Last and K. Prebble, *Chemometrics and intelligent laboratory systems*, 1996, **33**, 35-46.
17. T. M. Martin, P. Harten, D. M. Young, E. N. Muratov, A. Golbraikh, H. Zhu and A. Tropsha, *Journal of chemical information and modeling*, 2012, **52**, 2570-2578.
18. O. Roche, P. Schneider, J. Zuegge, W. Guba, M. Kansy, A. Alanine, K. Bleicher, F. Danel, E.-M. Gutknecht and M. Rogers-Evans, *Journal of medicinal chemistry*, 2002, **45**, 137-142.
19. A. C. Brown and T. R. Fraser, *Transactions of the Royal Society of Edinburgh*, 1868, **25**, 151-203.
20. E. Overton, *Z. phys. Chem*, 1896, **22**, 189-209.
21. T. Fujita, J. Iwasa and C. Hansch, *Journal of the American Chemical Society*, 1964, **86**, 5175-5180.
22. C. Hansch, R. M. Muir, T. Fujita, P. P. Maloney, F. Geiger and M. Streich, *Journal of the American Chemical Society*, 1963, **85**, 2817-2824.
23. C. Hansch, P. P. Maloney, T. Fujita and R. M. Muir, 1962.
24. A. T. Balaban and F. Harary, *Journal of Chemical Documentation*, 1971, **11**, 258-259.
25. M. Randić, *The Journal of Chemical Physics*, 1974, **60**, 3920-3928.
26. M. Randic, *Journal of the American Chemical Society*, 1975, **97**, 6609-6615.
27. L. B. Kier, L. H. Hall, W. J. Murray and M. Randi, *Journal of pharmaceutical sciences*, 1975, **64**, 1971-1974.
28. H. Timmerman, R. Mannhold, P. Krogsgaard-Larsen and H. Waterbeemd, *Chemometric methods in molecular design*, John Wiley & Sons, 2008.
29. J. Neter, M. H. Kutner, C. J. Nachtsheim and W. Wasserman, *Applied linear statistical models*, Irwin Chicago, 1996.
30. S. De Jong, *Chemometrics and intelligent laboratory systems*, 1993, **18**, 251-263.
31. Y.-L. Xie and J. H. Kalivas, *Analytica chimica acta*, 1997, **348**, 19-27.
32. H. Wold, *Encyclopedia of statistical sciences*, 1985.
33. P. J. Gilligan, G. A. Cain, T. E. Christos, L. Cook, S. Drummond, A. L. Johnson, A. A. Kergaye, J. F. McElroy and K. W. Rohrbach, *Journal of medicinal chemistry*, 1992, **35**, 4344-4361.
34. J. H. Friedman, *IEEE Transactions on Computers*, 1977, 404-408.
35. J. Zupan and J. Gasteiger, *Analytica Chimica Acta*, 1991, **248**, 1-30.
36. E. V. Ruiz, *Pattern Recognition Letters*, 1986, **4**, 145-157.
37. R. A. Fisher, *Annals of eugenics*, 1936, **7**, 179-188.
38. S. L. Kinnings, N. Liu, P. J. Tonge, R. M. Jackson, L. Xie and P. E. Bourne, *Journal of chemical information and modeling*, 2011, **51**, 408-419.
39. D. L. Massart, B. G. Vandeginste, L. Buydens, P. Lewi and J. Smeyers-Verbeke, *Handbook of chemometrics and qualimetrics: Part A*, Elsevier Science Inc., 1997.
40. R. U. Kadam and N. Roy, *Bioorganic & medicinal chemistry letters*, 2006, **16**, 5136-5143.
41. S. Wold, K. Esbensen and P. Geladi, *Chemometrics and intelligent laboratory systems*, 1987, **2**, 37-52.
42. E. R. Malinowski, 2002.
43. V. Barnett and T. Lewis, *Outliers in statistical data*, Wiley New York, 1994.
44. A. Tropsha, *Molecular Informatics*, 2010, **29**, 476-488.
45. M. T. Cronin and T. W. Schultz, *Journal of Molecular Structure: THEOCHEM*, 2003, **622**, 39-51.
46. L. P. Ammann, *Journal of the American Statistical Association*, 1993, **88**, 505-514.
47. B. Schölkopf, A. Smola and K.-R. Müller, *Neural computation*, 1998, **10**, 1299-1319.
48. F. Klepsch, P. Vasanthanathan and G. F. Ecker, *Journal of chemical information and modeling*, 2014, **54**, 218-229.
49. I. T. Jolliffe, *Applied statistics*, 1972, 160-173.
50. I. Jolliffe, *Applied Statistics*, 1973, 21-31.
51. F. Luan, R. Zhang, C. Zhao, X. Yao, M. Liu, Z. Hu and B. Fan, *Chemical research in toxicology*, 2005, **18**, 198-203.
52. Q. Zang, D. A. Keire, R. D. Wood, L. F. Buhse, C. Moore, M. Nasr, A. Al-Hakim, M. L. Trehy and W. J. Welsh, *Journal of pharmaceutical and biomedical analysis*, 2011, **54**, 1020-1029.
53. K. Roy and I. Mitra, *Combinatorial chemistry & high throughput screening*, 2011, **14**, 450-474.
54. T. Fawcett, *Pattern recognition letters*, 2006, **27**, 861-874.
55. A. Afantitis, G. Melagraki, H. Sarimveis, P. A. Koutentis, O. Igglessi-Markopoulou and G. Kollias, *Molecular diversity*, 2010, **14**, 225-235.
56. Y. Hung and Y. Liao, *Inf Technol J*, 2008, **7**, 890-896.
57. S. Wold and M. Sjöström, 1977.
58. C. B. R. d. Santos, C. C. Lobato, M. A. C. de Sousa, W. J. d. C. Macêdo and J. C. T. Carvalho, *Reviews in Theoretical Science*, 2014, **2**, 91-115.
59. P. Mazzatorta, E. Benfenati, P. Lorenzini and M. Vighi, *Journal of chemical information and computer sciences*, 2004, **44**, 105-112.
60. R. D. Clark, P. Wolohan, E. E. Hodgkin, J. H. Kelly and N. L. Sussman, *Journal of Molecular Graphics and Modelling*, 2004, **22**, 487-497.
61. J. D. Cunha, M. L. Lavaggi, M. I. Abasolo, H. Cerecetto and M. González, *Chemical biology & drug design*, 2011, **78**, 960-968.
62. V. Martinez-Merino and H. Cerecetto, *Bioorganic & medicinal chemistry*, 2001, **9**, 1025-1030.
63. T. Puzyn, J. Leszczynski and M. T. Cronin, *challenges and advances in computational chemistry and physics*, 2010, **8**.
64. H. Mager and A. Barth, *Die Pharmazie*, 1978, **34**, 557-559.
65. V. Consonni, R. Todeschini and M. Pavan, *Journal of chemical information and computer sciences*, 2002, **42**, 682-692.
66. R. Leardi, *Journal of Chromatography A*, 2007, **1158**, 226-233.
67. S.-S. So and M. Karplus, *Journal of Medicinal Chemistry*, 1996, **39**, 1521-1530.

