Chapter 1

Chemoinformatics (1; 2) *is an emerging science that concerns the mixing of chemical information resources to transform data into information and information into knowledge. It is a branch of theoretical chemistry based on its molecular model, and which uses its own basic concepts, learning approaches and areas of application.*

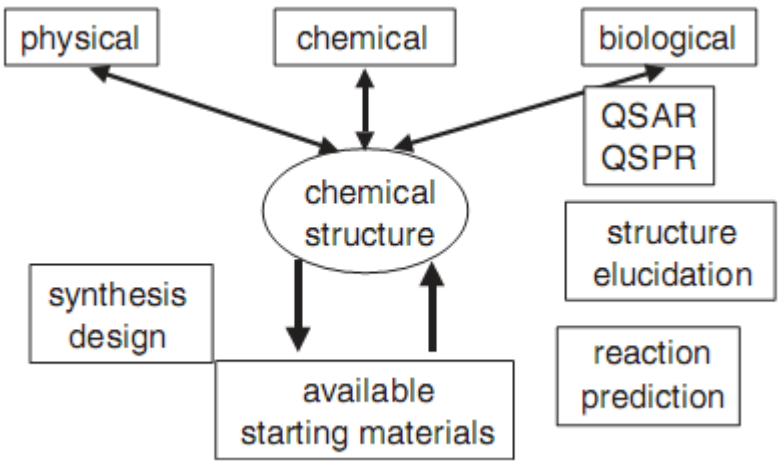
1.1. General introduction

A major task of chemists is to make compounds with desired properties. The society at large is not interested in beautiful chemical structures but in the properties that these structures carry with them. Chemical industry can only sell properties but they do so by conveying these properties through chemical structures. Thus, the first fundamental task in chemistry is to make inferences on which structure might have the desired property. This is the domain of establishing structure–property or structure–activity relationships (SPR or SAR) or even finding such relationships in a quantitative manner (QSPR or QSAR). Once we have an idea which structure we should make to obtain the desired property we have to make a plan on how to synthesize this compound, which reaction or sequence of reactions to perform to make this structure from available starting materials. This is the domain of synthesis design, and the planning of chemical reactions.

Once a reaction has been performed, we have to establish whether the reaction took the desired course, whether we obtained the desired structure. For, our knowledge on chemical reactions is still too cursory; the factors influencing the course of chemical reactions are too many that we are not always able to predict which products will be obtained, whether side reactions will be observed, or whether the reaction might take a completely different course than expected. Thus, we have to establish the structure of the reaction product. A similar problem is given, when the degradation of a xenobiotic in the environment or in a living organism has to be established. This is the domain of structure elucidation, that, in most parts, utilizes information from a battery of spectra (infrared, NMR, and mass spectra). These fundamental tasks of a chemist are summarized in Figure (1-1). All these tasks are, in general, too complicated to be solved from first principles. They require a lot of knowledge, knowledge that has to be derived by learning from data and from observations made on experiments. It has to be realized that there are two ways

of learning, deductive and inductive learning. In deductive learning a theory is used to make inferences, deductions. In chemistry this is usually achieved by calculations such as quantum mechanical or molecular mechanics calculations. Such calculations provide data that can assist in solving a problem.

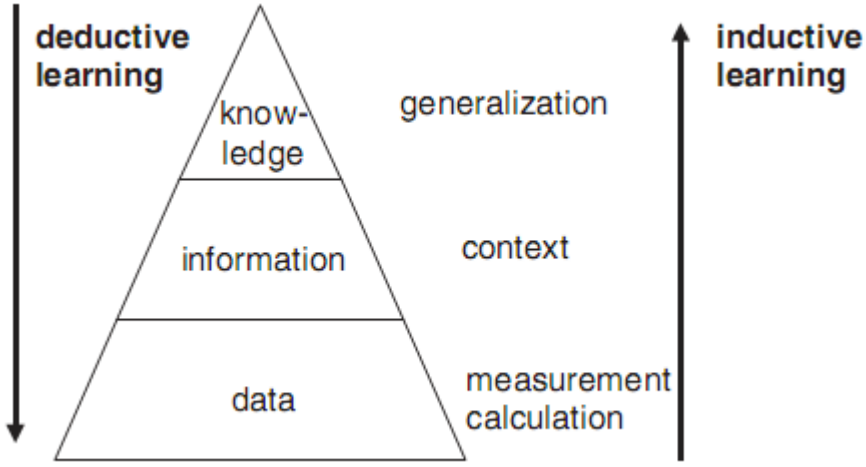
Inductive learning, on the other hand, learns from observations, from data. These data are put into context to obtain information. Information can then be generalized to obtain knowledge Figure (1-2). To give an example: the measurement of a certain biological activity is, by itself, not very useful. Only when we can associate such a biological activity with a chemical structure do we obtain information. Many such pieces of information of chemical structures and their associated biological activities can then be used to build a model for the relationships between chemical structure and biological activity. Such a model comprises knowledge that can be used to make predictions on the biological activity of new chemical structures.



Figure(1-1) The fundamental tasks of a chemist: property prediction, synthesis design, reaction prediction, and structure elucidation.

