Section III

Cheminformatics - Advanced: *Data Analysis, Classification Methods, Quantitative Structure-Activity Relationships (QSAR)*

Dan C. Fara and Tudor I. Oprea

1. Methods for data analysis and classification

Generally, *data analysis* is the process of looking at and summarizing data with the intent to extract useful information and develop conclusions. Data analysis is closely related to *data mining*, but data mining tends to focus on larger data sets, with less emphasis on making inference, and often uses data that was originally collected for a different purpose. Some people divide data analysis into exploratory data analysis (EDA) and confirmatory data analysis (CDA), where the EDA focuses on discovering new features in the data, and CDA on confirming or falsifying existing hypotheses.

Analysis of the data and, in principal, establishment of a relationship between chemical compounds and their activity / properties can be done by (i) a *statistical or pattern-recognition* method, and (ii) a *learning* method.

1.1 Statistical and/or pattern recognition methods

1.1.1 Multivariate data

Multivariate data refer to objects, features and properties. An *object* is any real or abstract item, in our case a chemical structure, which is characterized by a set of features. A *feature* is a numerical variable (measured or computed data) such as molecular descriptor. Application of statistical methods requires a reasonable number of objects and features, described by an (n x m) matrix X, containing a row for each of the n objects, and a column for each of the m features. In a geometrical interpretation each object is considered as a point (or a vector), with the features used as coordinates (or vector components) in an m-dimensional *feature space*. The applied hypothesis assumes that objects close together have attributes in common (for instance a similar value for the biological activity or physicochemical property or the same substance class). Additionally, a y-column or a Y-matrix might contain data for one or several properties of the objects. A *property* is an interesting attribute of the objects difficult or not directly measurable, i.e. biological activity. The goal of multivariate analysis is to model and predict the property y from the features x.

In other words, the aim of multivariate data analysis is to find a formal combination (a mathematical function or a more general algorithm) of the features by defining a new, so-called *latent variable* (*score*) which reveals problem-relevant information. The coefficients of the features are called *loadings* and the values of the latent variable are called *scores*. The loadings describe the individual contributions of the feature to the latent variables. Depending on the aim of data analysis different mathematical criteria are applied:

- For principal components the criterion is maximum variance of the score, providing an optimum representation of the Euclidean distances between the objects;
- For a discriminant variable an optimum separation of object classes is required;

- Latent variables with maximum correlation to a property of interest, **y**, are used in multivariate calibration.

Principal Component Analysis (PCA)[1] is the most frequently used method for computing linear latent variables. Latent variables from PCA optimally represent the distances between objects in a multivariate data set X. The direction that best describes the relative distances between the objects is the latent variable which has *maximum variance*. This direction is called the *first principal component* (PC1). The second principal component (PC2) is **orthogonal** to PC1 and again has the maximum possible variance. Further principal components can be determined by continuing this concept. Determination of all principal components corresponds to a rotation of the coordinate system of the feature space – see Figure 1.

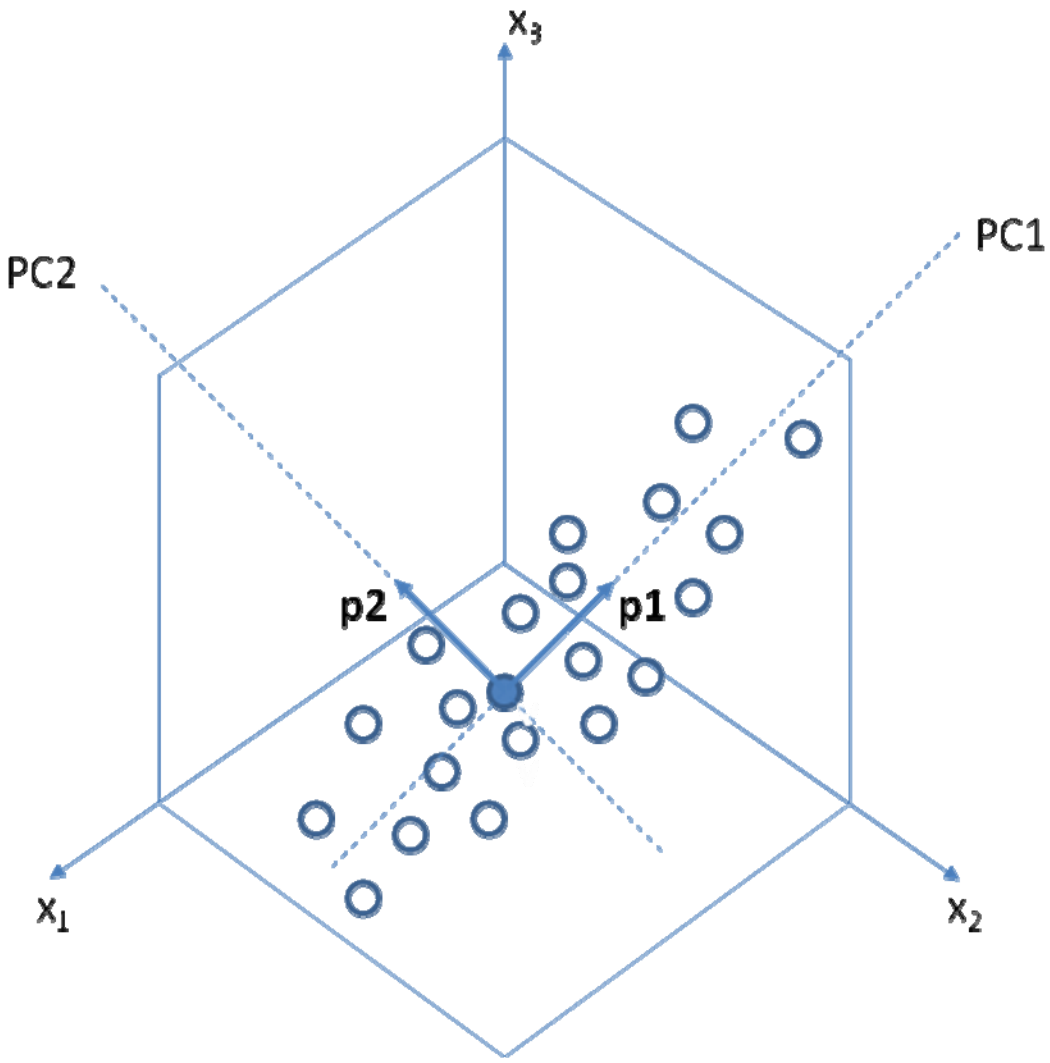

Figure 1. Principal component analysis (PCA). The first principal component, PC1, is the latent variable with maximum variance (loading vector $\mathbf{p}_1$). PC2 is **orthogonal** to PC1 and again has maximum possible variance (loading vector $\mathbf{p}_2$).

Partial Least Squares (PLS) Regression is a method of relating two data matrices, X and Y[1]. The criterion mostly applied in PLS is maximum covariance between latent variables and *y*. This criterion combines high variance of X (responsible for robustness) and high correlation with the property of interest. PLS is a linear method (although non-linear versions exist) and therefore the final latent variable that is used to predict property

y, is a linear combination of the features. X is a mean centered matrix with *n* rows (objects) and *m* features (predictor variables); Y is a mean centered matrix with *n* rows and *k* properties (response variables). The PLS algorithm produces – similar to PCA but with a different aim and with different methods – a score matrix T that models X and a score matrix U that models Y; both have *n* rows and *a* columns with *a* being the number of calculated PLS components (intermediate latent variables that are regressed with *y*). T and U model X and Y, respectively, but also include information about linear relationships between *x*- and *y*-data. The *x*-scores are alinear combination of the *x*-variables and can be regarded as good summaries of X. The -variables and can be regarded as good summaries of X. The *y*-scores are a linear combination of the *y*-variables and can be regarded as good summaries of Y. *x*- and *y*-scores are connected by the inner (linear) relationships as described in reference[1]. Characteristic to the PLS algorithm is the stepwise calculation of the components. After a component is computed the residual matrices for X and Y are determined. The next PLS component is calculated from the residual matrices and therefore its parameters (scores, loadings, weights) do not relate to X but to the residual matrices. When a PLS model has been created, it can be used to predict *y*-properties and *y*-scores from new *x*-data. The model is a linear combination of the predictors.