68. B. T. Luke, *Journal of Chemical Information and Computer Sciences*, 1994, **34**, 1279-1287.
69. R. V. Devi, S. S. Sathya and M. S. Coumar, *Applied Soft Computing*, 2015, **27**, 543-552.
70. T. C. Le and D. A. Winkler, *ChemMedChem*, 2015, **10**, 1296-1300.
71. D. Rogers and A. J. Hopfinger, *Journal of Chemical Information and Computer Sciences*, 1994, **34**, 854-866.
72. D. L. Selwood, D. J. Livingstone, J. C. Comley, A. B. O'Dowd, A. T. Hudson, P. Jackson, K. S. Jandu, V. S. Rose and J. N. Stables, *Journal of medicinal chemistry*, 1990, **33**, 136-142.
73. L. Maccari, M. Magnani, G. Strappaghetti, F. Corelli, M. Botta and F. Manetti, *Journal of chemical information and modeling*, 2006, **46**, 1466-1478.
74. V. Srivastav and M. Tiwari, *Arabian Journal of Chemistry*, 2013.
75. A. E. Hoerl and R. W. Kennard, *Technometrics*, 1970, **12**, 55-67.
76. J. B. Ghasemi, A. Abdolmaleki and N. Mandoumi, *Journal of hazardous materials*, 2009, **161**, 74-80.
77. F. H. Quina, E. O. Alonso and J. P. Farah, *The Journal of Physical Chemistry*, 1995, **99**, 11708-11714.
78. T. Lestander, *Multivariate NIR studies of seed-water interaction in Scots pine seeds (Pinus sylvestris L.)*, 2003.
79. P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, Siam, 1998.
80. A. Tikhonov, 1964.
81. J. H. Kalivas, J. B. Forrester and H. A. Seipel, *Journal of computer-aided molecular design*, 2004, **18**, 537-547.
82. B. E. Mattioni and P. C. Jurs, *Journal of chemical information and computer sciences*, 2002, **42**, 232-240.
83. K. Roy and G. Ghosh, *QSAR & Combinatorial Science*, 2004, **23**, 526-535.
84. A. Syahputra, M. Mudasir, N. Nuryono, A. Aziz and I. Tahir, *Indonesian Journal of Chemistry*, 2014, **14**, 94-101.
85. X. L. Li, X. P. Li and Y. Li, *Advanced Materials Research*, 2012, **599**, 151-154.
86. S. Wold, M. Sjöström and L. Eriksson, *Chemometrics and intelligent laboratory systems*, 2001, **58**, 109-130.
87. A. Lorber, L. E. Wangen and B. R. Kowalski, *Journal of Chemometrics*, 1987, **1**, 19-31.
88. P. Geladi and B. R. Kowalski, *Analytica chimica acta*, 1986, **185**, 1-17.
89. A. Höskuldsson, *Journal of chemometrics*, 1988, **2**, 211-228.
90. S. Wold, H. Martens and H. Wold, in *Matrix pencils*, Springer, 1983, pp. 286-293.
91. S. Wold, A. Ruhe, H. Wold and I. Dunn, WJ, *SIAM Journal on Scientific and Statistical Computing*, 1984, **5**, 735-743.
92. N. Kansal, O. Silakari and M. Ravikumar, *Letters in Drug Design & Discovery*, 2008, **5**, 437-448.
93. Z. G. Li, K. X. Chen, H. Y. Xie and J. R. Gao, *QSAR & Combinatorial Science*, 2009, **28**, 89-97.
94. N. Priolo, C. M. Arribére, N. Caffini, S. Barberis, R. N. Vázquez and J. M. Luco, *Journal of Molecular Catalysis B: Enzymatic*, 2001, **15**, 177-189.
95. Y. Shan-Bin, X. Zhi-Ning, S. Mao, M. Hu, L. Feng-Lin, Z. Mei, W. Yu-Qian and L. Zhi-Liang, *CHEMICAL JOURNAL OF CHINESE UNIVERSITIES-CHINESE*, 2008, **29**, 2213-2217.
96. C. Rücker, G. Rücker and M. Meringer, *Journal of chemical information and modeling*, 2007, **47**, 2345-2357.
97. R. D. Clark and P. C. Fox, *Journal of computer-aided molecular design*, 2004, **18**, 563-576.
98. M. Sjöstrom, S. Wold and B. Soderstrom, *Elsevier Science Publishers BV, North-Holland*, 1986, **136**, 203.
99. S. A. Greibach, *Lecture Notes in Computer Science*, 1975.
100. S. Wold, *Chemometrics and Intelligent Laboratory Systems*, 1992, **14**, 71-84.
101. S. Wold, N. Kettaneh-Wold and B. Skagerberg, *Chemometrics and Intelligent Laboratory Systems*, 1989, **7**, 53-65.
102. G. Baffi, E. Martin and A. Morris, *Computers & Chemical Engineering*, 1999, **23**, 1293-1307.
103. R. Rosipal and L. J. Trejo, *The Journal of Machine Learning Research*, 2002, **2**, 97-123.
104. R. Rosipal, L. J. Trejo and B. Matthews, 2003.
105. N. Aronszajn, *Transactions of the American mathematical society*, 1950, 337-404.
106. B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.
107. K. Bennett and M. Embrechts, *Nato Science Series sub series III computer and systems sciences*, 2003, **190**, 227-250.
108. Y. An, W. Sherman and S. L. Dixon, *Journal of chemical information and modeling*, 2013, **53**, 2312-2321.
109. K.-l. TANG, T.-h. LI and K. CHEN, *Chemical Research in Chinese Universities*, 2008, **24**, 541-545.
110. K. Hasegawa, M. Arakawa and K. Funatsu, *Chemometrics and Intelligent Laboratory Systems*, 2000, **50**, 253-261.
111. M. Goodarzi and M. P. de Freitas, *Molecular Simulation*, 2010, **36**, 267-272.
112. M. Bacilieri, A. Ciancetta, S. Paoletta, S. Federico, S. Cosconati, B. Cacciari, S. Taliani, F. Da Settimo, E. Novellino and K. N. Klotz, *Journal of chemical information and modeling*, 2013, **53**, 1620-1637.
113. S. L. Dixon, A. M. Smondyrev, E. H. Knoll, S. N. Rao, D. E. Shaw and R. A. Friesner, *Journal of computer-aided molecular design*, 2006, **20**, 647-671.
114. P. J. Huber, *The annals of Statistics*, 1985, 435-475.
115. J. H. Friedman and W. Stuetzle, *Journal of the American statistical Association*, 1981, **76**, 817-823.
116. H. Du, J. Wang, J. Watzl, X. Zhang and Z. Hu, *Chemometrics and Intelligent Laboratory Systems*, 2008, **93**, 160-166.
117. Y. Yuan, R. Zhang, R. Hu and X. Ruan, *European journal of medicinal chemistry*, 2009, **44**, 25-34.
118. S. B. Gunturi, K. Archana, A. Khandelwal and R. Narayanan, *QSAR & Combinatorial Science*, 2008, **27**, 1305-1317.
119. W. Zheng and A. Tropsha, *Journal of chemical information and computer sciences*, 2000, **40**, 185-194.
120. G. Alexander and T. Alexander, *J Mol Graph Model*, 2002, **20**, 269-276.
121. Y. Xue, C. W. Yap, L. Z. Sun, Z. W. Cao, J. Wang and Y. Z. Chen, *Journal of chemical information and computer sciences*, 2004, **44**, 1497-1505.
122. F. R. Burden, *Journal of chemical information and computer sciences*, 2001, **41**, 830-835.