Inductive learning has a long history in chemistry. In fact, it has been the most important method to further our understanding of chemistry since more than 100 years. In recent decades, methods have been developed that allow inductive learning to be put on a more formal and rigorous basis by mathematical methods. Different names have been attributed to this area such as machine learning, data mining, pattern recognition, chemometrics, or neural networks. All these methods are considered to be part of chemoinformatics.

There are other reasons that make chemoinformatics indispensable: the amount of information available in chemistry is enormous. Presently, more than 40 million different compounds are known; all have a series of properties, physical, chemical, or biological, all can be made in many different ways, made by a wide range of reactions, all can be characterized by a host of spectra. And each year more than a million new compounds are discovered or synthesized, each year about 800.000 new articles are published that somehow deal with aspects of chemistry. All this just aggravates the flood of information. This immense amount of information can only be processed by electronic means, by the power of the computer. This is again where chemoinformatics comes in! Thus, quite early on, in the sixties, databases for storing information on chemical compounds were built in order to ensure that the information accumulated by chemists can also in the future be made accessible to the scientific community.



Figure(1-2) Deductive and inductive learning: from data through information to knowledge.

Large as this flood of information is, there are also many areas where not enough information is available. Although 40 million compounds are known, we have experimental data on their 3D structure only for 250.000 compounds. And, the largest database on infrared spectra comprises only 220.000 spectra. Thus, we have experimental 3D structures and infrared spectra only for 0.5% of all known compounds. The question is then; can we develop methods to predict the 3D structure or the infrared spectra for the other 99.5% of compounds? Can we learn from the known 0.5% of the 3D structures enough about the construction principles of chemical structures to predict the 3D structures for the other 99.5% of compounds? Can we learn from the 0.5% infrared spectra stored in databases enough about the relationships between structure and infrared spectra to predict

IR spectra for the other 99.5% of compounds? This is again where chemoinformatics has to come in!

Thus, we see that chemistry provides a host of problems to be solved by novel methods: storage and retrieval of chemical compounds and reactions, structure–property relationships, synthesis design, reaction prediction, spectra simulation, structure elucidation. This wide variety of applications has matured to a new field:

Chemoinformatics, the application of informatics methods to the solution of chemical problems (3).

1.2. Applications of chemoinformatics

While not the most well-known part of chemistry, it plays an important part in the delivery of in silico techniques. Cheminformatics was originally defined by Brown in 1998 (4). However; the subject has been in existence far longer. Markush structures were first used in patents from 1924 for describing multiple substituents. Wiswesser line notation, the first line notation to describe complex molecules was created in 1949 (5). The American Chemical Society created the Journal of Chemical Documentation in 1961 which has now morphed into the Journal of Chemical Information and Modeling. It is no longer the only journal dedicated to cheminformatics.

Pivotal to cheminformatics development has been the growth and capabilities of the computer, core to any cheminformatics technique. Cheminformatics consists of several topics, which will be discussed briefly: chemical data storage, substructure searching, similarity searching, clustering, docking and QSAR. Most techniques are available as both 2D and 3D methods. 2D methods are primarily concerned with the topology of molecules. Conformers and stereochemistry are typically ignored un-like with 3D methods. 3D methods are typically more complex in order to model the extra data. Studies have found 2D methods can sometimes outperform 3D counterparts (6). This may sound counterintuitive but simpler methodologies can yield better results with less computational effort.

1.3. Chemometrics

It was recognized early on in the late sixties that the diversity and complexity of chemical data need powerful and diversified data analysis methods. Thus, the field of chemometrics was soon established and is flourishing since, being presented in journals of their own such as Journal of Chemometrics, Journal of Chemometrics and Intelligent

Laboratory Systems, and Quantitative Structure Activity Relationships. Multifaceted as these various problem areas are, from structure representation to chemometrics studies, they have nevertheless drawn success from similar methods, have benefited from many connections to such an extent that they have merged to a scientific discipline of its own: chemoinformatics.

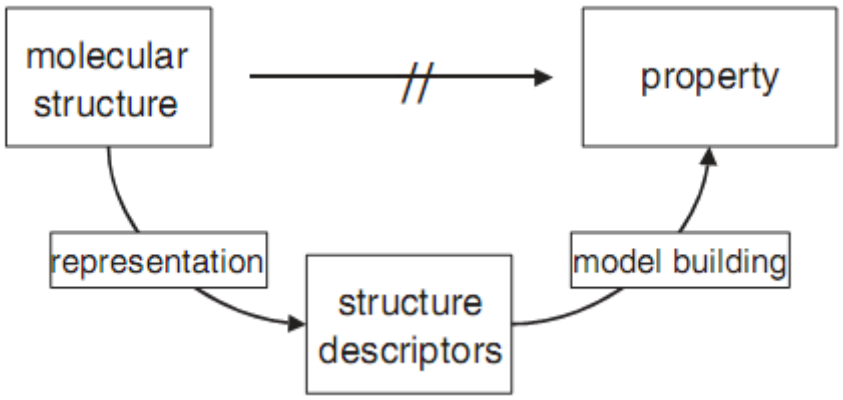
1.3.1. Development of chemometrics

Chemometrics has developed over the past decade from a fairly theoretical subject to one that is applied in a wide range of sciences. The word chemometrics was coined in the 1970s, and the early development went hand in hand with the development of scientific computing, and primarily involved using multivariate statistical methods for the analysis of analytical chemistry data.