1.1.2 Multivariate exploratory data analysis

If multivariate data only consist of a feature matrix X, data analysis is often called *exploratory* or *unsupervised*, because no additional information about the objects, i.e. compounds, is used or available to guide the work. The purposes of data interpretation are manifold: (i) obtaining a better insight into the data structure, (ii) searching for groups with similar objects (*cluster analysis*), (iii) searching for outliers, or (iv) searching for relevant features, i.e. descriptors. Exploratory data analysis is usually the first step and represents an objective inspection of data even if the final goal is *calibration* or *classification*[2]. The most used techniques are principal component analysis (PCA), cluster analysis with dendrograms, i.e. agglomerative hierarchical cluster, and *Kohonen mapping* (a non-linear method)[3,4].

PCA Is usually the first method applied to a data matrix X. Most powerful are scatter plots with a point for each object and the coordinates of the points given by the scores of the first and second principal components. This visualization of the data corresponds to a linear projection of the *m*-dimensional features space on to a plane by preserving a maximum of the relative distances (similarities) between the objects, i.e. chemical compounds.

Hierarchical cluster analysis The starting point is a distance matrix containing the distances between all possible pairs of objects. The pair with the smallest distance is merged, and thereby the number of objects is reduced by one. The distance matrix is updated and the procedure continued until all objects are merged into a single cluster – example given in Figure 2[5].

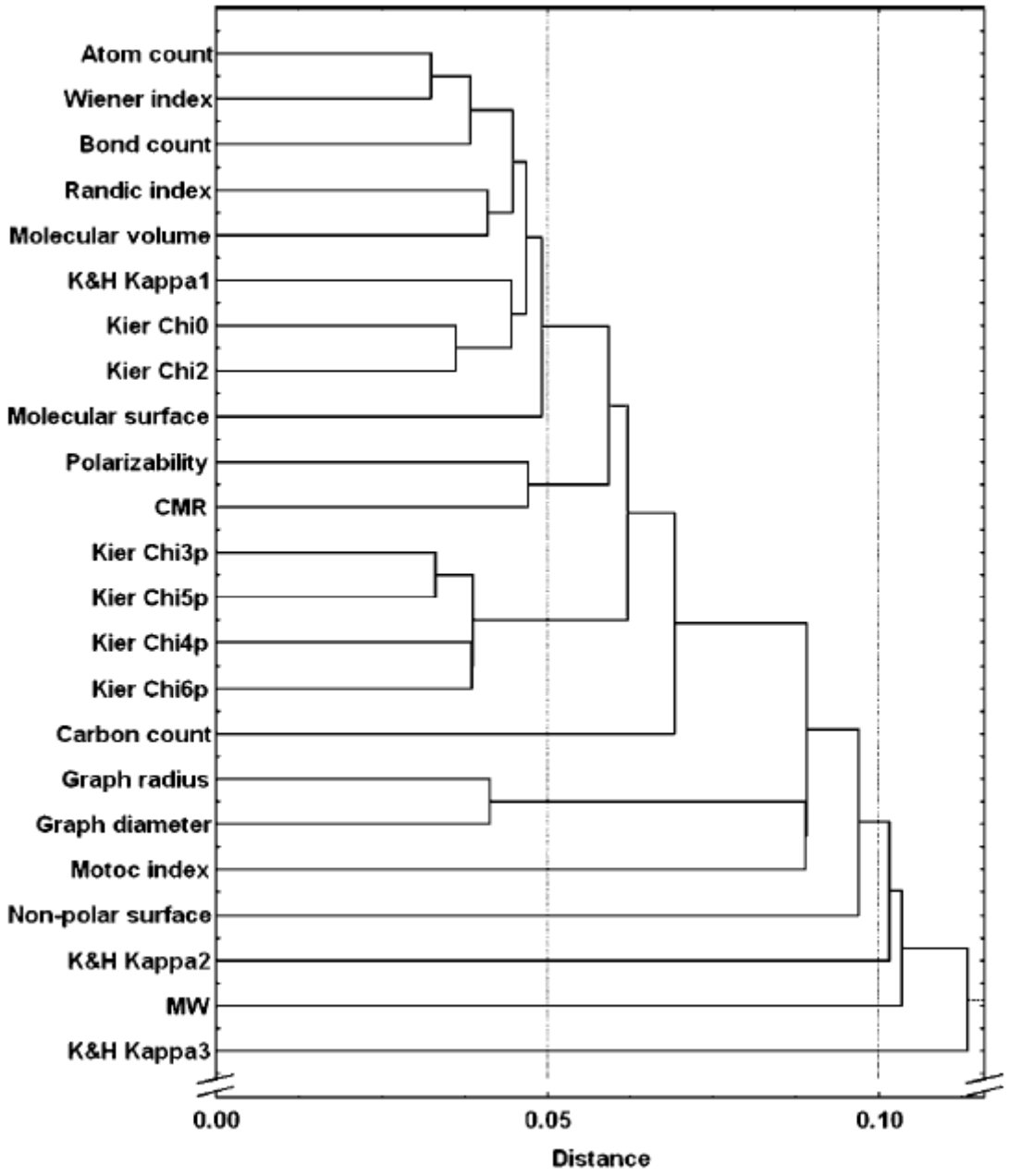

Figure 2. Clustering of the 2DProp descriptors (taken from reference[5])

The various types of agglomerative method differ in the method used to calculate the distance between merged points (clusters); in the simplest version the distance between the centroids is applied. It is up to the user to cut the dendrogram at some vertical position and to determine how many clusters are considered.

PLS mapping A less frequent used but powerful application of partial least squares (PLS) regression is the connected projection of X- and Y-data. Assume *n* objects (chemical compounds) are each characterized by *m* features (structural descriptors) and *k* properties (biological activities). These data can be collected in two matrices, X for the features and Y for the properties. The PLS approach uses X and Y together and generates a projection plane in X-space that is influenced by the Y-data and a projection plane in Y-space that is influenced by the X-data. An advantage of this connected projection is that clustering, i.e. of biological activity data, is influenced by structural data and *vice versa*; relationships between X and Y can be extracted more efficiently[6].

TASK 1. A good example on multivariate data analysis can be found in the below given paper. Please **read** and **understand** it! (download PDF1)

Olah, M.; Bologa, C.; Oprea, T.I. An Automated PLS Search for biologically relevant QSAR descriptors, *J. Comp.-Aided Mol. Design*, **2004**, *18*, 437-449.

1.1.3 Multivariate calibration

The aim of this method is the development of a mathematical model that describes the relationship between a set of *x*-features and one or several *y*-variables. There are several known techniques, such as (i) *ordinary least-squares regression* (OLS), complemented by (ii) *principal component regression* (PCR), (iii) *partial least squares regression* (PLS) (see above), and (iv) neural networks. The main applications in chemistry are quantitative analyses in infrared spectroscopy (especially NIR), evaluation of multi-sensor data, and investigations of structure-activity/property relationships (QSAR/QSPR).

