Drug Design with Machine Learning

OVIDIU IVANCIUC
Department of Biochemistry and Molecular Biology,
University of Texas, Medical Branch, Galveston, USA

Article Outline

Glossary

Bayesian classifier Bayes' theorem of conditional probability is a method of statistical inference that represents the basis of several classification ML used in drug design and chemoinformatics to classify libraries of compounds into active and inactive. A Bayesian classifier considers each structural feature or descriptor independent of the other descriptors, and the probability that a compound is active is proportional to the ratio of active to inactive compounds that have the same structural feature or have the same value for that descriptor. The final probability that a compound is active is a product of all descriptor-based probabilities. Structural descriptors that are real numbers are usually binned prior to their evaluation with a Bayesian classifier.

Decision tree A decision tree is a sequence of rules applied to selected structural descriptors. The training phase comprises the selection of the structural descriptors that are evaluated, the order in which the rules are applied, and the decision taken at each leaf. Usually, each rules evaluates a descriptor ($\geq$ or $<$ than a threshold) and splits the objects into two or more populations. Then each population is selected and the splitting procedure is performed with a new rule, until a stopping condition is met (for example, when all objects in the population belong to the same class). The prediction phase starts from the root node and evaluates each rule on a pathway determined by the outcome (true or false) of the previous rule. When a leaf is reached the algorithm predicts the class of the object (classification trees) or the numerical value of a property (regression trees).

***k*-nearest neighbors** *k*-nearest neighbors (*k*-NN) is a supervised learning algorithm that predicts the property of an object based on a local interpolation model. In classification, the class of a new object is predicted based on the majority vote of its *k* nearest neighbors. In regression, the property value for a new object is predicted as an average value of the property values for its *k* nearest neighbors.

Lazy Learning Lazy learning is a memory based local learning that defers the computation until a prediction is requested for an object. The first step is to insert the query object into the space of the training objects, and to identify the training objects located in a set neighborhood. The predicted property of the query object is based on an interpolation of the properties of the objects situated in the neighborhood.

Machine learning Machine learning is an important field of artificial intelligence, and includes a diversity of methods and algorithms that extract rules and functions from large datasets, such as decision trees, lazy learning, *k*-nearest neighbors, Bayesian methods, Gaussian processes, support vector machines, and kernel algorithms. Machine learning algorithms extract information from experimental data by computational and statistical methods and generate a set of rules, functions or procedures that allow them to predict the properties of novel objects that are not included in the learning set.

Quantitative structure-activity relationships Quantitative structure-activity relationships (QSAR) represent regression models that define quantitative correlations between the chemical structure of molecules and their physical properties (boiling point, melting point, aqueous solubility), chemical properties and reactivities (chromatographic retention, reaction rate), or biological activities (cell growth inhibition, enzyme inhibition, lethal dose). The fundamental hypotheses of QSAR is that similar chemicals have similar properties, and small structural changes result in small changes in property values. The general form of a QSAR equation is $P(i) = f(\mathbf{SD}_i)$, where $P(i)$ is a physical, chemical, or biological property of compound i, $\mathbf{SD}_i$ is a vector of structural descriptors of i, and f is a mathematical function such as linear regression, partial least squares, artificial neural networks, or support vector machines. A QSAR model for a property P is based on a dataset of chemical compounds with known values for the property P, and a matrix of structural descriptors computed for all chemicals. The

learning (training) of the QSAR model is the process of determining the optimum parameters of the regression function f. After the training phase, a QSAR model may be used to predict the property P for novel compounds that are not present in the learning set of molecules.

Support vector machines Support vector machines (SVM) are a class of supervised machine learning methods based on the structural risk minimization and the statistical learning theory of Vapnik. SVM may be applied to data classification and regression, using selected objects (support vectors) to generate the SVM model. Nonlinear classification problems are transformed into linear classification problems by using kernel functions that combine the input space into a higher-dimensional feature space in which a hyperplane may discriminate the classes. An SVM classification model computes a maximum margin hyperplane that separates the classes in the feature space. The maximal margin hyperplane maximizes the distance to the hyperplane of the closest patterns from the two classes. An SVM regression model builds a regression tube with the property that all objects inside the tube do not contribute to the overall error of the model. The shape of the regression tube is determined by selected objects (support vectors) situated outside the tube.

Structural descriptor A structural descriptor (SD) is a numerical value computed from the chemical structure of a molecule, which is invariant to the numbering of the atoms in the molecule. Structural descriptors may be classified as constitutional (counts of molecular fragments, such as rings, functional groups, or atom pairs), topological indices (computed from the molecular graph), geometrical (volume, surface, charged-surface), quantum (atomic charges, energies of molecular orbitals), and molecular field (such as those used in CoMFA, CoMSIA, or CoRSA).