1.4. Quantitative structure activity/property relationships

Returning to the fundamental questions of a chemist mentioned in the introduction we want to further delve into the relationships between chemical structure and a desired property. This field, quantitative structure–property relationships (QSPR), or quantitative structure–activity relationships (QSAR) if the property of interest is a biological activity, is the prototypal area of application of chemoinformatics methods as it emphasizes certain problems that are also important in other domains of chemoinformatics.

It has already been emphasized that many properties of a chemical compound, such as its biological activity, cannot be calculated from first principles. This is where inductive learning methods have to come in. Firstly, the chemical structure of compound has to be represented by a set of structure descriptors. Then, a series of compounds and their associated properties have to be compiled and submitted as a training set to an inductive



Figure(1-3) The indirect way for predicting properties of chemical compounds.

learning method to build a model for the relationships between chemical structure and its property Figure (1-3). This process will be analyzed in some detail as it involves methods that are of importance in other areas of chemoinformatics.

1.5. 2D-QSAR methodology

1.5.1. Subset selection in QSAR/QSPR data

Data splitting performed at the initial stage of the QSAR/QSPR development is particularly significant, as it determines, which data are utilized to train (fit) the model, and which are employed for its external validation. The quest to find the most appropriate methodology for selecting training and test set compounds has led to active investigations in this area. A vast range of recently published contributions focused on the importance of data splitting, for example (7; 8; 9), highlight two major conditions that should be met: (i) representivity of both training and test sets and (ii) sufficient diversity of the training set. However, no model, even when properly validated and yielding ‘‘good’’ values of validation statistics, is able to provide reliable predictions for the entire universe of chemicals. The model usually works much better for the compounds falling inside its applicability domain (typically defined by structural/ mechanistic similarity) and the range of activity/property values within the training set. Hence, in the ideal modelling case, chemical structures and the predicted response values for training and test sets should be possibly similar—the representative objects in the training set should be close to the objects in the test set and vice versa (10). In other words, the training and test sets should scatter over the whole range of the considered space, defined by the descriptors of molecular structure (**X**) and the response (**y**) values (11).

In practice, several algorithms are employed to split the input data. The most common ones are based on the endpoint (**y**) values only (e.g. the repeated test set technique, random selection or activity sampling) (12; 13; 14), while more sophisticated techniques take into account also the values of molecular descriptors (**X**) (e.g. maximum dissimilarity method, the Kennard–Stone algorithm, the duplex algorithm, Kohonen’s self-organising maps, D-optimal design or sphere exclusion) (15; 16; 17). Endpoint-value-based methods of data splitting generate even distributions of compounds along with the endpoint values in both created sets. However, there is a danger that the application of such algorithms may be associated with significant loss of information, as the resulting training sets do not necessarily represent the entire descriptor space of the input data. Consequently, the test set compounds may be distant from those included in the training

set. In contrast, algorithms in which **X** values contribute to the data splitting are more likely to generate representative sets consisting of compounds evenly distributed within the chemical space ranged by values of both **y** vector and **X** matrix. Such an approach should ensure the closeness between test and training set compounds (18). Although opinions have been expressed in the academic literature, no firm and practical recommendations related to dataset splitting have been available so far in any of the official guidelines for QSAR/ QSPR modelers.

1.5.1.1. Most descriptive compound (MDC) method

The generation of representative subsets of compounds from chemical databases is an important problem in pharmaceutical research. Subset selection may be necessary to select subsets from smaller databases, for example as training sets for QSAR studies; the subsets are both diverse and representative of the parent database. The "Most Descriptive Compound" (MDC) method has similarities to cluster analysis but avoids many of its ambiguities.

The aim of this algorithm is to select a subset of compounds which most effectively represents the compounds in the original population in terms of the chosen parameters. From a table of the compounds and the autoscaled values of the chosen parameters the following procedure is performed:

a) Calculate the Euclidean distance of compound 1 from all other compounds.

b) Rank the calculated distances from compound 1 to all other compounds and take the reciprocal of the rank. Store this in the information vector I. (e.g. the value of I, is 1 and the closest compound to 1 has I = 1/2, the second closest has I = 1/3 etc.).

These values give a quantitative measure of the information we might expect to get about compound 1 by testing another compound. The advantage of using reciprocal ranks, rather than the distances, is that the transformation from distance to similarity and the associated, arbitrary, choice of a transformation function is unnecessary.

c) Repeat stages a and b for all compounds in the data set and add the reciprocal ranks to I. This now contains a measure of the information we might hope to get on the entire data set by testing any given compound.

d) Find the compound with the largest value in I. This is labeled the most descriptive compound (MDC) and is selected. It corresponds to the compound with the smallest overall distance to all the other compounds.

We next want to select the compound that gives us the greatest additional information on our set of compounds given that the MDC has already been selected. Thus we need a way of removing the contribution of the MDC from I. This is achieved by steps (e) and (f).

e) Recalculate the distances of the MDC to all the other compounds and the reciprocal ranks as above. Subtract these reciprocal ranks from 1 and store in the rank vector R.

f) Multiply values in I by the corresponding values in R. Store the result in I.