Two topics of multiple linear regressions (MLR) are especially essential in chemometrics:

- The *x*-features are often highly inter-correlated. The idea is not to select a subset of less correlated features but rather to consider highly correlated features as multiple measurements that increase the stability of the model. Regression methods insensitive to correlated features are therefore to be applied.
- The complexity of the model is optimized to give the best prediction of the *y*-variable but not for best fit of given data. In PCR and PLS the model complexity is controlled by the number of used components (intermediate latent variables). A *training set* of data is used to calculate and to optimize the model parameters; a *test set* (*prediction set*) of data – not used in model generation – is applied to test the model. If the data set is small a *cross validation* procedure is used to estimate the predictive power of the model.

Ordinary least-squares regression (OLS) is a method of multiple linear regression that directly uses the features *x* to determine a final latent variable with maximum correlation coefficient with *y*. This criterion is equivalent to a minimum sum of squared residuals. OLS does not necessarily give the best model for prediction and is sensitive to outliers.

Principal component regression (PCR) models the dependent variable *y* as a linear combination of PCA scores. By definition, the PCA scores are uncorrelated and the regression model can be optimized for best prediction by varying the number of PCA components used. However, the optimum number of PCA used is determined by crossvalidation. The advantages of PCR are: (i) is strictly defined, (ii) the regressor variables (PCA scores) are uncorrelated and the directions of the corresponding intermediate latent variables are orthogonal, (iii) the same regressor variables can be used for different *y*-properties, and (iv) the regressor variables have maximum variance. A disadvantage I sthat the regressor variables are computed without considering *y* and, therefore, might not be optimum for building the model. The alternative method PLS is, therefore, often preferred.

1.1.4 Multivariate classification

Classification of objects to one of pre-defined classes is called *pattern recognition* and was one of the first applications of multivariate data analysis methods in chemistry[7,8]. One of the typical applications is classification of chemical substances with regard to biological activity.

As previously discussed, a matrix X contains features that characterize the objects, i.e., chemical compounds, and a matrix Y (or a vector *y*) contains variables with the encoded class membership of the objects. In binary classification, the objects are assigned to one of two mutually exclusive classes (yes/no decision); in these cases usually a single *y*-variable is used that for instance has the values -1 or +1 for class 1 or class 2, respectively. For more than two classes a separate binary variable can be used for each class, with the value 1 if the object belongs to this class and 0 otherwise.

One group of multivariate classification methods is based on the concept of linear latent variables. A so-called *discriminant variable* is calculated that optimally separates classes (in most cases, two) – see Figure 3.

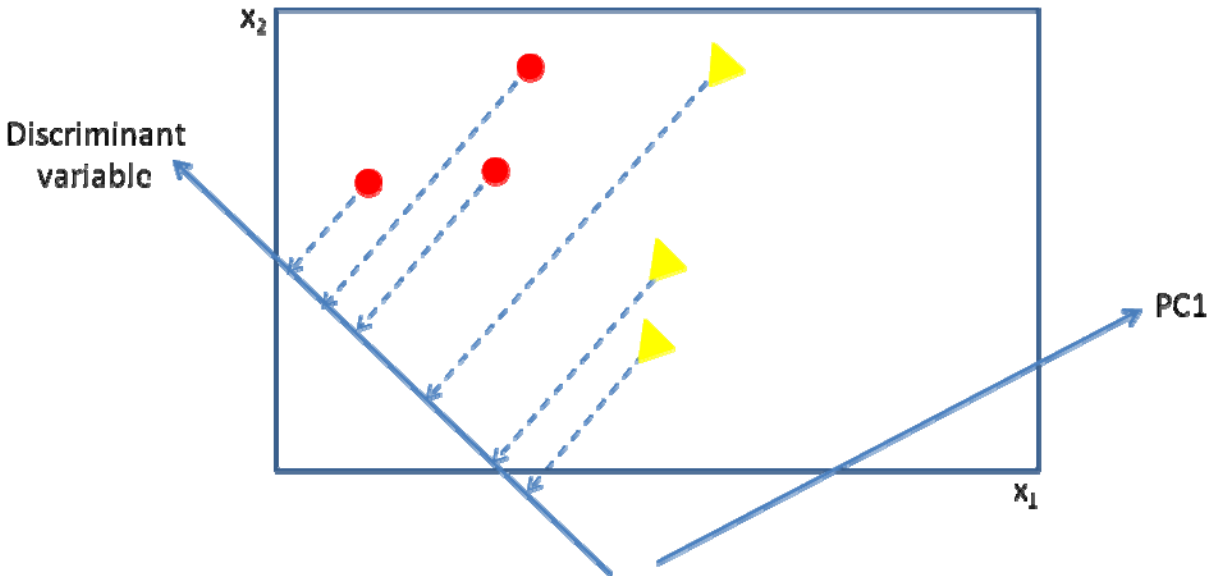

Figure 3. A discriminant variable is a latent variable that gives optimum separation of object classes. Projection on to the first principal component, PC1, does not separate the classes.

The loading vector that defines this latent variable is called *decision vector* or *classifier*. The mathematical criterion applied is maximum discrimination of the scores for objects from class 1and class 2, or optimum prediction of the *y*-variable that indicates class membership. For this approach, the methods of multiple regression are applied, i.e., linear discriminant analysis (LDA), PLS discriminant analysis. However, non-linear discriminant variables can be realized by neural networks[3] or by support vector machines[9].

Another group of classification methods is based on a modeling of the object classes and on distance measurements such as classification by distance measurement to class prototypes, and SIMCA – Soft/Simple Independent Modeling of Class Analogy[10,11]. The *k*-nearest neighbor classification (KNN) avoids any generalization of the data and does not define latent variables; a new object is classified by the majority of known objects situated nearest to the unknown.

Classification by distance measurement to class prototypes

A simple prototype of a class is the class centroid which is defined by the class mean vector. In a first approximation a new object is assigned to the class with the smallest Euclidean distance to the class centroid. Points with identical Euclidean distances to two class prototypes define a decision plane that is equivalent to the symmetry plane between the centroids. In a more general approach, Mahalanobis distances are used, considering the correlations between the features. Points with equal Mahalanobis distances from a centroid

define an ellipsoid with a shape similar to the distribution of the class. Calculation of Mahalanobis distances requires the inversion of the covariance matrix.

Soft/Simple Independent Modeling of Class Analogy (SIMCA) was created by Svante Wold[11]. Each class of objects is represented by a separate PCA or PLS model. For each object the probability of its belonging to a certain class or being an outlier can be estimated on the basis of *F*-test statistics[12]. Because of this flexibility SIMCA is called *soft*. Other classification methods like discriminant analysis are called *hard* because a discrete answer is given for class membership.

The simplest model of a class is the class centroid that constitutes the center of a hypersphere. The radius can be chosen so that 95% of the objects are situated within the hypersphere. More complicated models use PCA components as axes of a hyperellipsoid; the optimum number of components is determined by crossvalidation. Because SIMCA is based on PCA, it is not sensitive to a large number of features or to collinear features.

KNN classification

An unknown object can be classified by investigating the *k* nearest neighboring objects for which the class memberships are known. To find the nearest neighbors of an object it is necessary to compute the distances to all objects of a given data set. An unknown object is assigned to the class to which the majority of the *k* neighbors belong. Typical values for *k* are between 1 and 10.

The advantages of the KNN classification method are mathematical simplicity, classes of objects need to be linearly separable, easy application to multiclass problems, and no training of classifiers is required. The disadvantage is the lack of data reduction because a KNN classifier consists of all object vectors in the training set; this results in large computation times for large data sets.