Structure-activity relationships Structure-activity relationships (SAR) represent classification models that can discriminate between sets of chemicals that belong to different classes of biological activities, usually active/inactive towards a certain biological receptor. The general form of a SAR equation is $C(i) = f(\mathbf{SD}_i)$, where $C(i)$ is the activity class of compound i (active/inactive, inhibitor/non-inhibitor, ligand/non-ligand), $\mathbf{SD}_i$ is a vector of structural descriptors of i, and f is a classification function such as k-nearest neighbors, linear discriminant analysis, random trees, random forests, Bayesian networks, artificial neural networks, or support vector machines.

Definition of the Subject

The process of drug discovery has the goal to identify lead chemicals that have a significant activity against a selected biological target. A disease state may be the result of changes in the structure and function of cell-signaling receptors, enzymes, hormone receptors, or other functional proteins. The drug target is a protein whose activity is modulated by its interaction with a chemical compound, and thus may control a disease. The lead compounds identified in the drug discovery step are optimized in the drug development phase that results in a small number of chemicals that are evaluated in human clinical trials. The first priority in drug development is to increase the biological activity of a lead compound while preserving its drug-like properties. The lead compound is expanded into a chemical library that conserves the structure responsible for the biological activity (pharmacophore) and adds chemical groups that might improve its activity. Then the chemicals are synthesized and tested in biological assays against the target, which may result in the identification of more active compounds. The cycle "library design – chemical synthesis – biological assay" is repeated several times until a number of good candidates are identified for clinical trials. The design of effective drug candidates is a multi-objective optimization problem, because simultaneous with a good biological activity the chemicals must pass several other important tests, including pharmacokinetics, pharmacodynamics, toxicity, mutagenicity, metabolism, and excretion.

Computer-assisted drug design (CADD) uses computational chemistry to increase the chances of finding valuable drug candidates. CADD methods include machine learning, structure-activity relationships (SAR), quantitative structure-activity relationships (QSAR), molecular mechanics, quantum mechanics, molecular dynamics, and drug-protein docking. SAR and QSAR are based on the theory that chemical structure determines all physical, chemical and biological properties of a molecule. To obtain structure-activity models, the chemical structure is characterized with structural descriptors and then ML models are used to identify a statistical relationship between the descriptor space and a molecular property. Other considerations for the application of SAR and QSAR in drug design are that similar molecules have similar properties, and that a small modification in a chemical structure result in a small modification in its properties.

Machine learning (ML) procedures are applied in drug discovery in an iterative way, in which experimental activity data are used to train ML models which in turn offer predictions for novel molecules with improved bi-

ological properties. Then the molecules designed with ML are tested in biological assays, and the best candidates are selected for further cycles of optimization. ML models may greatly improve the chances of finding lead molecules by screening a larger diversity of chemical topologies. In the classical approach, a number of chemicals in the range 10^3–10^4 is synthesized and assayed to identify lead compounds, whereas the introduction of combinatorial chemistry and high-throughput screening (HTS) increased these numbers to 10^5–10^6. Although the robots offer great speed and reliability in HTS, the chemical compounds still have to be synthesized before screening, thus limiting the diversity of structures evaluated in HTS. This is why ML models are used to enhance the lead identification process with a virtual HTS (vHTS) screening of 10^7–10^8 molecules. Based on experimental results from a HTS campaign, ML models are computed and then used in vHTS experiments. These ML are mainly classification models that predict molecules that have a high probability of interacting with the selected target. vHTS has several obvious advantages compared to HTS, mainly coming from the fact that the chemical compounds are generated only in a computer, which opens the possibility to explore a larger diversity of chemical skeletons and a much higher number of molecules.

Classification (SAR) and regression (QSAR) ML models are applied during the drug development cycles to optimize the biological activity, target selectivity, and other physico-chemical and biological properties of selected chemicals. ML models are essential tools in increasing the chances of bringing a drug to market, but only when integrated with the usual chemical, biological, pharmacological and clinical procedures used by the pharmaceutical industry. Boid enumerated several successful CADD applications in which computational methods had a decisive contribution to the discovery of a drug, namely norfloxacin (Merck, antibacterial), losartan (Merck, antihypertensive), dorzolamide (Merck, antiglaucoma), ritonavir (Abbott, antiviral), indinavir (Merck, antiviral), donepezil (Esai, anti-Alzheimer's disease), zolmitriptan (AstraZeneca, antimigraine), nelfinavir (Pfizer, antiviral), amprenavir (GlaxoSmithKline, antiviral), zanamivir (GlaxoSmithKline, antiviral), oseltamivir (Roche, antiviral), lopinavir (Abbott, antiviral), imatinib (Novartis, antineoplastic), erlotinib (OSI, antineoplastic) [17]. This is an impressive list that adds convincing evidence to advocate the integration of machine learning and other chemoinformatics methods in drug discovery and development.