123. O. Obrezanova, G. Csányi, J. M. Gola and M. D. Segall, *Journal of chemical information and modeling*, 2007, **47**, 1847-1857.
124. P. Constans and J. D. Hirst, *Journal of chemical information and computer sciences*, 2000, **40**, 452-459.
125. J. D. Hirst, T. J. McNeany, T. Howe and L. Whitehead, *Bioorganic & medicinal chemistry*, 2002, **10**, 1037-1041.
126. G. Harper, J. Bradshaw, J. C. Gittins, D. V. Green and A. R. Leach, *Journal of Chemical Information and Computer Sciences*, 2001, **41**, 1295-1300.
127. B. Chen, R. F. Harrison, G. Papadatos, P. Willett, D. J. Wood, X. Q. Lewell, P. Greenidge and N. Stiefl, *Journal of computer-aided molecular design*, 2007, **21**, 53-62.
128. K.-T. Fang, H. Yin and Y.-Z. Liang, *Journal of chemical information and computer sciences*, 2004, **44**, 2106-2113.
129. L. A. Berrueta, R. M. Alonso-Salces and K. Héberger, *Journal of Chromatography A*, 2007, **1158**, 196-214.
130. N. Baurin, J.-C. Mozziconacci, E. Arnoult, P. Chavatte, C. Marot and L. Morin-Allory, *Journal of chemical information and computer sciences*, 2004, **44**, 276-285.
131. P. Itskowitz and A. Tropsha, *Journal of chemical information and modeling*, 2005, **45**, 777-785.
132. R. Khashan, W. Zheng and A. Tropsha, *Molecular Informatics*, 2014, **33**, 201-215.
133. V. E. Kuz'min, A. G. Artemenko, E. N. Muratov, I. L. Volineckaya, V. A. Makarov, O. B. Riabova, P. Wutzler and M. Schmidtke, *Journal of medicinal chemistry*, 2007, **50**, 4205-4213.
134. V. Kuz'min, A. G. Artemenko and E. N. Muratov, *Journal of computer-aided molecular design*, 2008, **22**, 403-421.
135. M. Hagan, H. Demuth and M. Beale, *Journal*, 1996.
136. F. Despagne and D. L. Massart, *Analyst*, 1998, **123**, 157R-178R.
137. C. M. Bishop, *Journal*, 2006.
138. G. Schneider and P. Wrede, *Progress in biophysics and molecular biology*, 1998, **70**, 175-222.
139. D. W. Salt, N. Yildiz, D. J. Livingstone and C. J. Tinsley, *Pesticide science*, 1992, **36**, 161-170.
140. F. Rosenblatt, *Psychological review*, 1958, **65**, 386.
141. T. B. Blank and S. D. Brown, *Analytical chemistry*, 1993, **65**, 3081-3089.
142. M. L. Minsky and S. A. Papert, *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*, MIT press Boston, MA:, 1987.
143. S. Kaski, J. Kangas and T. Kohonen, *Neural computing surveys*, 1998, **1**, 1-176.
144. T. Kohonen, *Proceedings of the IEEE*, 1990, **78**, 1464-1480.
145. P. Selzer and P. Ertl, *QSAR & Combinatorial Science*, 2005, **24**, 270-276.
146. S. Lee and R. M. Kil, *Neural Networks*, 1991, **4**, 207-224.
147. J. Gasteiger and J. Zupan, *Angewandte Chemie International Edition in English*, 1993, **32**, 503-527.
148. R. Hecht-Nielsen, in *Neural computers*, Springer, 1989, pp. 445-453.
149. D. G. Stork, *Synapse Connection*, 1988, **1**, 9-11.
150. J. Park and I. W. Sandberg, *Neural computation*, 1993, **5**, 305-316.
151. A. Levit, T. Beuming, G. Krilov, W. Sherman and M. Y. Niv, *Journal of chemical information and modeling*, 2013, **54**, 184-194.
152. K. Kaiser, S. Niculescu and G. Schuurmann, *Water Quality Research Journal of Canada*, 1997, **32**, 637-657.
153. B. Hemmateenejad, M. Akhond, R. Miri and M. Shamsipur, *Journal of chemical information and computer sciences*, 2003, **43**, 1328-1334.
154. D. González-Arjona, G. López-Pérez and A. Gustavo González, *Talanta*, 2002, **56**, 79-90.
155. M. Goodarzi, E. V. Ortiz, L. d. S. Coelho and P. R. Duchowicz, *Atmospheric Environment*, 2010, **44**, 3179-3186.
156. J. C. Patra and O. Singh, *Journal of computational chemistry*, 2009, **30**, 2494-2508.
157. G. Ramírez-Galicia, R. Garduño-Juárez, B. Hemmateenejad, O. Deeb, M. Deciga-Campos and J. C. Moctezuma-Eugenio, *Chemical biology & drug design*, 2007, **70**, 53-64.
158. Y. Akhlaghi and M. Kompany-Zareh, *Journal of chemometrics*, 2006, **20**, 1-12.
159. F. R. Burden and D. A. Winkler, *Journal of medicinal chemistry*, 1999, **42**, 3183-3187.
160. D. A. Winkler and F. R. Burden, *Molecular Simulation*, 2000, **24**, 243-258.
161. D. A. Winkler and F. R. Burden, *Journal of Molecular Graphics and Modelling*, 2004, **22**, 499-505.
162. F. e. R. Burden and D. e. A. Winkler, *QSAR & Combinatorial Science*, 2009, **28**, 645-653.
163. V. C. Epa, F. R. Burden, C. Tassa, R. Weissleder, S. Shaw and D. A. Winkler, *Nano letters*, 2012, **12**, 5808-5812.
164. V. Epa, A. Hook, C.-y. Chang, J. Yang, R. Langer, D. Anderson, P. Williams, M. Davies, M. Alexander and D. Winkler, *Advanced Functional Materials*, 2014, **24**, 2085-2093.
165. D. A. Winkler and F. R. Burden, *Molecular BioSystems*, 2012, **8**, 913-920.
166. H. Autefage, E. Gentleman, E. Littmann, M. A. Hedegaard, T. Von Erlach, M. O'Donnell, F. R. Burden, D. A. Winkler and M. M. Stevens, *Proceedings of the National Academy of Sciences*, 2015, **112**, 4280-4285.
167. F. R. Burden and D. A. Winkler, *QSAR & Combinatorial Science*, 2009, **28**, 1092-1097.
168. V. N. Vapnik and V. Vapnik, *Statistical learning theory*, Wiley New York, 1998.
169. C. Cortes and V. Vapnik, *Machine learning*, 1995, **20**, 273-297.
170. J. A. Suykens and J. Vandewalle, *Neural processing letters*, 1999, **9**, 293-300.
171. J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens and T. Van Gestel, *Least squares support vector machines*, World Scientific, 2002.
172. H. Briem and J. Günther, *Chembiochem*, 2005, **6**, 558-566.
173. R. N. Jorissen and M. K. Gilson, *Journal of chemical information and modeling*, 2005, **45**, 549-561.
174. T. Sato, H. Yuki, D. Takaya, S. Sasaki, A. Tanaka and T. Honma, *Journal of chemical information and modeling*, 2012, **52**, 1015-1026.
175. M. E. Tipping, *The journal of machine learning research*, 2001, **1**, 211-244.
176. F. R. Burden and D. A. Winkler, *Journal of chemical information and modeling*, 2015, **55**, 1529-1534.
177. J. R. Quinlan, *Machine learning*, 1986, **1**, 81-106.
178. S. B. Gelfand, C. Ravishankar and E. J. Delp, 1989.
179. L. Breiman, J. Friedman, C. J. Stone and R. A. Olshen, *Classification and regression trees*, CRC press, 1984.