Check that all numbers in the I exceed 1. If they do then go to step d and repeat on the new Information vector. Repeat until all values in the I are less than 1 (no more information to extract) or until the required number of compounds have been selected. This termination condition is arbitrary but has been found practically to be a reasonable one (19).

1.5.2. Molecular descriptors

Any computational analysis of chemicals requires a comparison between them. In the early days of QSAR, the common used characterization of molecules was by experimentally obtained descriptors. As these descriptors cannot always be obtained due to experimental difficulty or because one is dealing with virtual molecules, there was great interest in finding other ways to describe these molecules. Nowadays, chemists represent molecules by some form of understandable structural representation. The analysis of molecules is then based on an appropriate selection of features of the chemical - the so-called descriptors - that can be derived from the molecular structure. In the following a brief overview is given of the available descriptors used in 2D-QSAR. Due to the vast number of different methodologies available, this overview is not meant to be a definite reference but rather it tries to give a broad outline of the different techniques and the mathematical principles they are based on. For a more in-depth discussion on the subject of descriptors the reader is referred to the "Handbook of Molecular Descriptors" by Todeschini and Consonni which includes an extensive bibliography of about 3000 references (20).

Constitutional descriptors:

Constitutional descriptors are widely used in QSAR analysis. The descriptor group uses the atomic or molecular properties and is therefore independent of the overall molecular connectivity. They encode the size of molecules and chemical properties. The group of descriptors includes a great variety of descriptors such as molecular mass or refractivity, element count, element quotient and many others. If one accepts experimental

properties as real descriptors, they are often included among these descriptors. These property-based descriptors include many kinds of empirical parameters, e.g. Hammett constants as discussed earlier.

Geometrical descriptors:

These descriptors reflect features of the molecular geometry. Examples of such descriptors include distances between particular points on the molecular surface and distances between given chemical groups. The most widely used geometrical descriptors are molecular surface area and molecular volume. Molecular surface area is the area of the outer surface of the volume from which the solvent molecules are excluded due to the presence of the solute molecule in a solution. It is based on the van der Waals molecular surface - defined by the van der Waals radii of the atoms in the molecule - however, the van Der Waals molecular surface contains small gaps and crevices that are inaccessible to other atoms and molecules. The molecular surface area is defined by excluding these gaps and crevices. The Molecular Polar Surface area is defined as the sum of surface contributions of only the polar atoms (oxygens, nitrogens and attached hydrogens) in a molecule. The calculation of the polar surface area however is relatively time consuming, because of the necessity to create a reasonable 3D molecular geometry and to calculate the surface itself. In order to enable virtual bioavailability screening of very large collections of molecules, a new methodology to calculate the PSA from fragment contributions is used throughout the QSAR analysis. The method of choice is termed TPSA - topological PSA (21; 22).

Topological descriptors:

Topology deals with the type and the connection of atoms in the 2D space. They are evaluated using molecular graph theory which is based on the construction of graphs by replacing atoms with vertices and bonds with edges. A large class of descriptors relies solely on the molecular graph and therefore takes into account only the topology of the molecular graph and discards the chemical information available about the underlying compound. Popular topostructural 2D descriptors include the Wiener index (23), the Zagreb index (24), the Randic connectivity index (25) and the Balaban index (26).

1.5.3. Model derivation and model validation

A QSAR analysis starts with collecting experimental data and generating numerical descriptors of the molecular structures. Both of these datasets need to be transformed at the start of the statistical analysis, to obtain interpretable answers. This topic is covered in the

section Autoscaling. At this stage, the analysis has to cope with a huge set of descriptors. Therefore, the superfluous variables need to be eliminated and the most important descriptors selected. Both techniques are described in the section Variable Elimination and Selection. After these preparative steps, the actual statistical analysis starts. The available methods are described in the section Methods of Regression.

1.5.3.1. Autoscaling

Scaling of data values is important, because then each variable is treated equally. Descriptors with very small values or descriptors with very large values are all treated with equal emphasis. Autoscaling can be interpreted as the translation and the normalization of the descriptor coordinate axes. This can be achieved by mean centering in which the mean of a variable coincides with the origin, and a normalization that makes the length of the data vector equal to unity. The advantage of scaled descriptors is that the magnitude of their coefficients in the regression equation allows the comparison of the relative contribution of each independent variable in the prediction of the dependent variable. As the mean centering only affects the absolute values of the variable and leaves the relative positions unchanged it usually does not have any negative impact on the efficacy of the statistical analysis. Variance scaling is used to normalize each of the descriptors variables to unit variance which ensures that all variables have equal weight in the statistical analysis. However, in some cases the differences in the ranges of variables can act as intrinsic weight factors and the variance scaling that removes them, actually reduces the accuracy of the statistical model.