Introduction

Machine learning is an important field of artificial intelligence, and includes a diversity of methods and algorithms that extract rules and functions from large datasets, such as linear discriminant analysis (LDA), artificial neural networks (ANN), decision trees, lazy learning, *k*-nearest neighbors, Bayesian methods, Gaussian processes, support vector machines (SVM), and kernel algorithms. Several influential books are recommended for a detailed overview of ML algorithms: *Pattern Recognition and Neural Networks* by Ripley [110], *Machine Learning and Data Mining* by Kononenko and Kukar [83], *Pattern Recognition and Machine Learning* by Bishop [15], *Data Mining: Practical Machine Learning Tools and Techniques* by Witten and Frank [151], *Introduction to Machine Learning* by Alpaydin [4], *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman [52], *Pattern Classification* by Duda, Hart, and Stork [35], *Machine Learning* by Mitchell [96], and *Neural Networks for Pattern Recognition* by Bishop [16].

ML algorithms extract information from experimental data by computational and statistical methods and generate a set of rules, functions or procedures that allow them to predict the properties of novel objects that are not included in the learning set. In drug design the task is to learn how chemical structure determines some important drug property, such as physico-chemical properties (aqueous solubility, skin penetration, hydrophobicity, intestinal absorption, blood-brain barrier penetration), biological activity (enzyme inhibition), metabolism, toxicity, mutagenicity, or excretion. Each molecule is associated with a set of structural descriptors and an experimental property. The structural descriptors are used as input for an ML algorithm and the property is the target (output) of the model. A structural descriptor is a structural feature of the chemical structure, and can be a list of substructures or substituents, graph descriptors [19,64,136], topological indices [8,18,65], connectivity indices [74,75] electrotopological indices [76], or indices derived from the molecular geometry and quantum calculations [73,133]. The experimental property that is modeled may be a class label $(+1/-1)$, such as soluble/non-soluble, inhibitor/non-inhibitor, ligand/non-ligand, toxic/non-toxic, mutagen/non-mutagen, carcinogen/non-carcinogen. For this type of data one obtains a classification model or SAR. Classification models are mainly used to filter large chemical libraries and reduce them a small number of chemicals with a high probability of having a desired property (ligand for a biological target, inhibitor for an enzyme, non-toxic, non-mutagen, or non-

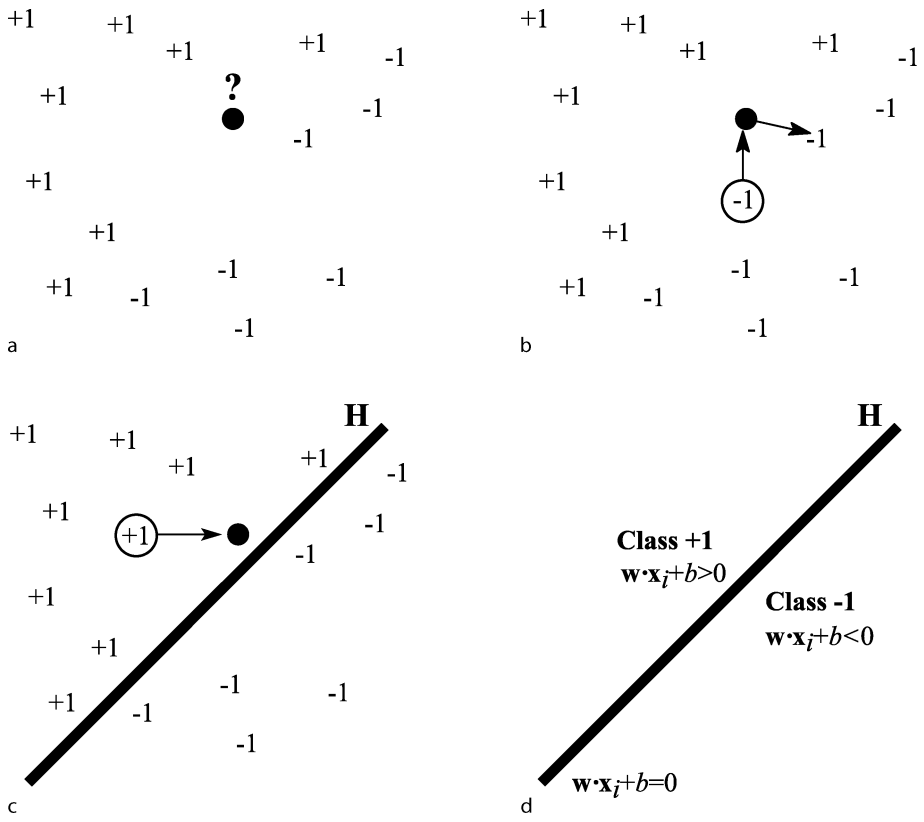
carcinogen). The experimental property may be also a continuous value, such as inhibition constant to an enzyme, binding constant to a target, hydrophobicity, aqueous solubility, skin penetration, or intestinal absorption, in which case one computes a regression model or QSAR. Regression models, which may be linear or non-linear, are used to optimize the drug-like properties of a chemical compound. A drug-related property may be considered categorical variable or a continuous variable, depending on the drug development phase in which is applied. For example, in developing CNS (central nervous system) drugs, the blood-brain barrier penetration is used as a categorical variable in the early stages of drug discovery, to identify and eliminate compounds that do not pass the blood-brain barrier. In later stages of drug development this property is used as a continuous variable in QSAR models to optimize the brain concentration of a small number of selected chemicals that show promising pharmacological properties. Both SAR and QSAR models belong to the class of supervised learning algorithms, because the target (output) value is provided together with the input chemical structures and descriptors. Supervised learning algorithms have two distinct phases. The first one is learning or training, in which a ML method is used to learn the relationships between input (structural descriptors) and output (an experimental property provided for each molecule). The result is a model that can be a statistical function or a set of rules. The second phase of a ML algorithm is the prediction, when the trained model is used to predict the property for novel chemicals that were not present in the training set or that are not even synthesized. The predictions obtained are then used as a guide in synthesizing and testing novel chemicals.