180. R. Put, C. Perrin, F. Questier, D. Coomans, D. Massart and Y. Vander Heyden, *Journal of Chromatography A*, 2003, **988**, 261-276.
181. E. R. Ziegel, *Technometrics*, 2000, **42**, 218-219.
182. P. Tino, I. T. Nabney, B. S. Williams, J. Lösel and Y. Sun, *Journal of chemical information and computer sciences*, 2004, **44**, 1647-1653.
183. V. Svetnik, T. Wang, C. Tong, A. Liaw, R. P. Sheridan and Q. Song, *Journal of Chemical Information and Modeling*, 2005, **45**, 786-799.
184. M. Daszykowski, B. Walczak, Q.-S. Xu, F. Daeyaert, M. R. de Jonge, J. Heeres, L. M. Koymans, P. J. Lewi, H. M. Vinkers and P. Janssen, *Journal of chemical information and computer sciences*, 2004, **44**, 716-726.
185. R. K. DeLisle and S. L. Dixon, *Journal of chemical information and computer sciences*, 2004, **44**, 862-870.
186. J. P. Bai, A. Utis, G. Crippen, H.-D. He, V. Fischer, R. Tullman, H.-Q. Yin, C.-P. Hsu, L. Jiang and K.-K. Hwang, *Journal of chemical information and computer sciences*, 2004, **44**, 2061-2069.
187. W. Tong, H. Hong, H. Fang, Q. Xie and R. Perkins, *Journal of Chemical Information and Computer Sciences*, 2003, **43**, 525-531.
188. L. Breiman, *Machine learning*, 2001, **45**, 5-32.
189. P. G. Polishchuk, E. N. Muratov, A. G. Artemenko, O. G. Kolumbin, N. N. Muratov and V. E. Kuz'min, *Journal of chemical information and modeling*, 2009, **49**, 2481-2488.
190. J. B. Ghasemi, N. Meftahi, S. Pirhadi and H. Tavakoli, *Journal of Chemometrics*, 2013, **27**, 287-296.
191. L. Breiman, *Machine learning*, 1996, **24**, 123-140.
192. D. M. Hawkins, S. C. Basak and D. Mills, *Journal of chemical information and computer sciences*, 2003, **43**, 579-586.
193. L. Breiman, *Out-of-bag estimation*, Citeseer, 1996.
194. J. H. Friedman, *Annals of Statistics*, 2001, 1189-1232.
195. Y. Freund and R. E. Schapire, 1995.
196. J. H. Friedman, *Computational Statistics & Data Analysis*, 2002, **38**, 367-378.
197. L. Breiman, *The annals of statistics*, 1998, **26**, 801-849.
198. C. W. Codrington, 2001.
199. D. Meyer, F. Leisch and K. Hornik, *Neurocomputing*, 2003, **55**, 169-186.
200. D. Hawkins and B. Musser, *Computing Science and Statistics*, 1998, 534-542.
201. A. M. van Rhee, *Journal of chemical information and computer sciences*, 2003, **43**, 941-948.
202. R. E. Schapire, *Machine learning*, 1990, **5**, 197-227.
203. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu and S. Y. Philip, *Knowledge and Information Systems*, 2008, **14**, 1-37.
204. J. Friedman, T. Hastie and R. Tibshirani, *The annals of statistics*, 2000, **28**, 337-407.
205. C. Rudin, I. Daubechies and R. E. Schapire, *The Journal of Machine Learning Research*, 2004, **5**, 1557-1595.
206. D. K. Agrafiotis, W. Cedeno and V. S. Lobanov, *Journal of chemical information and computer sciences*, 2002, **42**, 903-911.
207. K. P. Singh and S. Gupta, *RSC Advances*, 2014, **4**, 13215-13230.
208. R. C. Braga, V. M. Alves, M. Silva, E. Muratov, D. Fourches, A. Tropsha and C. H. Andrade, *Current topics in medicinal chemistry*, 2014.
209. K.-R. Müller, G. Rätsch, S. Sonnenburg, S. Mika, M. Grimm and N. Heinrich, *Journal of chemical information and modeling*, 2005, **45**, 249-253.
210. Q. Zang, D. M. Rotroff and R. S. Judson, *Journal of chemical information and modeling*, 2013, **53**, 3244-3261.
211. R. D. Cramer, D. E. Patterson and J. D. Bunce, *Journal of the American Chemical Society*, 1988, **110**, 5959-5967.
212. S. Pirhadi, F. Shiri and J. B. Ghasemi, *Journal of the Iranian Chemical Society*, 2014, **11**, 1329-1336.
213. J. B. Ghasemi and F. Shiri, *Medicinal Chemistry Research*, 2012, **21**, 2788-2806.
214. F. Shiri, S. Pirhadi and J. B. Ghasemi, *Saudi Pharmaceutical Journal*, 2015.
215. C. C. Melo-Filho, R. C. Braga and C. H. Andrade, *Curr. Comput. Aided Drug Des*, 2014, **10**, 148-159.
216. J. Verma, V. M. Khedkar and E. C. Coutinho, *Current topics in medicinal chemistry*, 2010, **10**, 95-115.
217. A. Hopfinger, S. Wang, J. S. Tokarski, B. Jin, M. Albuquerque, P. J. Madhav and C. Duraiswami, *Journal of the American Chemical Society*, 1997, **119**, 10509-10524.
218. A. Vedani, H. Briem, M. Dobler, H. Dollinger and D. R. McMasters, *Journal of medicinal chemistry*, 2000, **43**, 4416-4427.
219. V. Lukacova and S. Balaz, *Journal of chemical information and computer sciences*, 2003, **43**, 2093-2105.
220. A. Vedani and M. Dobler, *Journal of medicinal chemistry*, 2002, **45**, 2139-2149.
221. M. G Damale, S. N Harke, F. A Kalam Khan, D. B Shinde and J. N Sangshetti, *Mini reviews in medicinal chemistry*, 2014, **14**, 35-55.
222. A. Vedani, M. Dobler and M. A. Lill, *Journal of medicinal chemistry*, 2005, **48**, 3700-3703.