1.5.3.2. Variable elimination and selection

Due to the nature of QSAR analysis, it is common that there are more descriptors than there are samples. Besides, many descriptors are more or less co-linear, that in turn causes problems for many statistical analysis methods. There are numerous techniques of variable selection. In the context of PLS regression, a review can be found in (27). In the general domain of machine learning, the following taxonomy in three groups is commonly used (28):

With filter methods, variable selection is done independently of the model that eventually makes use of them. Filter methods use the intrinsic characteristics of the whole data set in order to select some variables and/or eliminate others. This selection can be viewed as a pre-treatment of predictive variables. In the field of multivariate calibration,

different filter criteria are used such as the absolute value of correlation or covariance between predictors and response (29). The theory of information is also used for selecting the predictive variables that maximise the mutual information with the variable to be predicted. However this method is difficult to implement when multi-responses are involved. An application in chemomectrics is found in (30). The UVE method (31) allows variable elimination by comparing them with noisy artificial variables.

Wrapper methods scan the space of possible selections and use the prediction model as a black box to test the relevancy of selections. This is often evaluated by means of a simple or cross-validation. Depending on the strategies to perform the scan, there exist different wrapper methods (32). These are in most cases stochastic optimization methods inspired by natural phenomena: Genetic algorithms (33) or simulated annealing (34).

These methods are not repeatable due to their random nature. Moreover, their complex algorithms may pose a problem when the searching space is large and the relevancy of the selection is not easy to assess in the case of multiple responses.

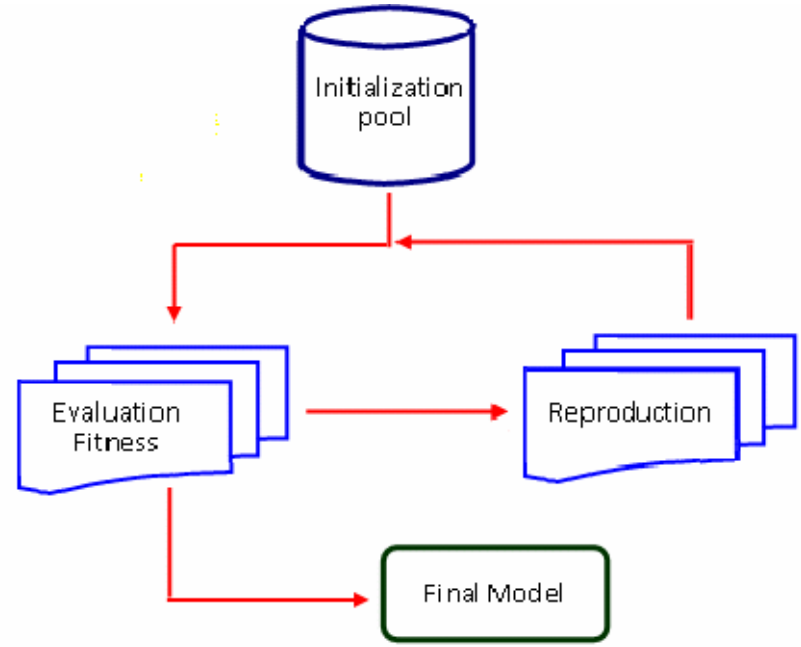
Embedded methods accomplish the variable selection during the calibration process. The subset of selected variables, optimising the training criterion, can be constructed by successive additions (forward), elimination (backward) or a combination of both approaches. Backward methods are not well adapted to the high multivariate cases because, at the beginning of the selection process, they take into account all the variables. Stepwise multiple linear regression (SMLR) (35) is one of the most popular examples of this kind of methods.

Successive Projection Algorithm (36) is a forward selection method that minimises colinearity between predictors by means of successive projections on interlinked sub-spaces. At each step, the selected variable is the one showing the maximum projection on the orthogonal sub-space generated by the already selected variables. SPA is a hybrid between filter and embedded methods.

1.5.3.2.1 Genetic algorithm (GA)

Genetic algorithms are search algorithms inspired by Charles DarwinPs principle of “the survival of the fittest”. GA’s have been used widely in the field of QSAR modeling, cheminformatics and chemometrics (37). The application of genetic algorithms in this work is focused on their use as efficient tools to search large dimensional spaces. More specifically, one application of GA’s in QSAR modeling is to search a descriptor space to find optimal subsets of descriptors that can be used to build predictive models. Figure (1-4)

dedicates a diagram of the generic genetic algorithm and this section describes the steps in detail.



Figure(1-4) A diagram for describing the genetic algorithm

As mentioned above, GA is based on the principles of evolution. As a result much of the terminology from the field of biological evolution has been adapted for use in the field of genetic algorithms. Thus we define an individual as consisting of a chromosome and an associated fitness value. When using a GA for descriptor selection, the chromosome is simply a subset of descriptors (of user specified length) chosen from the descriptor pool that is being searched. A population is defined as a collection of individuals. The first step of the GA is to initialize the population. This is achieved by randomly generating a user specified number (usually 40 to 50) of descriptor subsets of user specified size. Each descriptor subset is used to build a model (which can be a linear regression model or a ANN model). The root mean square error (RMSE) for each model is used to determine the fitness of the individual. The implementation used in this work does not use the raw RMSE value but instead uses a linearly scaled form. The actual form of the fitness function depends on the nature of the model to be developed.