1.2 Learning methods

1.2.1 Supervised learning

Supervised techniques involve learning from pre-classified training examples. Each training example is identified as belonging to a particular class. Typically, the goal of such methods is to derive a classification mechanism for correctly assigning new examples to the set of classes represented in the original set of training examples. The assumption is that the training data contained examples of all possible classes so that post-training examples do not represent members of previously unseen classes. The classification mechanism, whether it is in the form of rules or a network, can be thought of as a function that maps a description of an example to the name of the example's class. The accuracy of the classification mechanism can be determined with respect to a set of test examples by applying it to those examples and checking to see if they are properly classified.

Decision trees

Like most supervised learning methods, the goal of the decision tree methodology is to develop classification rules that determine the class of any object from the values of the object's attributes. In the case of decision trees, as the name implies, the classification rules are embodied in knowledge representation formalism called a decision tree[13]. This method has been used to derive structure-activity relationships (SAR)[14] and to learn classification rules for reactions.

Decision or *Classification Tree* analysis is a hierarchical, highly flexible set of techniques for predicting membership of objects or cases (i.e., chemicals) in the classes of a categorical dependent variable (i.e., aggregator, nonaggregator) from their measurements on one or more predictor variables (i.e., theoretical molecular descriptors, fingerprints). This method of statistical analysis represents one of the main techniques used in *data mining* and is technically known as binary recursive partitioning; (i) binary, because parent nodes are always split into exactly two child nodes, and (ii) recursive, because the process can be repeated by treating each child node as a parent.. The key elements of decision tree analysis are a set of rules for: (i) specification of the criteria for predictive accuracy, which is operationally defined as the prediction with the minimum "costs"; the term "costs" corresponds to the proportion of misclassified cases; (ii) selection of the splits on the predictor variables, which are used to predict membership in the classes of the dependent variables for the cases or objects in the analysis; these splits are selected one at time, starting with the split at the root node, and continuing with splits of resulting child nodes until splitting stops, and the child nodes which have not been split become terminal nodes; (iii) determination of the stop splitting, which is the user defined criteria that allow splitting to continue until either all terminal nodes are pure or contain no more than a specified minimum (a) number of cases or objects, or (b) fraction of the sizes of one or more classes; and (iv) selection of the "best-sized" tree is done from the sequence of optimally pruned trees based on the cross-validation (CV) costs; usually, the "best-sized" tree is the smallest-sized (least complex) tree whose CV costs do not differ appreciably from the minimum CV costs.

A decision tree used for the classification of a set of chemicals as (i) aggregators (promiscuous aggregating inhibitors), and (ii) nonaggregators is shown in Figure 4 (*unpublished work*). According to McGovern *et al.*[15,16] aggregators are nonspecific / non-lead-like small organic molecules that form submicrometer aggregates (~30 – 400 nm), which are responsible for the inhibition of many different enzymes through protein adsorption onto the surface of aggregates, although absorption certainly cannot be excluded. In fact, these molecules form large colloid-like aggregates that sequester and thereby inhibit enzymes.

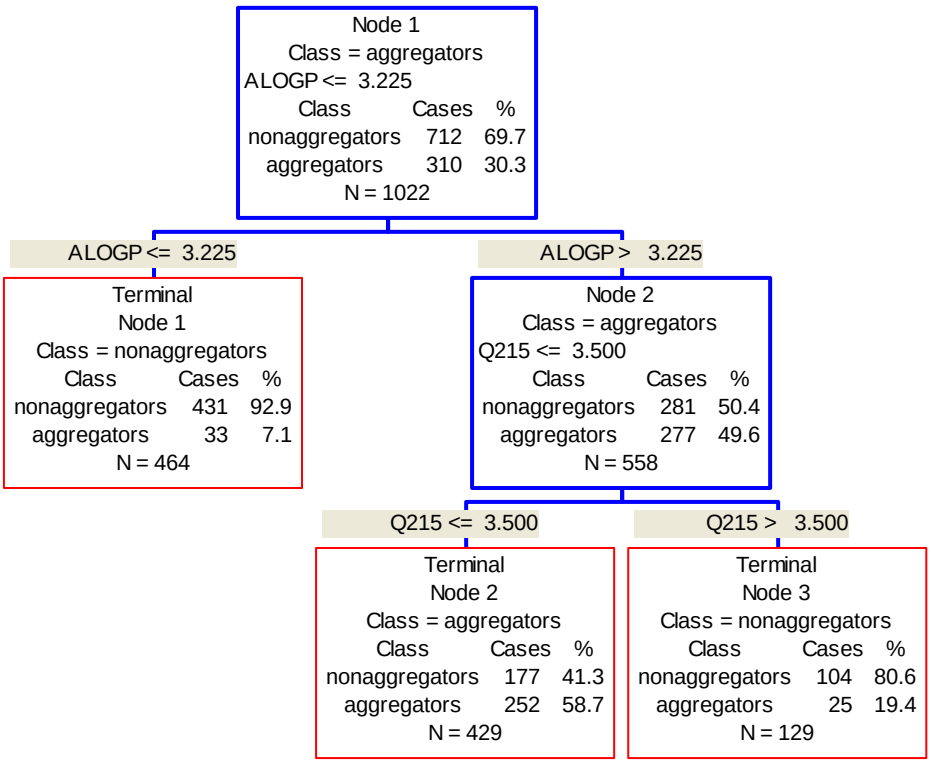

Figure 4. The classification trees with lowest relative cost developed using a set of 1D and 2D descriptors. As can be seen, the major splitter (at root node) is ALOGP (estimated value of the octanol/water partition coefficient). The second splitter (at child node) is represented by one of the SMDL SMARTS – Q215, [#7!H]~[#6R!H]

In general, many different tree structures are possible. Shorter trees are usually preferred because they express simpler classification rules that are easier to understand.

TASK 2: What is the 2D representation of the SMDL SMARTS Q215?

Bayesian Belief Networks

An important consideration in many learning scenarios is the reliability of data and/or missing values. One learning method that is designed to reason in cases of uncertainty is that of Bayesian probabilistic networks[17].

At a qualitative level, a Bayesian belief network is a directed acyclic graph in which the vertices represent domain objects. The links (arcs) between the vertices of the network denote casual relations between the objects represented by the vertices.

Quantitatively, the casual relations expressed between vertices by the arcs are represented by conditional probabilities. For each vertex with parents there is a conditional probability distribution. For each vertex without parents there is a prior-probability distribution. These probabilities describe the uncertainty in the casual relationship between the attributes represented by the vertices. When the prior and conditional

probabilities have been learned from pre-classified training examples, it is possible to compute any probability of the form. In particular, given an unclassified instance, it is possible to determine its classification with an associated probability. In addition, the method still works even when values for some of the attributes are missing. This method is ideal when complete information is not available.

The approach used to calculate missing values or to determine the propagation of effects caused by changing or setting the value of a specific variable in a network is that of probabilistic propagation in trees. A hallmark of this method is that the probabilities can be computed solely on the basis of communication between neighboring vertices in which the intervertex communication occurs. In bioinformatics, Bayesian belief networks are used for modeling knowledge in gene regulatory networks, protein structure information etc.

Support Vector Learning

The Support Vector Machine (SVM) belongs to the class of kernel-based learning methods. A key idea underlying SVMs is the selection of features from a very high dimensional feature space and the use of a simple linear hyper-plane decision surface to effect a classification. SVMs provide a new approach to the problem of learning with a clear connection to statistical learning theory.

Statistical learning theory shows that for a given number of training examples, there is a trade-off between the degree to which the empirical error can be minimized and to which the empirical error will deviate from the true error. It has been shown that the capacity for generalization is related to the complexity of the classification functions from which the learning machine can choose[18,19]. Essentially, the theory says that a “simple” classifier that explains most of the training data is preferable to a complex one (Occam’s razor). SVMs have been successfully applied in computational chemistry[20,21,22].