Figure captions

Figure 1. Hierarchical clustering depicted in a dendrogram, with two selected clusters.

Figure 2. A line of best fit to the cantered objects in space is the first principal component. The distance to the line from each point is minimized in a least-squares fashion.

Figure 3. A projection of simulated data onto the two-dimensional space spanned by original variables a) without outliers b) with outliers

Figure 4. (A) Score plot of first two principal components analysis shown. Inhibitors are displayed in green circles and non-inhibitors are demonstrated in red dots. (B) Loading plot of descriptors used for PCA analysis based on PC1 and PC2.

Figure 5. Applicability domain experiment using "Euclidean distances (ED)", approach Compounds shown as follows: Training compounds: Gray dots, FDA compounds: red square, Test compounds: Black cross

Figure 6. Distribution of the samples in two classes on a transformed axis, L.

Figure 7. Class modeling by using SIMCA approach.

Figure 8. CA IV harmonious plots using 63 descriptors. (a) $\left\|\hat{b}\right\|_2$ against RMSEC for PCR (□), PLS (○), and RR (Δ). Filled symbols denote optimal models with a ridge value of 750 and 6 and 8 PLS and PCR factors, respectively. Ridge values vary from 45 in the upper left corner to 7050 in the lower right corner. PLS and PCR models varying from 9 and 16 factors in the upper left corner, respectively, to 5 and 6 factors in the lower right corner, respectively. (b) $\left\|\hat{b}\right\|_2$ against RMSEV for PCR (□), PLS (○), and RR (Δ). Filled symbols denote optimal models as in (a). Ridge values and PLS models vary as in (a). PCR models varying from 15 factors in the upper centre to 6 factors in the lower right corner.

Figure9. Classification rule provided by kNN approach with different k values. Here the unknown sample is the star.

Figure 10. Feed-forward NN training: a, forward pass; b, error

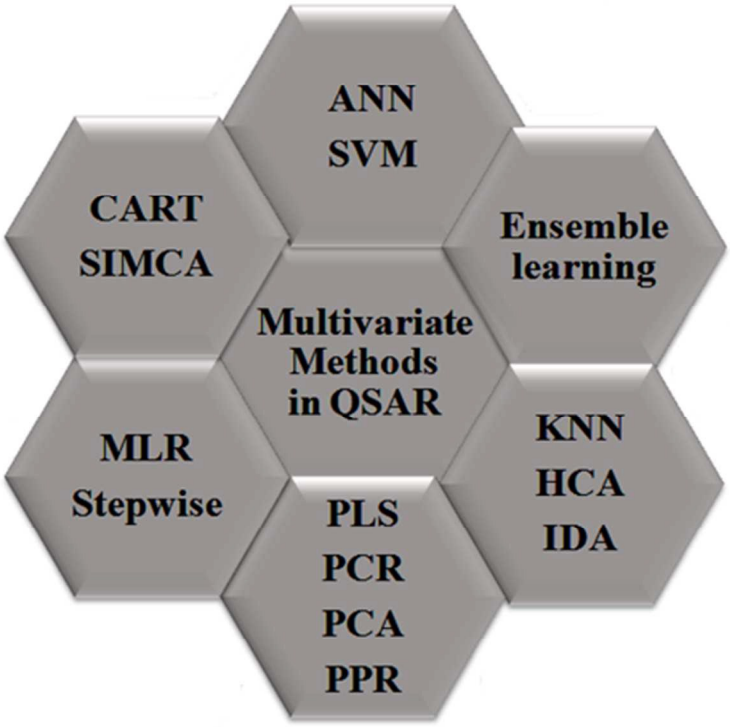
Figure 11. A self-organization map network

Figure 12. Support vectors and decision boundary

Figure 13. Non-linear classification using kernel function

Figure 14. Scheme of SVM learning based on 3D similarity profiles.