In unsupervised learning the dataset contains only chemical structures and structural descriptors, without output values. The ML objective is to identify how molecules form clusters based on their structural similarity. Obviously, starting from a particular set of structural descriptors, different similarity indices and different clustering algorithms will result in different clusters. Clustering algorithms are applied in drug design to identify similar molecules in chemical libraries. For example, starting from a lead compound one can find similar compounds in catalogs and databases, and thus purchase chemicals that might be useful in the drug discovery process. Another application is to identify which chemical structures are under-represented or missing from a company collection, and to guide compound synthesis and acquisition.

To illustrate two basic approaches in data classification we consider the objects from Fig. 1a. The two classes of objects may represent two populations of molecules, for example enzyme inhibitors form class +1, whereas non-inhibitors form class −1. From the distribution of the two classes one can identify a cluster for class +1 and a distinct cluster for class −1. Such a clear-cut situation is not the norm in drug design application, because classes of molecules may overlap which makes more difficult the property prediction for novel chemicals. The problem considered in Fig. 1a is to use the information provided by the two classes of objects to predict the class of the unknown object marked with "?". The first ML tested is *k*-nearest neighbors (*k*-NN), that belongs to the class of lazy learning algorithms, and performs a local approximation of the model that is computed only in the moment of the prediction. An object is classified in the most populated class among its *k* nearest neighbors, where for two-class problems *k* must be odd to avoid undecided situations. We test here the simplest variant, 1-nearest neighbor, which considers only the closest neighbor. An inspection of the distances between the prediction object • and all other objects shows that the closest object belongs to class −1, and thus the prediction object is assigned to class −1 (Fig. 1b). The predicted class may change if one considers a larger number of nearest neighbors, and the optimum value for *k* is usually determined through cross-validation.

An efficient procedure to classify linearly separable classes is represented by a hyperplane H (Fig. 1c) that separates the descriptor space into a region for class +1 and another region for class −1. A new object that is located in the space region that belongs to class +1 will be assigned to class +1 irrespective of its distance to the separating hyperplane. Similarly, if the new object is situated in the space region that belongs to class −1, then the object is predicted in class −1. The unknown object from Fig. 1a is assigned to class +1 by the hyperplane H (Fig. 1c). For the same classification problem we obtained two different predictions for the unknown object, namely class −1 with 1-NN and class +1 with a hyperplane. Such divergent predictions obtained with different ML methods are quite common, and it is not unusual to obtain different predictions for the same object just by changing the ML parameters. These practical aspects of ML should receive proper consideration, and it is a good practice to evaluate a large diversity of ML methods to obtain consensus predictions.

The two ML demonstrated here are very different in their formulation and their properties. *k*-NN can be used for multi-class discrimination and it is a nonlinear classifier that is useful for classes that cannot be separated with a linear hypersurface. Also, *k*-NN does not produce a statistical function to replace the data, and the learning phase is missing. The second classifier replaces the data with the equation of a hyperplane (Fig. 1d). This function may be



Drug Design with Machine Learning, Figure 1
Classification with machine learning: a a class prediction (+ 1/—1) is sought for the object •; b a *k*-nearest neighbor classifier ($k = 1$) predicts the object • in class —1; c a linear classifier defined by the hyperplane *H* predicts the object • in class + 1; d the classification hyperplane *H* separates the space in region for class + 1 and another region for class —1

used only for two-class problems in which the objects may be separated by linear hypersurface. A hyperplane H has the formula $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ is the normal vector to H. A molecule i characterized by a vector of structural descriptors $\mathbf{x}_i$ belongs to class + 1 if $\mathbf{w} \cdot \mathbf{x}_i + b > 0$, or to class −1 if $\mathbf{w} \cdot \mathbf{x}_i + b < 0$. The rules for predicting the class for an unknown object k are:

$$\text{class}(k) = \begin{cases} +1 & \text{if } \boldsymbol{w} \cdot \boldsymbol{x}_k + b > 0 \\ -1 & \text{if } \boldsymbol{w} \cdot \boldsymbol{x}_k + b < 0\,. \end{cases} \quad (1)$$

The hyperplane classifier may be extended with kernel functions to nonlinear classification, as it will be shown in Sect. “Comparative Studies” in which we give an overview of support vector machines and their applications in structure-activity models.