Genetic Algorithms (GA)

The area of genetic algorithms has been inspired by ideas from biology such as genetic recombination, natural selection, and random mutation. Under the genetic algorithm paradigm, objects, typically solutions to problems are represented as bit strings. A fitness function is used to measure the goodness of solutions. The genetic operators are then applied to objects to generate new objects in proportion to their fitness. The goal is to find objects with maximum fitness.

1.2.2 Unsupervised learning

The problem of unsupervised learning is more difficult than that of supervised learning. The task involves looking for regularities in training examples that have not been pre-classified. Because of that, part of the learning task entails deriving classifications. Typically, such learning systems implement a bias to restrict the hypothesis space and to form a preference among hypotheses. The motivation is to restrict hypotheses from what might be combinatorially possible to those that are conceptually meaningful and then rank them according to the utility of the concepts they express.

Unsupervised concept learning can be accomplished by first analyzing the unclassified training examples and forming a taxonomy. Next, the descriptions of the examples in the resulting classes can be generalized to form class descriptions expressing the concepts embodied by the members of each class.

The basic taxonomy task is that of classifying a set of objects represented by attribute-value pairs. This task can also be viewed as a two-part process, first identifying the classes represented by the original set of objects, and then classifying those objects according to the identified classes.

Artificial Neural Networks (ANN)

ANN are a method or, more precisely, set of methods that has recently found very intensive use among chemists. ANN are not one method, but several different computational methods featuring a wide variety of formal layouts, learning strategies, and sought goals. In the context of ANN the term *layout* means a *set-up* or *architecture* of artificial neurons forming the network; *learning strategy* means a mathematical procedure and conditions put on to the data set to obtain the final (the *trained*) ANN. By the term *sought goal* the final result as produced by the trained ANN, like *mapping*, *classification*, *model*, etc., is understood.

ANN are difficult to summarize within a simple definition. Maybe the closest definition would be comparison with a black box having multivariate input $X = (x_1, x_2, x_3, \ldots, x_m)$ and multi-response output $Y = (y_1, y_2, y_3, \ldots, y_n)$ – see Figure 5.

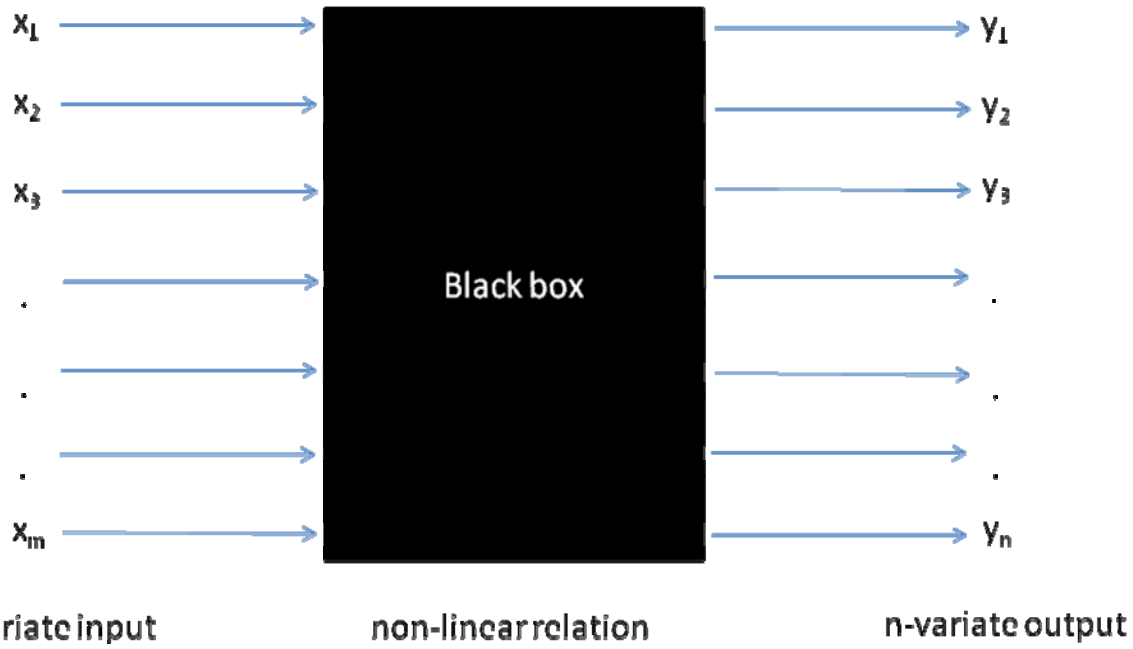

Figure 5. Artificial neural network (ANN) as a black box. Multivariate input is transformed in a non-linear manner to the n-response output

ANN methods can only be used if a comparably large set of multivariate data is available which enables one to train an ANN by example. Note that the ANN methods work best if they are dealing with non-linear relationship between complex inputs and outputs. Although in principle ANN can be used to model one-to-one input/output relations this is rarely done in practice. To make an ANN operational it must be trained with a known set of data. Even ANN can model linear relationships also, the final result might often be worse than results obtained by simpler standard statistical techniques. The first thing to be considered when employing an ANN is the nature of the problem: does it require a supervised or an unsupervised approach?

In general, problems associated with data handling at an early stage require unsupervised methods. Only further on in the later steps of data handling, after one became more familiar with the measurement space, with the

distribution of input variables, and with the behavior of the responses, can one select sets of data on which supervised methods (modeling for example) can be conducted.

The basic types of problems in chemistry for which ANN can be used are:

- Selecting samples from among a large number of samples for further handling (sampling);
- Assigning an unknown sample to a specific class of a pre-defined (known in advance) number of classes (classification);
- Finding the inner structure of relationships between samples within the given measurement space (clustering of objects);
- Making direct and inverse models for quantitative prediction of behavior, time changes, or effects of unknown or not-yet seen samples (modeling).

2. Quantitative Structure-Activity Relationships (QSAR) – 3D QSAR

The underlying theory of Quantitative Structure-Activity/Property Relationship (QSAR/QSPR) analysis is that biological activity/physicochemical property is directly related to molecular structure. Therefore molecules with similar structure will possess similar bioactivities for similar proteins/receptors/enzymes and the changes in structure will be represented through the changes in the bioactivities/physicochemical properties.

The aim of QSAR/QSPR is to find a quantitatively correlation between the structure of compounds and their biological activity or physicochemical properties and derive a model to predict the activity or properties of novel compounds. For example, biological activity can be expressed quantitatively as in the concentration of a substance required to give a certain biological response. Additionally, when physiochemical properties or structures are expressed by numbers, one can form a mathematical relationship, or quantitative structure-activity relationship, between the two. The mathematical expression can then be used to predict the biological response of other chemical structures. A model usually consists of a linear or multilinear combination of features, descriptors, and coefficients.

The most general mathematical form of a QSAR/QSPR model is:

$$\text{Activity / Property} = \boldsymbol{f}\,(\text{Structural Descriptors})$$

As it is possible to generate a large number of descriptors for each compound, the selection of features that yield a reliable relationship is a complex and time-consuming task. Nowadays, there are several methods widely used in this area.

The *linear or nonlinear multiple regression analysis* (MLR) is one of the primarily methods used as a statistical tool to derive quantitative models, and to check the significance of these models and of each individual term in the regression equation (a brief mathematical description of the method can be found on PDF2). Other statistical methods, such as multivariate data analysis (principal component analysis, PCA, or partial least squares, PLS, analysis), and discriminant analysis are alternatives to regression analysis (see above).