Figure 15. Transformation of a linear classifier into a nonlinear classifier using a multiple-planar classifier.



Graphical Abstract

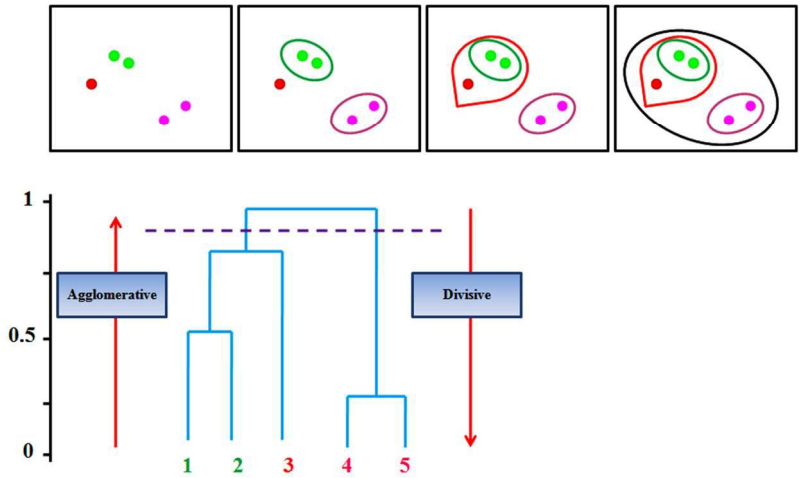


Figure 1

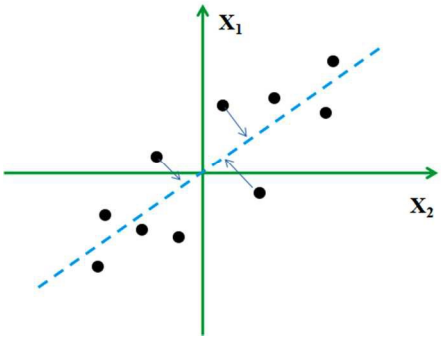


Figure 2

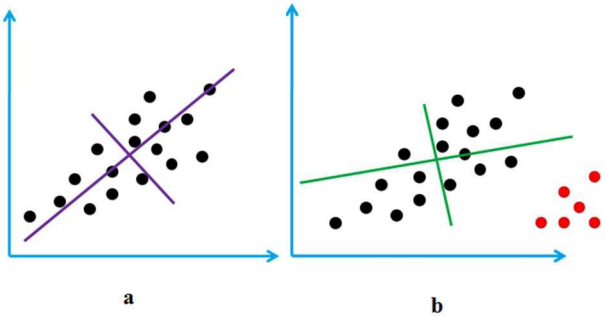


Figure 3