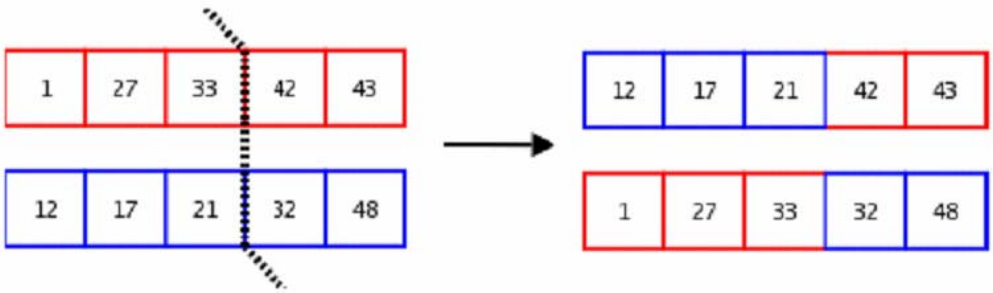
For linear models the fitness for the i^{th} individual in the population is defined as

$$Fitness_i = \left[2 - \frac{RMSE_i}{RMSE_{avg}}\right]^{-1} \quad i^{th} \qquad (1.1)$$

where $RMSE_i$ is the RMSE for the i^{th} individual and $RMSE_{avg}$ is the average RMSE for the whole population.

The next step is to create a child population. First a mating list is created, which is of the same size as the current population. Those individuals with fitness greater than the population average (which from equation 1.1 is greater than 1.0) are automatically placed in the mating list. By definition, this will fill up half of the available slots. The remaining slots in the mating list are filled by using a roulette wheel selection procedure to select individuals from the current population. Once the mating list is created a child population is then generated by successively selecting two individuals from the mating list at random and applying genetic operations. The first operation is termed crossover, and involves the swapping of portions of the chromosomes of a pair of individuals. The GA literature describes a number of variations of the crossover operation (38). The current implementation restricts itself to the single point crossover. In this type of crossover a split point is chosen in the descriptor subset.

Then the descriptors from one side of the split point in the two individuals are swapped to give rise to two new individuals. This operation is shown graphically in Figure 1-5. The figure represents a crossover performed on two individuals having a chromosome (descriptor subset) of length 5.



Figure(1-5) A schematic diagram of the single point crossover operation. The grids on the left represent the parents and the grids on the right represent the children formed after crossover. The portions of the chromosomes to the left of the split point are swapped.

The split point is chosen at the fourth descriptor and the descriptors on the left of the split point are swapped resulting in two new individuals. The goal of crossover is to generate new individuals that will have the good features of the parent individuals. That is, if two individuals have a high fitness this implies that certain parts of their chromosomes (i.e., certain descriptors) are responsible for their fitness. By combining a portion of the chromosomes of two fit individuals, we expect that the children will exhibit equal if not better fitness.

The second genetic operation is termed mutation and is performed on a single child

individual. It should be noted that mutation is not performed on all individuals in a population but is carried out only 5% of the time, mirroring the low frequency of mutation in biological evolution. In a genetic algorithm the mutation operation is performed by randomly changing a part of the chromosome of an individual. That is, a random descriptor within an individual is replaced with a randomly chosen descriptor from the descriptor pool. This is shown schematically in Figure (1-6). The goal of the mutation operation is twofold. First, random mutations prevent the algorithm from getting stuck in a local minimum and second, mutations prevent the phenomenon of premature convergence. This occurs when the algorithm creates very similar (or even identical) individuals whose fitness is high, but not necessarily optimal.

Figure (1-6) A schematic diagram of the mutation operation

The mutation operation can also be viewed as a method to maintain diversity within a population, though this does not entirely solve the problem of premature convergence as noted by Goldberg (38). With the application of these two operations we end up with a second, child, population. The fitness of the individuals in this population is evaluated and the individuals ranked. The second generation population is then created by randomly selecting individuals from the top 50% of the previous population and the child population. Finally, if the best model in the child population is of lower fitness than the best model from the previous population, the best model from the previous population is kept in the second generation. With the formation of the second generation population, the whole process is repeated. This continues for a user specified number of cycles (usually 1000) and at the end the top ranked individuals (i.e., the top ranked descriptor subsets and associated RMSE values) are reported to the user. Genetic algorithms (GA) are a general methodology for searching a solution space in a manner analogue to the natural selection procedure in biological evolution.

1.5.3.2.2. Stepwise regression method

Groups of commonly used regression methods are proposed to evaluate only a small number of subsets by either adding or deleting variables one at a time according to a

specific criterion. The forward selection method adds variables to the model one at a time. The first variable included in the model is the one which has the highest correlation with the independent variable *y*. The variable that enters the model as the second variable is one which has the highest correlation with *y*, after *y* has been adjusted for the effect of the first variable. This process terminated when the last variable entering the model has insignificant regression coefficient or all the variables are included in the model.

In contrast to forward selection, backward elimination begins with the full model and successively eliminates one at a time. The first variable deleted is the one with the smallest contribution to the reduction of predictive error sum of squares (PRESS).