A decision tree performs a mapping of the structural descriptors to a conclusion about a property of the object. The learning process establishes a series of successive rules that are based on the numerical values of a subset of descriptors. Rules are applied in a set sequence and at each branching point a rule is evaluated and its outcome (true or false) determines which branch is selected in the next step. Staring from the root node, the rules are evaluated and depending on the actual numerical values of the descriptors the pathway ends on a leaf that assigns a class label or a value to the investigated property. The class of decision tree algorithms includes the C4.5 decision tree [108], the NBTree decision tree with naïve Bayes classifiers at the leaves [82], the alternating decision tree ADTree [42], random trees and random forests [22]. Decision trees are computationally very efficient, and their predictions are usually better than those obtained with more complicated ML algorithms. Drug design applications of decision trees include diverse applications, such as prediction of drugs that cross the blood-brain barrier [30], structure-activity relationships for estrogen receptor ligands [135], prediction of *P*-glycoprotein substrates [91], evaluation of the cytochrome P450 metabolism [154], identification of drug-like chemicals [114], design of target chemical libraries [31], prediction of protein-protein interaction [98], and modeling the

oral absorption of drugs [57]. A review of these studies is presented in Sect. "Decision Trees".

Lazy learning is a memory-based local learning method that stores the training objects and performs all computations only when a prediction request is received by the system [5,6]. The prediction is performed by inserting the unknown object into the prediction space and then identifying all objects situated in its neighborhood. Each prediction is made by computing a local model from the neighboring objects. Lazy learning may be used both for classification and regression, with predictions based on local interpolations of selected objects according to a distance measure. Local learning may perform better than global learning, as shown in a number of SAR and QSAR studies that use various techniques of lazy learning [47,85,159].

The *k*-nearest neighbors (*k*-NN) algorithm is a lazy learning method that can be used for classification and regression [1]. The training set of objects is stored in the memory, and then each prediction is made by computing a local interpolation model. The prediction object is added to the descriptor space in which the training objects are placed and then its *k* nearest neighbors are identified. For classification, the class of the prediction object is assigned to class of the majority of its neighbors. Ties are avoided by selected an odd value for *k*, and the best value for *k* is usually determined by cross-validation. For regression, the predicted property for the query object is an average value of the property values for its *k* nearest neighbors. A distance weighting may be added in such a way to increase the contribution of closer objects and to decrease the influence of more distant objects. Another lazy learning algorithm is K^* which uses an entropy-based distance function [26]. Basak was an early proponent of the *k*-NN regression for physico-chemical and biological properties [10,11,49]. The method was further refined by Tropsha by optimizing the descriptor selection, *k*, and the distance weighing function [60,160,161]. Other *k*-NN applications were proposed for three-dimensional QSAR [2] and for ligand-based virtual screening of chemical libraries [54]. In Sect. "Lazy Learning and *k*-Nearest Neighbors" we review SAR and QSAR models computed with lazy learning and *k*-nearest neighbors.

Bayes' theorem provides mathematical tools that explain how the probability that a theory is true is affected by a new piece of evidence [12]. Several Bayesian classifiers [71,151] were proposed to estimate the probability that a chemical is active based on a set of descriptors and a chemical library with known class attributes (active/inactive). Bayesian classifiers consider that each descriptor is statistically independent of all other descriptors. Binary descriptors, structural counts, fingerprints and other descriptors that have integer values are directly evaluated with Bayesian classifiers, whereas real number descriptors are discretized and transformed into an array of bins. The probability that a compound is active is proportional to the ratio of active to inactive compounds that have the same structural feature or have the same value for that descriptor. A final probability is computed as the product of all descriptor-based probabilities. Bayesian classifiers have numerous applications in chemoinformatics and structure-activity models, such as for the prediction of multidrug resistance reversal [125], to estimate the phospholipidosis inducing potential [105], to identify chemical similar to natural products [39], to improve high-throughput docking [78,79], and to screen virtual chemical libraries [77]. Drug design applications of Bayesian methods are presented in Sect. "Bayesian Methods".

Support vector machines belong to a heterogeneous group of machine learning methods that use kernels to solve efficiently high-dimensional nonlinear problems. SVM extend the generalized portrait algorithm developed by Vapnik [144] by using elements of statistical learning theory [143] that describe the ML properties that guarantee dependable predictions. Vapnik elaborated further the statistical learning theory in three more recent books, *Estimation of Dependencies Based on Empirical Data* [138], *The Nature of Statistical Learning Theory* [139], and *Statistical Learning Theory* [140]. Vapnik and co-workers developed the current formulation of the SVM algorithm at AT&T Bell Laboratories [20,25,28,33,50,116,141,142].