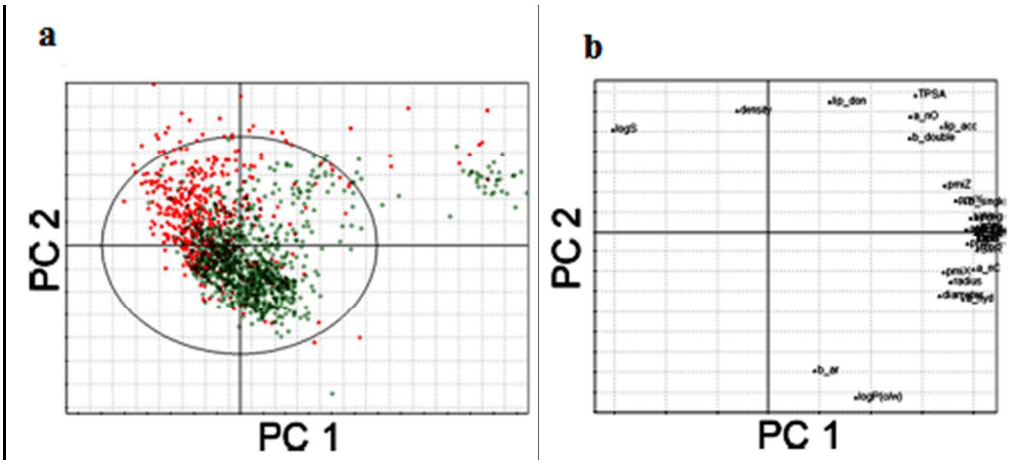


Figure 4

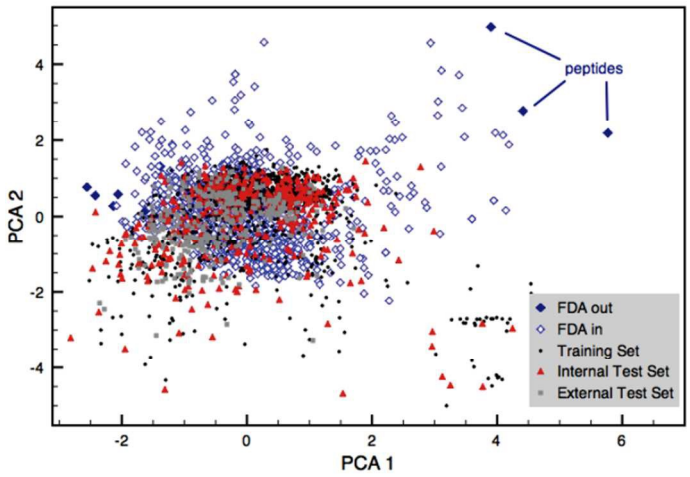


Figure 5

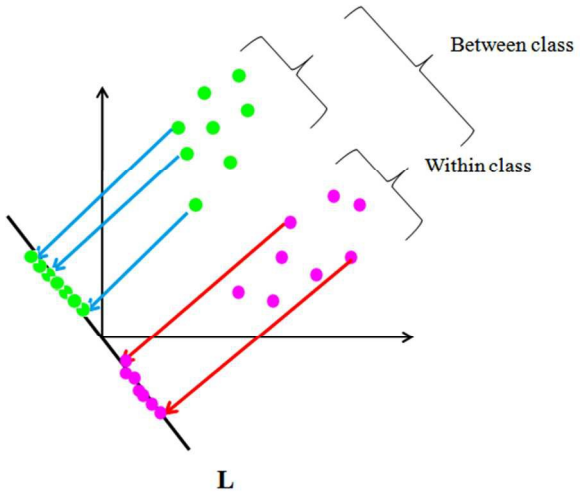


Figure 6

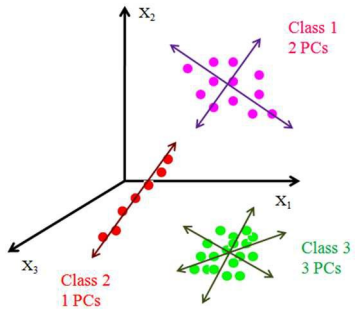


Figure 7

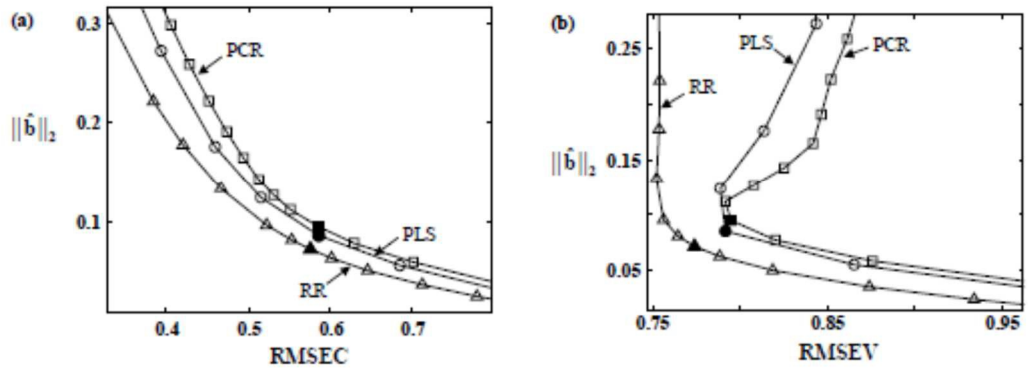


Figure 8

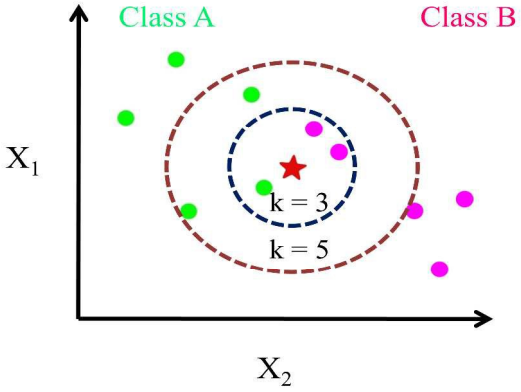


Figure 9