Assuming that there are more variables that are insignificant, the process operates by eliminating the next most insignificant variable. The process is terminated when all the variables are significant or all but one variable has been deleted. In stepwise procedure a variable that entered the model in the earlier stages of selection may be deleted at the later stages. The calculations made for inclusion and elimination of variables are the same as forward selection and backward procedures. That is, the stepwise method is essentially a forward selection procedure, but at each stages the possibility of deleting a variable, as in backward elimination, is considered. The number of variables retained in the model is based on the levels of significance assumed for inclusion and exclusion of variables from the model.

One common problem in multiple regression analysis is multicollinearity of the input variables. The input variables may be as correlated with each other as they are with the response. If this is the case, the presence of one input variable in the model may mask the effect of another input. Stepwise regression used as a canned procedure is a dangerous tool because the resulting model may include different variables depending on the choice of starting model and inclusion strategy (39).

1.5.3.3. Method of regression

In Quantitative Structure-Activity relationships, molecular descriptors collected in the design matrix (**X**) are correlated with a response variable collected in the dependent column vector (**y**). The y values are assumed to be linearly dependent on the independent variables contained in the **X** matrix. The objective of the analysis usually is to increase the understanding of the biological system under investigation or to predict the response of objects not yet tested (e.g., predict the potency of a compound not yet synthesized). The conclusions drawn from a regression analysis are dependent on the assumptions on the

regression model. If it is assumed that the relationship is well represented by a model that is linear in the descriptors, a suitable model may be represented by

$$\boldsymbol{y} = b_0 + b_1 * \boldsymbol{X}_1 + \cdots + b_k * \boldsymbol{X}_k + e \quad (1.2)$$

In equation 1.2 the b's are unknown constants called regression coefficients and the objective of regression analysis is to estimate these constants.

1.5.3.3.1. Multiple linear regressions (MLR)

In order to establish a relationship between **X** and **y** in Figure (1-7), Multiple Linear Regression (MLR) (39) has until recently been the obvious method of choice. In MLR, it is assumed that **X** is of full rank and the x_{ij} are measured with negligible error. The algebraic MLR model is defined in Equation 1.2 and in matrix notation:

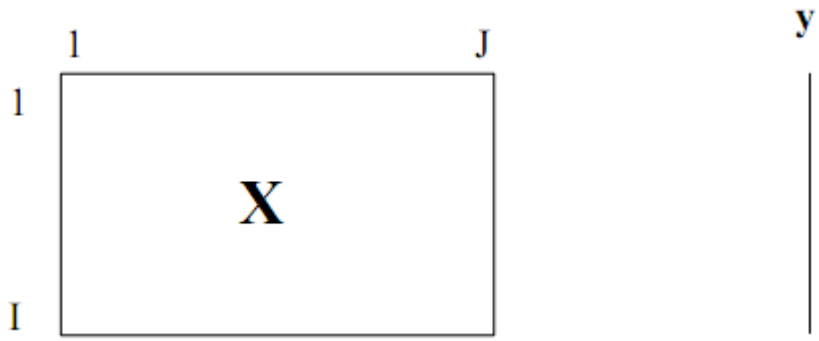
$$\boldsymbol{y} = \boldsymbol{X}b + e \quad (1.3)$$

where $\mathbf{X} = [\mathbf{x}_0|\mathbf{x}_1|\ldots\mathbf{x}_J]$, $\mathbf{b}^T = [b_0,b_1,\ldots,b_J]$ and **e** is an error vector. Note that the first column in **X**, *i.e.*, $\mathbf{x}_0$ consists of only constants which, after mean-centering, become zero and consequently $\mathbf{x}_0$ is omitted. When **X** is of full rank the least squares solution is:

$$\hat{b} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (1.4)$$

where $\hat{b}$ is the estimator for the regression coefficients in b. An obvious disadvantage using MLR as regression method in QSAR is: when $I \leq J$ Figure (1-7) **X** is not of full rank and $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ in Equation 1.4, is not defined and b cannot be estimated. In the following section the problem with multicollinearity (39), *i.e.* the case when **X** not is of full rank, will be discussed.

The stepwise multiple linear regression is a commonly used variant of MLR. In this case, also a multiple-term linear equation is produced, but not all independent variables are used. Each variable is added to the equation at a time and a new regression is performed. The new term is retained only if the equation passes a test for significance. This regression method is especially useful when the number of variables is large and when the key descriptors are not known. However, if the number of variables exceeds the number of structures, alternative methods such as projection methods should be considered. As a consequence, MLR can be used but a careful selection within the set of available descriptors has to be performed as proposed in the previous section.



Figure(1-7) A typical QSAR data set: X is of the dimensions *I* ×*J* where *J* > *I* with a single response variable y (*I* ×1).

1.5.3.3.2. Support vector machine (SVM)

The Support vector machine (SVM) is a machine-learning technique for classification that involves a non-linear mapping of data into a high-dimensional feature space, then using structural risk management to find a separating hyperplane with the largest margin between the transformed data. These learning machines have been shown to classify with accuracy at least as good as the various neural net methods.

While SVMs achieve excellent predictive power, they are not simple to interpret, and little work has been done in this area (40). They are popular in a variety of disciplines as they perform well on various data sets. The drawback of this method is the models build time, due to the quadratic programming step of the algorithm for building a SVM. By nature they avoid local minima, thus aiding predictive power. Like most classifiers, researchers have modified SVMs to improve them. Most of this work has been focused on the kernel, either creating new or modifying the common radial basis function (RBF) or Polynomial kernels. Vapnik is credited with the original work on SVMs (41). SVMs work effectively on both linear and non-linear problems.