SVM models have several interesting and appealing properties, namely the maximum margin classification, the kernel transformation of the input space into a feature space where a hyperplane may separate the classes, and a unique solution. The SVM introduction generated an enormous interest, comparable only with that produced by the development of artificial neural networks. Many applications were investigated in a short period of time, simultaneous with developments of novel SVM algorithms, such as least squares SVM [126,127] and other kernel algorithms [21,58,119]. The theory and applications of SVM are presented in a number of books, including *Learning with Kernels* by Schölkopf and Smola [115], *Learning Kernel Classifiers* by Herbrich [53], *An Introduction to Support Vector Machines* by Cristianini and Shawe-Taylor [29], and *Advances in Kernel Methods: Support Vector Learning* by Schölkopf, Burges, and Smola [117]. SVM have numerous applications in drug design, from screening of chemical libraries to QSAR [70], with accurate predictions that frequently surpass those obtained with more established

algorithms. In Sect. "Support Vector Machines" we review the most important aspects of SVM models and we present several applications to drug design, SAR and QSAR.

With an ever-increasing number of machine learning algorithms available, it is difficult to select those methods that have better chances of solving a particular SAR or QSAR problem. To address this topic, we review in Sect. "Comparative Studies" several studies that compare diverse ML models for drug design and structure-activity relationships. A comparison of 21 machine learning algorithms is presented for data selected from the National Cancer Institute 60-cell line screening panel (NCI-60) [69]. The structure-anticancer activity models are developed for three cell lines, namely for lung large cell carcinoma NCI-H460, glioma SF-268, and melanoma SK-MEL-5. Finally, we present an assessment of the results obtained in the machine learning competition CoEPrA 2006 (Comparative Evaluation of Prediction Algorithms, http://www.coepra.org/).

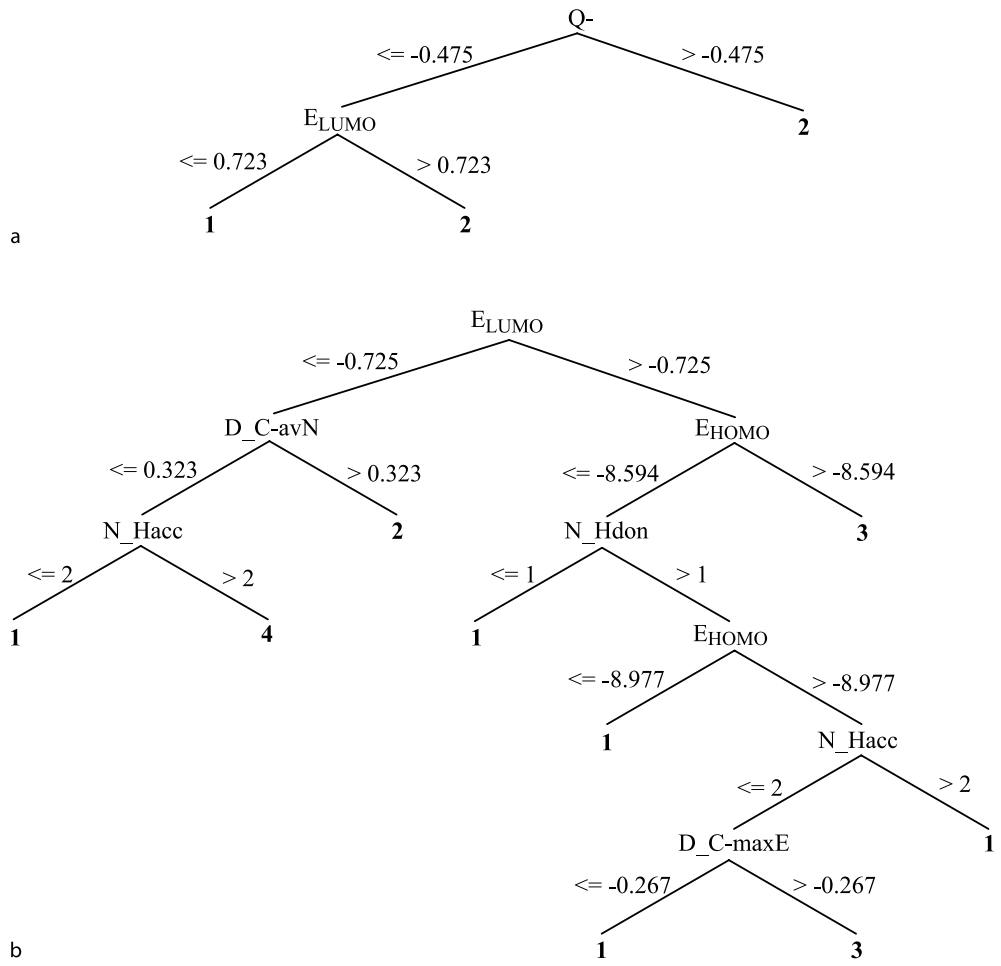
Decision Trees

A decision tree represents a series of rules that perform a mapping of the structural descriptors to a prediction for a molecular property. The class of decision tree algorithms includes the C4.5 decision tree [108], the NBTree decision tree with naïve Bayes classifiers at the leaves [82], the alternating decision tree ADTree [42], random trees and random forests [22]. A typical decision tree algorithm, such as C4.5 and related methods, generates a sequence of rules based on splitting the objects into two subgroups based on a selected structural descriptor. A threshold of the descriptor separates the objects into a subgroup of objects that have a descriptor value lower than the threshold, and another subgroup of objects that have a descriptor value higher than the threshold. The descriptor and the threshold are selected such as to maximize the difference in entropy, which maximizes the separation of objects based on their class, i. e., one subgroup contains mainly object from one class whereas the second subgroup contains mainly objects from the other class. The process is repeated until a proper separation in classes is obtained for all objects.