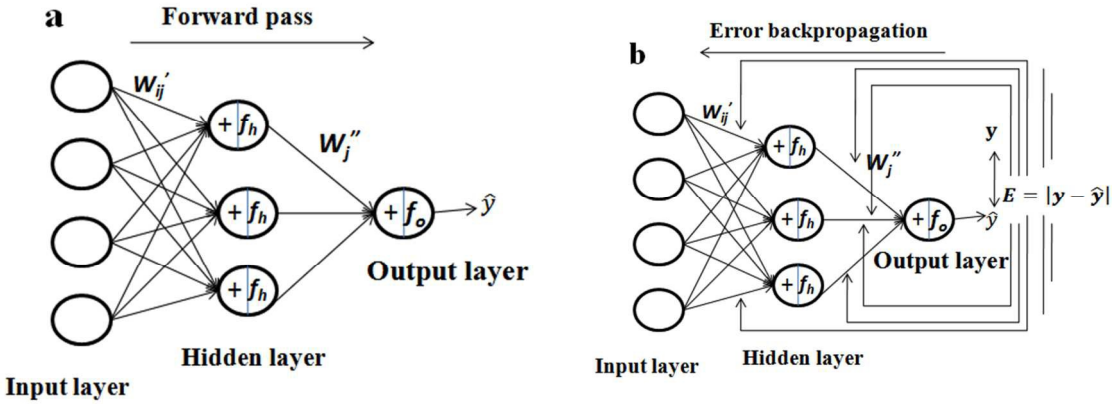


Figure 10

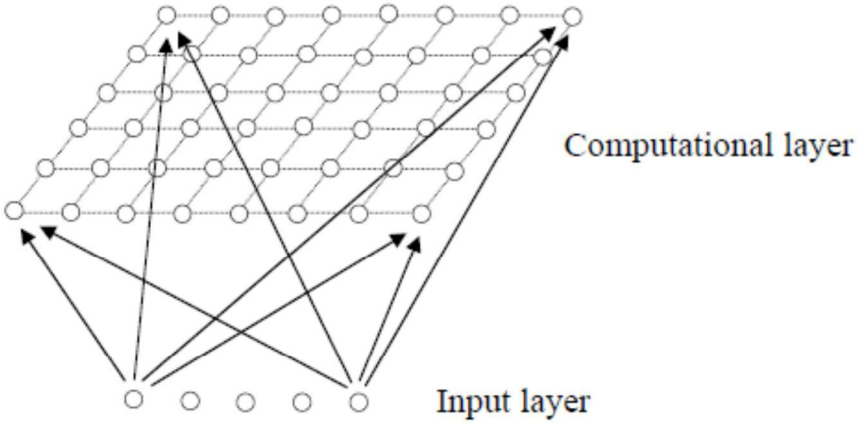


Figure 11

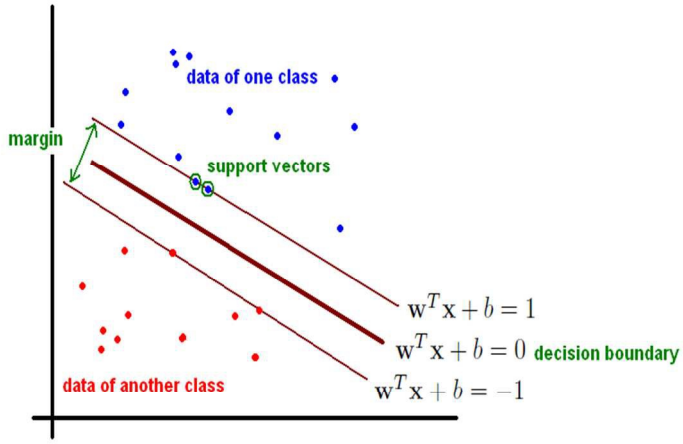


Figure 12

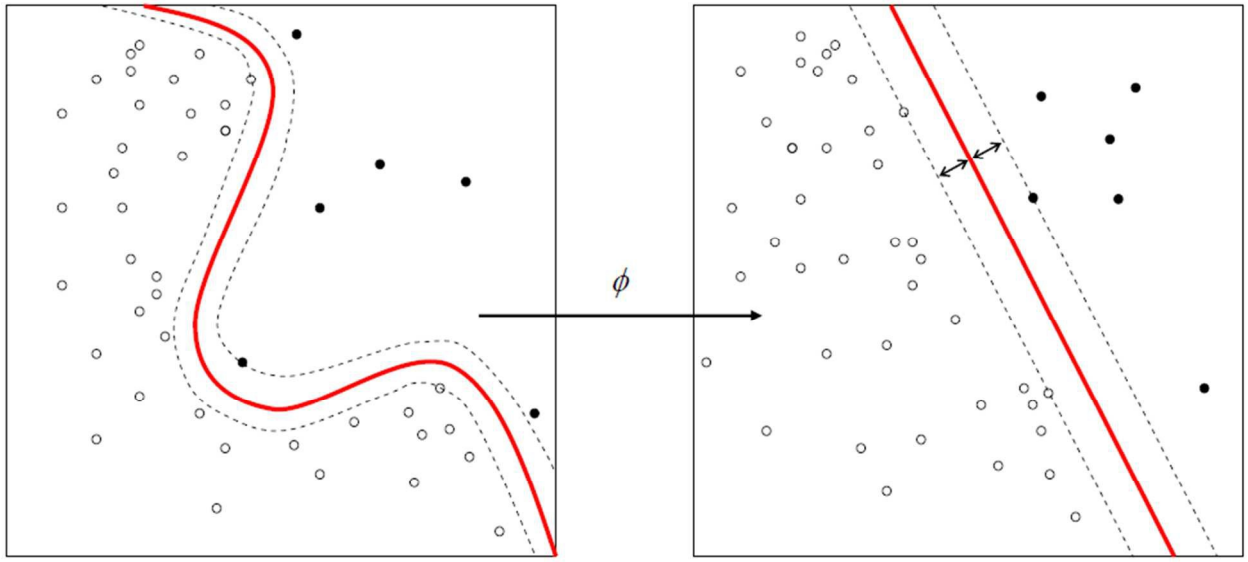


Figure 13

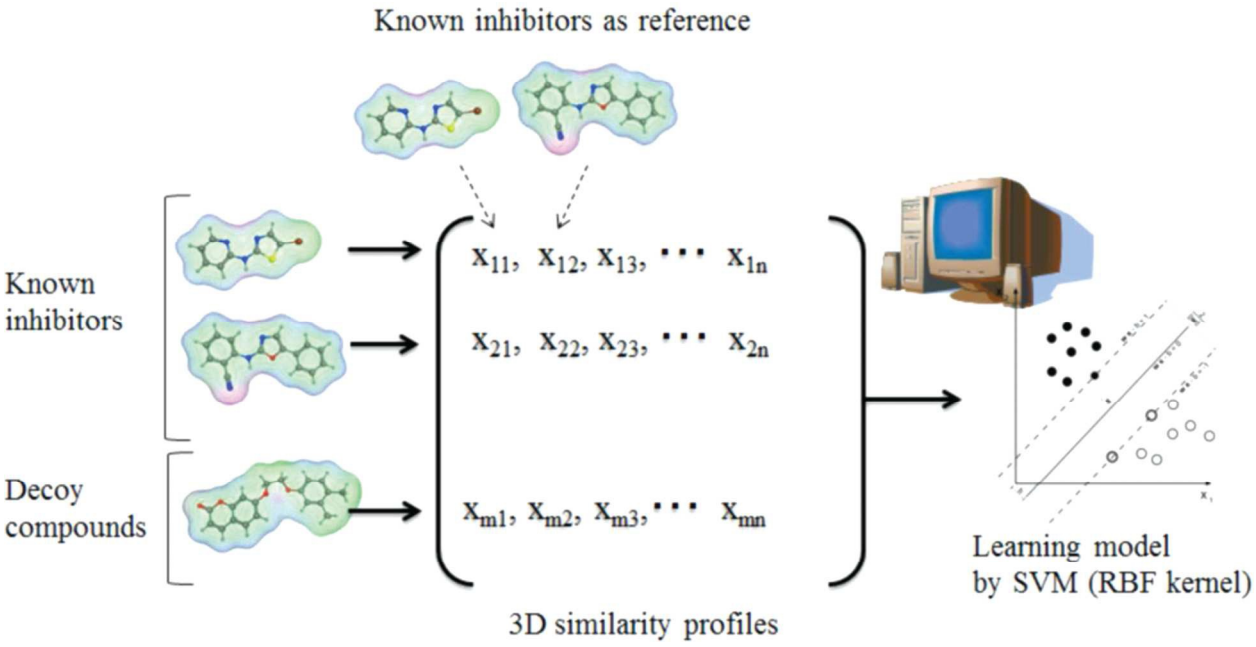


Figure 14

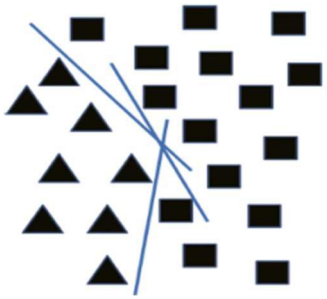


Figure 15