2.1 QSAR/QSPR – general algorithm[23]

The QSAR/QSPR model development and validation involves several major steps, as presented in Figure 6:

i. *decision* (of the property to be modeled);
ii. *construction* (of the data set);
iii. *generation* (of the molecular structures);
iv. *feature selection* (of the most informative subset of descriptors);
v. *development* (of the QSPR model);
vi. *interpretation* (of the model);
vii. *validation* (of the model);
viii. *prediction* (of the property of interest).

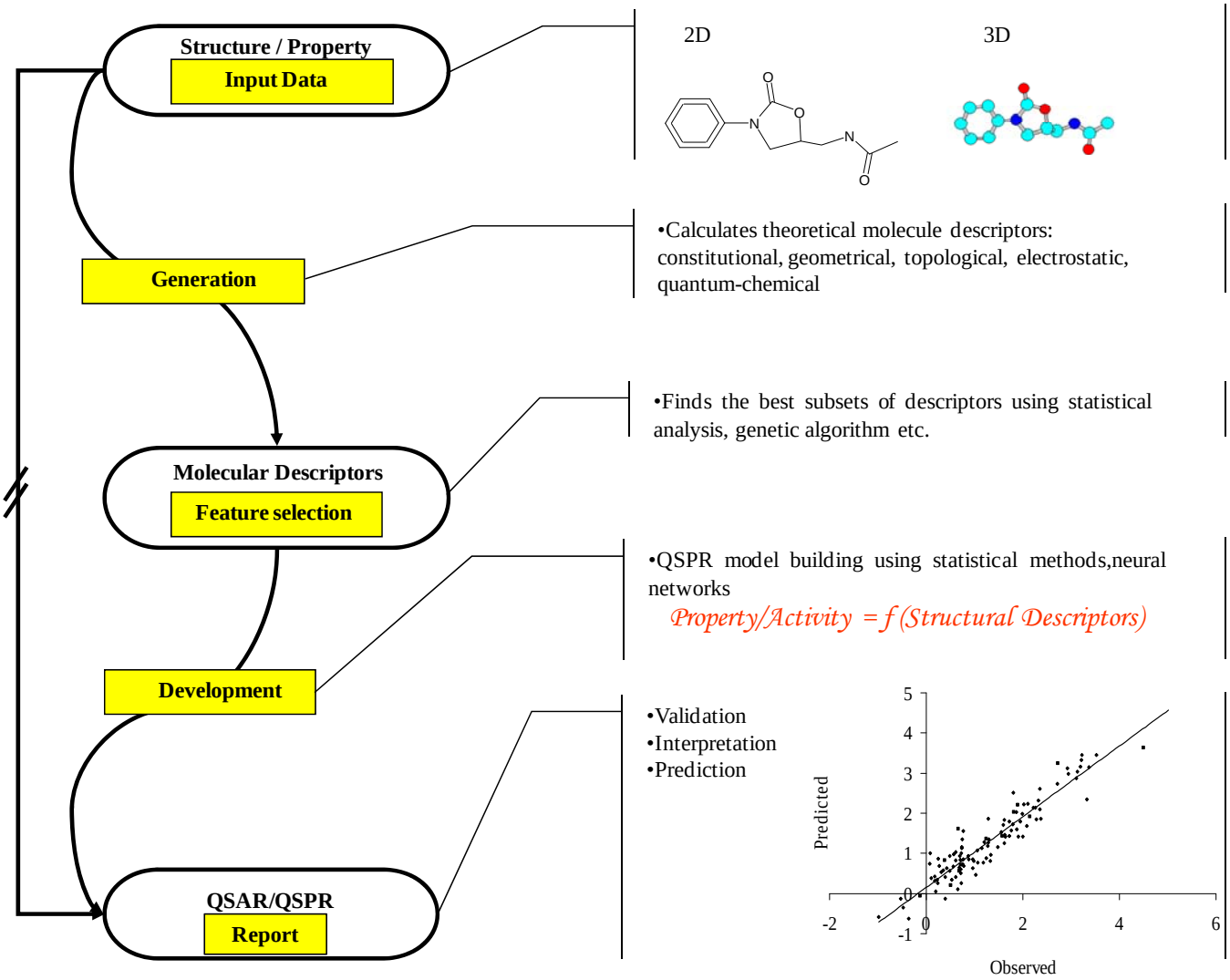

Figure 6. Flow chart of a QSAR/QSPR model generation (taken from reference[23])

A ***Decision*** implies choosing of the property to be studied that is highly dependent on the availability of experimental data. A large amount of heterogeneous experimental values lead to more stable and valid QSAR models.

A ***Construction*** represents the assembling of the data set that consists of 2D and 3D molecular structures obtained by molecular modeling (the three-dimensional rendering of molecules in an environment produced by computer graphics). The molecular modeling, as part of the computational chemistry, includes the so-called computational methods by which the spatial coordinates of the atoms in molecules and the respective

interconnecting bonds are produced. These computational methods can be divided into two major groups: (i) molecular mechanics methods, and (ii) quantum mechanics/quantum chemical methods. Group (ii) is usually subdivided into *ab initio* and semi-empirical methods[24]. In molecular mechanics, the potential energy of a given conformation of molecular systems is calculated as a function of the coordinates of their atomic nuclei, which are treated as Newtonian particles under the influence of a potential energy function or force field. A force field is given by equation (1) together with the data (parameters) required to describe the behavior of different kinds of atoms and bonds.

$$E = E_{\text{stretching}} + E_{\text{bending}} + E_{\text{torsion}} + E_{\text{non-bonded interaction}} \quad (1)$$

where E – potential energy, $E_{\text{stretching}}$ – bond stretching energies, E_{bending} – angle bending energies, E_{torsion} – torsional energies and $E_{\text{non-bonded interaction}}$ – energies of non-bonding interactions. The potential functions generated by molecular mechanics are often useful for comparing different configurations of a molecule.

Most molecular mechanics methods use empirical data to determine individual force constants (for example, bond stretch constants) and equilibrium ("strain-free") values for each geometrical characteristics of a molecule (for example, bond lengths)[25].

The quantum mechanical methods are based on Schrödinger equation[1] in which the electrons are described by the respective molecular wavefunctions. For molecular systems, the Schrödinger equation can only be solved approximately.

Typical *ab initio* electronic structure methods consider the potential energy operator as time-independent and solve only the spatial wavefunction. The a*b initio* calculations can be performed at the Hartree-Fock level of approximation[24] or using various post-Hartree-Fock theories (configuration-interaction, multiconfiguration self-consistent field etc.).

Semi-empirical methods have been developed as approximations to *ab initio* techniques that were fitted to experimental data (e.g. structures and formation energies of organic molecules, spectroscopic data and ionization energies). In this case, the Hartree-Fock self-consistent field method is used to solve the Schrödinger equation with various approximations. Semi-empirical methods are usually classified according to their treatment of electron-electron interactions[25].

A ***Generation*** is based on previous presented computational methods (*ab initio*, semi-empirical) and involves the calculation of the so-called molecular structure descriptors in order to encode the compounds (see *Section III.* **Molecular descriptors and fingerprints**)

A ***Feature selection*** of descriptors involves finding the most informative subsets of descriptors from those in the descriptor pool due to the fact that models with too many variables lead to bad predictions because of over fitting. Various methods, like the classical forward selection, backward elimination and stepwise regression[26], or, more recently, the genetic algorithms (GA) can be used for the selection of the best combination of descriptors. The genetic algorithms have been shown to be very effective in performing descriptor selection in the case of large descriptor pools[26,27].

A ***Development*** involves the application of the descriptors using (i) multilinear regression analysis, such as the Heuristic and best multilinear regression (BMLR) methods[28], (ii) multivariate statistical methods, such as principal component analysis (PCA) and partial least squares (PLS)[29,30,31] or (iii) non-linear methods, such as

artificial neural networks[32] (ANNs) to build mathematical models linking the descriptors directly to the chemical property under investigation.