The structure of a C4.5 decision tree is demonstrated for the classification of narcotic pollutants into polar chemicals (class 1) and nonpolar chemicals (class 2) [66,109,137]. The dataset consists of 190 compounds, from which 76 are polar and 114 are nonpolar. The structural descriptors are the octanol-water partition coefficient $\log K_{ow}$, the energy of the highest occupied molecular orbital E_{HOMO}, the energy of the lowest unoccupied molecular orbital E_{LUMO}, the most negative partial charge on any non-hydrogen atom in the molecule $Q-$, and the most positive partial charge on a hydrogen atom $Q+$. The J48 (C4.5) decision trees computed with Weka [41,151] shows that only two descriptors, namely $Q-$ and E_{LUMO}, can separate the leaning dataset (Fig. 2a). The first rule of this decision tree separates molecules based on their $Q-$ value, namely those with $Q- \leq -0.475$ are sent to the second rule whereas those with $Q- > -0.475$ are classified in class 2. The second rule separates chemicals according to their E_{LUMO} value, namely those with $E_{LUMO} \leq 0.723$ are classified in class 1 whereas those with $E_{LUMO} > 0.723$ are classified in class 2.

The second example considers a J48 (C4.5) decision trees with four classes (Fig. 2b) representing four modes of toxic action of phenols in the *Tetrahymena pyriformis* assay, namely polar narcotics (class 1), oxidative uncouplers (class 2), proelectrophiles (class 3), and soft electrophiles (class 4). The dataset consists of 220 chemicals and seven structural descriptors, namely $\log K_{ow}$, E_{HOMO}, E_{LUMO}, the maximum donor (electrophilic) delocalizability for C atoms D_C-maxE, the average acceptor (nucleophilic) delocalizability for C atoms D_C-avN, the hydrogen bond donor count N_Hdon, and the hydrogen bond acceptor count N_Hacc. Out of the seven descriptors, only $\log K_{ow}$ is not selected to form a splitting rule, whereas E_{HOMO} and N_Hacc are selected in two rules each. Decision trees are efficient ML algorithms that provide accurate and fast predictions, with the added benefit of performing also a selection of the most useful descriptors.

Combinatorial chemistry and high-throughput screening accelerate the drug discovery process by producing vast quantities of biological assay results. The data mining of all these chemical and biological experiments requires faster structure-activity algorithms, as shown by Rusinko et al for a library of monoamine oxidase inhibitors that was evaluated with recursive partitioning [112]. The advantage of RP is that it scales linearly with the number of descriptors, and it can be used for datasets containing more than 10^5 chemicals and 2×10^6 structural descriptors. To generate chemical libraries focused on a specific target, Deng et al developed the structural interaction fingerprints, a class of descriptors that encode ligand-target binding interactions [31]. These fingerprints were used in a decision tree model to identify ligands for MAP kinase p38, showing that the method can be used with success for the structure-based screening of combinatorial chemical libraries. Statistical studies of FDA approved drugs show that there are certain physico-chemical properties that define drug-like structural requirements, independent of the drug targets. Drug-like filters are used



Drug Design with Machine Learning, Figure 2

Classification with J48 (C4.5) decision trees: a polar (class 1) and nonpolar (class 2) narcotic pollutants; b toxicity modes of phenols against the *Tetrahymena pyriformis*, namely polar narcotics (class 1), oxidative uncouplers (class 2), proelectrophiles (class 3), and soft electrophiles (class 4)

to eliminate at early stages those compounds that do not have the general structural features of a drug. Schneider et al applied decision trees to the classification of drug-like compounds based on drug-like indices, hydrophobicity, and molar refractivity [114]. This simple filter correctly identifies 83% of drugs and 39% of the non-drugs. The classification trees suggest several properties that separate drugs from non-drugs, such as a molecular weight higher than 230, a molar refractivity higher than 40, as well as the presence of structural elements such as rings and functional groups. Natural products are frequently screened to identify active compounds for selected targets. A database of 240 Chinese herbs containing 8264 compounds was screened with a random forest algorithm to identify inhibitors for several targets, including cyclooxygenases, lipoxygenases, aldose reductase, and three HIV targets [36]. The screening results show that random forests give dependable predictions even for unbalanced libraries in which inactive compounds are more numerous than active compounds. A literature search confirmed the computational predictions for 83 inhibitors identified in Chinese herbs.