In support vector regression (SVR), the basic idea is to map the data x into a higher-dimensional feature space F via a nonlinear mapping Φ, and then to do linear regression in this space. Therefore, regression approximation addresses the problem of estimating a function based on a given data set G= $\{(x_i,d_i)\}i^n$ (x_i is the input vector, d_i is the desired value, and n is the total number of data patterns). SVM approximate the function using the following:

$$y = f(x) = w\Phi(x) + b \qquad (1.5)$$

where Φ(x) denotes the element wise mapping from x into feature space. The coefficients w and b are estimated by minimizing

$$R_{SVMs}(C) = C\frac{1}{n}\sum_{i=1}^{n} L_e(d_i, y_i) + \frac{1}{2}||w||^2 \quad (1.6)$$

$$L_\varepsilon(d, y) = \begin{cases} |d-y| - \varepsilon & |d-y| \geq \varepsilon \\ 0 & otherwise \end{cases} \quad (1.7)$$

In Equation 1.6, *RSVMs* is the regularized risk function, and the first term $C\frac{1}{n}\sum_{i=1}^{n} L_e(d_i, y_i)$ is the empirical error (risk). They are measured by the ε-insensitive loss function (L_ε) given by Equation 1.7. This loss function provides the advantage of enabling one to use sparse data points to represent the decision function given by Equation 1.5. The second term $\frac{1}{2}||w||^2$, on the other hand, is the regularization term. C is referred to as the regularized constant, and it determines the tradeoff between the empirical risk and the regularization term. Increasing the value of C will result in the relative importance of the empirical risk with respect to the regularization term to grow. ε is called the tube size, and it is equivalent to the approximation accuracy placed on the training data points. Both C and ε are user-prescribed parameters.

Finally, by introducing Lagrange multipliers (a_i,a_i^*) and exploiting the optimality constraints, the decision functiongiven by Equation 4 has the following explicit form:

$$f\left(x, a_i, a_i^*\right) = \sum(a_i - a_i^*)K(x, x_i) + b \quad (1.8)$$

Based on the Karush-Kuhn-Tucker (KKT) conditions of quadratic programming, only a number of coefficients (a_i ,a_i^*) will assume nonzero values, and the data points associated with them could be referred to as support vectors. In Equation 1.8, the kernel function K corresponds to $K(x,x_i) = \Phi(x).\Phi(x_i)$. One has several possibilities for the choice of this kernel function, including linear, polynomial, splines, and radial basis function. The elegance of using the kernel function lies in the fact that one can deal with feature spaces of arbitrary dimensionality without having to compute the map Φ(x) explicitly. The overall performances of SVM models were evaluated in terms of root mean square error (RMSE), which was defined as below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_S}(y_k - \hat{y}_k)^2}{n_s}} \quad (1.9)$$

where y_k is the desired output, $\hat{y}_k$ is the predicted value and ns is the number of samples in the analyzed set.

1.5.3.4.Model validation

Since the real utility of a QSAR model is its ability to accurately predict the modeled property for new chemicals and to interpret the model from the point of view of influential descriptors, a realistic assessment of the power of the model is necessary for a confident application.

1.5.3.4.1. Measure of goodness of fit

To assess the goodness-of-fit, the coefficient of multiple determinations is used. R^2 estimates the proportion of the variation in the response that is explained by the predictor.

$$R^2 = 1 - \frac{\sum_{i=1}^{I}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{I}(y_i - \bar{y})^2} \qquad (1.10)$$

where y_i is the observed dependent variable, $\bar{y}$ the mean value of the dependent variable and $\hat{y}$ the calculated dependent variable. If there is no linear relationship between the dependent variable and the descriptors then R^2 = 0.00; if there is a perfect fit then R^2 = 1.00. R^2 values higher than 0.50 indicate that the explained variance by the model is higher than the unexplained one. The end-user(s) of a QSAR model should decide what value of R^2 is sufficient for the specific application of the model. The value of R^2 can generally be increased by adding additional predictor variables to the model, even if the added variable does not contribute to reduce the unexplained variance of the dependent variable. It follows that R^2 should be used with caution. This can be avoided by using another statistical parameter - the so-called adjusted R^2 (R^2_{adj}).

$$R^2_{adj} = 1 - (1 - R^2)(\frac{I-1)}{(I-k)} \qquad (1.11)$$

R^2_{adj} is interpreted similarly to the R^2 value, except that it takes into consideration the number of degrees of freedom. The value of R^2_{adj} decreases if an added variable to the equation does not reduce the unexplained variance.

1.5.3.4.2. Measure of the validity of the model

A necessary condition for the validity of a regression model is that the multiple or squared correlation coefficient R^2 is as close as possible to one and the standard error of the estimate se is small. However, these conditions measure how well the model is able to mathematically reproduce the end point data of the training set. But they are an insufficient condition for model validity, as they do not express the ability of the model to make reliable predictions on data outside the training set. Therefore, extra conditions need to be