The multilinear regression methods gave a linear regression equation in which the property, Y, is a linear function of descriptor values, X: Y = f(X). Several statistical characteristics give information about the "goodness" of the model: R^2 – squared correlation coefficient, R^2_{CV} – squared cross-validated correlation coefficient, F – Fisher criterion value, s^2 – squared standard error.

In the principal component analysis (PCA), the original data matrix is decomposed into new latent variables, and the variation among the objects (e.g. organic compounds), is illustrated in score plots. The corresponding loading plots describe the corresponding variation among the descriptors. In the case of partial least squares (PLS), the latent information in the independent parameters is related to the dependent variable (i.e. property) through a weight vector. In both cases, the number of significant descriptors is determined by a cross-validation procedure[33] and the reliability and stability of the models are estimated by the following statistical measures: the variation explained in the X-matrix (descriptors) (R^2X), the variation explained in Y (property) (R^2Y), the cross-validated explained variance (Q^2), and the root-mean-square error of estimation (RMSEE) and prediction (RMSEP)[34].

A neural network is a non-linear method capable of modeling extremely complex functional relationships. It consists of a process of assimilation in the cognitive system similar to the neurological functions of the brain and is capable to predict new observations from other observations (modeling) after the executing of the so-called "learning process" from existing data. During the last decades, by using diverse "training" algorithms[27] various types of neural network architectures have been developed. Recent investigations involving computational neural networks and genetic algorithms serve as examples of the application of the QSAR/QSPR methods[35]. Three-layer, feed-forward neural networks trained with a quasi-Newton method have provided excellent results in several QSAR/QSPR studies[35].

An ***Interpretation*** of the model implies (i) the explanation of how each of the descriptors from the selected "best" subset contributes to the property of interest, (ii) the assessment of reliable physicochemical meaning to descriptors, especially for those provided by multivariate statistical analysis.

A ***Validation*** of the developed models is an important aspect of any QSPR study. Once a regression equation is obtained, it is important to determine its reliability and significance. Several procedures are available to assist in this. These can be used to check that the size of the model is appropriate for the data available, as well as to provide some estimate of how well the model can predict activity for new molecules.

An Internal validation uses the data set from which the model is derived and checks for the internal consistency. For the internal validation, the parent data set can be divided into several subsets; e.g. the 1^{st}, 4^{th}, 7^{th}, etc. entries go into the first subset (#1), the 2^{nd}, 5^{th}, 8^{th}, etc. into the second subset (#2), and the 3^{rd}, 6^{th}, 9^{th}, etc. into the third subset (#3). Then, three training sets *Set 1, Set 2 and Set 3* are prepared as combination of two subsets (#1 and #2), (#1 and #3), and (#2 and #3), respectively. The corresponding test sets relate to remaining subsets (#3, #2 and #1, respectively). For each training set, the correlation equation is derived with the same descriptors, and the obtained equation is used to predict the values of the property of interest for the compounds from the corresponding test set[36,37].

An External validation evaluates BEST how well the equation generalizes the data. The original data set is divided into two groups – the training set and the test set. The training set is used to derive a model that further is used to predict the properties of the test set members, which were NOT used to train the model.

The Cross-validation repeats the regression many times on subsets of the data. Usually, each molecule is left out in turn, and the R^2 (cross-validated correlation coefficient) is computed using the predicted values of the missing molecules (*leave-one-out*, LO). Another possibility is to leave out more than one molecule at a time (i.e. 30%), though not all possible combinations of molecules are used, a concession that makes the computation tractable (*leave-many-out*, LM). However, if n molecules are removed once from a total set of m, then nXm regressions are performed[38].

In multivariate statistical analysis, cross-validation is often used to determine the number of principal components retained in the model – those that give the highest cross-validated R^2 are to be considered.

The Monte Carlo cross-validation (MCCV)[39] was developed as an asymptotically consistent method in determining the number of components that can avoid an unnecessary large model and therefore decreases the risk of over-fitting in the model.

A ***Prediction*** involves the use of the developed QSPR models to estimate the property of interest for unknown compounds. If prediction is the main goal of the regression, then a test set data must be reserved strictly for evaluation until the equation is finalized: this reserved data cannot enter the regression process in any form.

This general QSAR/QSPR methodology can be applied, in principle, to any property that can be related to the molecular structure, and it has been applied for the description and prediction of a wide variety of chemical, physical and environmental properties and activities[35].

2.2 3D-QSAR methodology

It is an improvement on "traditional' (two-dimensional; 2D) QSAr by incorporating the molecules' 3D structure, steric, electrostatic, and interactive phrmacophore elements into building the QSAR model. There are two sub-fields of 3D-QSAR: (i) receptor-independent (Ri) methods construct QSAR models without the assistance of a 3D structure of the receptor, and (ii) receptor-dependent (RD) methods use the 3D structure of the receptor to aid in the construction of the QSAR model through the use of ligand-receptor interaction descriptors.

There are two major points of contention common to all 3D-QSAR methodologies: the alignment and the conformation of the molecules in a study. The most common reason a 3D-QSAR model fails is that the non-bioactive conformation of the ligand is used in the QSAR study. In 1980 Hopfinger[40] included the additional pseudo-dimension of an ensemble of conformations to the 3D-QSAR paradigm, thus creating the field of nD-QSAR. This additional dimension eliminated the question of which conformation to use in a QSAR study. A qualitative 3D-QSAR method, Probabilistic Receptor Potentials[41], calculates ligand atomic preferences in the active site. The potentials for PRP are constructed by fitting analytical functions to experimental properties of the substrates using knowledge-based methods. Another advance in the field of QSAR was the development of Virtual High-Throughput Screening (VHTS) methods[42] and Virtual 3D-Pharmacophore Screening[43]. These QSAR methods have been devised to assist in computer-aided drug design by contributing different information about how the ligand interacts with its potential receptor site.

The focus of 3D-QSAR is to identify and characterize quantitatively the interactions between the ligand and the receptor's active site. The interactions between the 3D atomic coordinates of the molecule and the receptor are correlated with the bioactivities, producing a 3D-QSAR model. There are several methods of achieving the creation of QSAR models based on the atomic coordinates. The most widely known 3D-QSAR methodology is

Comparative Molecular Field Analysis (CoMFA)[44], which is based on placing the ligands in a 3D grid and using a probe to map the surface of the ligand on the basis of the ligand's interaction with the probe. Comparative Molecular Similarity Index Analysis (CoMSIA)[45] and Self-Organizing Molecular Field Analysis (SOMFA)[46] are based on the same methodology as CoMFA. CoMSIA works by separating and projecting where physicochemical properties rationalize the binding affinity in 3D space by this method. Using fundamental molecular properties and "mean centered activity", SOMFA creates a QSAR model based on molecular shape and electrostatic potentials (ESP)[46]. 3D-QSAR methods that do not rely on a grid or alignment are Autocorrelation of Molecular Surface Properties (AMSP)[47], Molecular Representation of Structures based on Electron diffraction (3D-MoRSE)[48], and Comparative Molecular Moment Analysis (CoMMA)[49]. The methods of AMSP and 3D-MoRSE map the physical properties of the ligands to a van der Waals surface and individual atoms, respectively. The physical properties are transformed into a vector of equal length for all the ligands in the study, eliminating the problem of molecular alignment. CoMMA removes the alignment issue by calculating molecular moments and uses them as descriptors to construct a QSAR model. The qualitative method of Probabilistic Receptor Potentials (PRP)[41] is able to predict the type of receptor atoms most likely to interact with the ligand, thus providing an approximation of the type of atoms in the active site. The technique of 3D-QSAR is built on methods that take into consideration the 3D structure of the molecules and their inherent physical properties (with respect to how the ligand will interact with the receptor) to construct a QSAR model. All of these methods are receptor-independent (RI); the models are created without any firm knowledge of the composition or 3D structure of the receptor.

TASK 3. A comprehensive review on 3D-QSARs in lead design has been recently written by Leitao *et al.* Please read it! (download PDF3).

---

[1] Wold, S.; Sjöström, M.; Eriksson, L. *The Encyclopedia of Comptational Chemistry*, Schleyer, P.v.R.; Allinger, N.L.; Clark, T.; Gasteiger, J.; Kollman, P.A., Schaefer, H.F., III; Schreiner, P.R. (Eds.), Wiley and Sons, Chichester, UK, 1998, pgs. 2006-2021.

[2] Siedlecki, W.; Siedlecka, K.; Sklansky, J. *Pattern Recognition* **1988**, *21*, 411-429.

[3] Zupan, J.; Gasteiger, J. *Neural Networks in Chemistry and Drug Design*, 2nd Edition, Wiley-VCH, Weinheim, 1999.

[4] Kohonen, T. *Self-Organizing Maps*, Springer, Berlin, 1995.

[5] Olah, M.; Bologa, C.; Oprea, T.I. *J. Comp.-Aided Mol. Design* **2004**, *18*, 437-449.

[6] Werther, W.; Demuth, W.; Krueger, F.R.; Kissel, J.; Schmid, E.R.; Varmuza, K. *J. Chemometrics* **2002**, *16*, 99-110.

[7] Crawford, L.R.; Morrison, J.D. *Anal. Chem.* **1968**, *40*, 1469-1474.

[8] Jurs, P.C.; Kowalski, B.R.; Isenhour, T.L. *Anal. Chem.* **1969**, *41*, 21-27.

[9] Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*, Cambridge University Press, UK, 2000.

[10] Brereton, R.G. *Chemometrics. Applications of Mathematics and Statistics to Laboratory Systems*, Ellis Horwood, New York, 1990.

[11] Wold, S. *Pattern Recognition* **1976**, *8*, 127-139.

[12] Kvalheim, O.M.; Karstang, T.V. *Multivariate Pattern Recognition in Chemometrics, Illustrated by Case Studies*, R.G. Brereton (Eds.), Elsevier, Amsterdam, 1992, pgs. 209-248.

[13] Quinlan, J.R. *Machine Learning* **1986**, *1*, 81-106.

[14] King, R.D.; Hirst, J.D.; Sternberg, M.J.E. *Appl. Artif. Intell.* **1994**, *9*, 213-234.

[15] McGovern, S.L; Caselli, E.; Grigorieff, N.; Shoichet, B.K. *J. Med. Chem.* **2002**, *45*, 1712-1722.

[16] McGovern, S.L.; Helfand, B.T.; Feng, B.; Shoichet, B.K. *J. Med. Chem.* **2003**, *46*, 4265-4272.

[17] Neapolitan, R.E. *Probabilistic Reasoning in Expert Systems. Theory and Algorithm*, Wiley, New York, 1990.

[18] Vapnik, V.N. *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[19] Vapnik, V.N. *Statistical Learning Theory*, Wiley, New York, 1998.

[20] Demiriz, A.; Bennett, K.P., Breneman, C.M. *Support vector machine regression in chemometrics*. In *Computing Science and Statistics*: Proceedings of Interface, 2001.

[21] Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. *Drug design by machine learning: support vector machines for pharmaceutical data analysis*, *Computers and Chemistry*, **2001**, *26*, 4-15.

[22] Trotter, M.; Buxton, B.; Holden, S.B. *Support Vector Machines in Combinatorial Chemistry, Measurement and Control*, **2001**, *34*, 235-239.

[23] Fara, D.C. *QSPR Modeling of Complexation and Distribution of Organic Compounds, Dissertationes Chimicae Universitatis Tartuensis*, Tartu University Press, Tartu, 2004.

[24] Lowe, J. P. *Quantum Chemistry*; 2nd ed. Academic Press, San Diego, California, 1993.
[25] Burkert, U.; Allinger, N. L. (Eds.), *Molecular mechanics*, Am. Chem. Soc. Monograph, Washington D.C., 1982.

[26] Xu, L.; Zhang, W.-J. *Anal. Chim. Acta* **2001**, *446*, 477-483.

[27] Deaven, D. M.; Ho, K. M. *Phys. Rev. Lett.* **1995**, *75*, 288.

[28] Katritzky, A. R.; Lobanov, V. S.; Karelson, M.; Murugan, R.; Grendze, M. P.; Toomey, J. E. *Rev. Roum. Chim.* **1996**, *41*, 851-867.

[29] Gabrielsson, J.; Lindberg, N.-O.; Lundstedt, T. *J. Chemom.* **2002**, *16*, 141-160.

[30] Willett, P. *Similarity and Clustering in Chemical Information Systems*; John Wiley: New York, 1987.

[31] Jackson, J. E. *A users guide to principal components*; John Wiley & Sons: New York, 1991.

[32] Schneider, G.; Wrede, P. *Progr. Biophys. Mol. Biol.* **1998**, *70*, 175-222.

[33] Wold, S. *Technometrics* **1978**, *20*, 397-405.

[34] *SIMCA-P 8.0*; product of Umetrics AB, Umeå, Sweden, 2000; http://www.umetrics.com/

[35] Katritzky, A. R.; Fara, D.; Tatham, D.; Petrukhin, R.; Maran, U.; Lomaka, A.; Karelson, M. *Curr. Top. Med. Chem.* **2002**, *2(12)*, 1333-1356.

[36] Katritzky, A. R.; Wang, Y.; Sild, S.; Tamm, T.; Karelson, M. *J. Chem. Inf. Comput. Sci.* 1998, *38*, 720.

[37] Maran, U.; Karelson, M.; Katritzky, A. R. *Quant. Struct. Act. Relat.* 1999, *18*, 3.

[38] Allen, D. M. *Technometrics* **1974**, *16*, 125-127.

[39] Xu, Q.-S.; Liang, Y.-Z. *Chemom. Intell. Lab. Syst.*. **2001**, *56*, 1-11.

[40] Hopfinger, A.J. *J. Am. Chem. Soc.* **1980**, *102*, 7196-7206.

[41] Labute, P. *J. Chem. Comput. Group* **2001**; http://www.chemcomp.com/feature/cstat.htm

[42] Labute, P. *J. Chem. Comput. Group* **2001**; http://www.chemcomp.com/feature/htsbqsar.htm

[43] Hopfinger, A.J.; Reaka, A.; Venkatarangan, P.; Duca, J.S.; Wang, S. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1151-1160.

[44] Cramer, R.D., III; Patterson, D.E.; Bunce, J.D. *J. Am. Chem. Soc.* **1998**, *110*, 5959-5967.

[45] Klebe, G.; Abraham, U.; Mietzner, T. *J. Med. Chem.* **1994**, *37*, 4130-4146.

[46] Robinson, D.D.; Winn, P.J.; Lyne, P.D.; Richards, W.G. *J. Med. Chem.* **1999**, *42*, 573-583. SOMFA is available at http://bellatrix.pcl.ox.ac.uk/downloads/

[47] Wagener, M.; Sadowski, J.; Gasteiger, J. *J. Am. Chem. Soc.* **1995**, *117*, 7769-7775.

[48] Schuur, J.H.; Seizer, P.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 334-344.

[49] Silverman, B.D.; Platt, D.E. *J. Med. Chem.* **1996**, *39*, 2129-2140.