Torsade de pointes (TdP) is a polymorphic ventricular arrhythmia that may be caused by drugs that induce the prolongation of the QT interval by inhibiting the heart potassium channel hERG (human ether-á-go-go). Gepp and Hutter investigated the TdP potential of 339 drugs, and found that a decision tree has a success rate of up to 80% in predicting the correct class [46]. Ekins et al combined a recursive partitioning (RP) tree with Kohonen and Sammon maps in order to identify hERG inhibitors [38]. RP is used as a fast and accurate filter to screen and prior-

itize drug databases for an in-depth assessment with Kohonen and Sammon maps. The human intestinal absorption of potential drugs may be investigated with in silico methods that assess the intestinal permeability of a chemical compound before synthesis. Hou et al used recursive partitioning to classify the intestinal absorption (poor or good) of chemical compounds [57]. The SAR model based on decision tree classification has good results for a training set of 481 compounds (95.9% success rate for the poor absorption class and 96.1% for the good absorption class) and a prediction set of 98 chemicals (100% success rate for the poor absorption class and 96.8% for the good absorption class).

All drugs that have a CNS (central nervous system) action must penetrate the blood-brain barrier (BBB), and BBB permeability models are used to filter compounds that cannot pass the barrier. Andres and Hutter found that a decision tree provides fast and accurate results, with an accuracy of 96% for 186 training compounds and 84% for 38 test chemicals [3]. The drugs that pass the BBB are predicted with a higher accuracy (94%) than those that do not pass the BBB (89%). Deconinck et al predicted BBB permeability with classification and regression trees (CART) alone and aggregated in a boosting approach [30]. The training was performed for 147 drugs, and the structural descriptors were computed with Dragon [133]. The percentage of correctly classified drugs in cross-validation is 83.6% for a single tree, and increases to 94% for a boosting model that contains 150 trees. The single tree model may be used for a fast screening of large libraries, but for more reliable predictions a boosting model should be preferred.

The drug affinity for the cytochrome P450 2C9 was predicted with a consensus method based on four ML algorithms [59]. Two of these algorithms are variants of recursive partitioning, the third is a local interpolation method, and the fourth is a subgraph search method. These four models were trained with a set of 276 compounds, and then the SAR models are tested for a prediction set of 50 chemicals. The consensus predictions have an accuracy of 94%, demonstrating that the computational methods may predict which drugs are metabolized by P450 2C9. The metabolic stability of 161 drugs against six isoforms of the cytochrome P450 (1A2, 2C9, 2C19, 2D6, 2E1, and 3A4) was evaluated with a classification tree [154]. The SAR models several structural characteristics that determine the P450 specificity, namely 3A4 substrates are larger compounds, 2E1 are smaller molecules, 2C9 substrates are anionic, and 2D6 substrates are cationic.

P-glycoprotein (Pgp) mediated drug efflux is responsible for the low cellular accumulation of anticancer drugs, for reduced oral absorption, and for low blood-brain barrier penetration. Pgp affects also the hepatic, renal, or intestinal elimination of drugs. Li el al. proposed a decision tree model to differentiate Pgp substrates from non-substrates [91]. Four-point 3D pharmacophores obtained from the three-dimensional structure of 163 chemical compounds were subsequently used as descriptors for the classification tree. Nine pharmacophores were selected in the decision tree, giving an accuracy of 87.7% for the training set and 87.6% for the prediction set. These pharmacophores highlight the structural features of Pgp substrates, and can be used as a fast filter for chemical libraries. The renal clearance of 130 drugs was investigated with Molconn-Z topological indices [76] and recursive partitioning [32]. RP results for the separation of high-clearance compounds from low-clearance compounds show 88% correct predictions for the training set of 130 compounds and 75% correct predictions for a prediction set of 20 compounds.

The prediction power of classification and regression algorithms can be significantly improved by using an ensemble of models that are combined into a final prediction by an averaging or a voting procedure. Such ML algorithms are called ensemble, committee, or jury methods. Tong et al proposed an ensemble method, decision forest (DF), which is based on classification and regression trees [134]. A DF is obtained in four steps: (a) generate a CART; (b) generate a second CART based only on structural descriptors that were not used in the first CART; (c) repeat the first two steps until no more trees can be generated; (d) predict a property of a chemical compound based on all trees generated. DF was used to predict the estrogen receptor binding activity of organic compounds from Molconn-Z topological indices [135]. After 2000 simulations of a 10-fold cross-validation, the predicted accuracy for a dataset of 232 compounds was 81.9%, and the accuracy for another dataset of 1092 chemicals was 79.7%. The DF models can be used to identify potential endocrine disruptors, which are chemicals that affect the endocrine functions. A similar algorithm is the random forest (RF) [22], which was extensively tested for several structure-activity studies [128]. RF has three properties that are important in SAR, namely the algorithm evaluates the importance of descriptors, identifies similar compounds, and measures the prediction performance. The skin sensitization activity of 131 chemicals was modeled with RF, using Dragon and Molconn-Z indices [90]. Compared with single tree models, RF improve significantly the ability to identify chemical allergens.

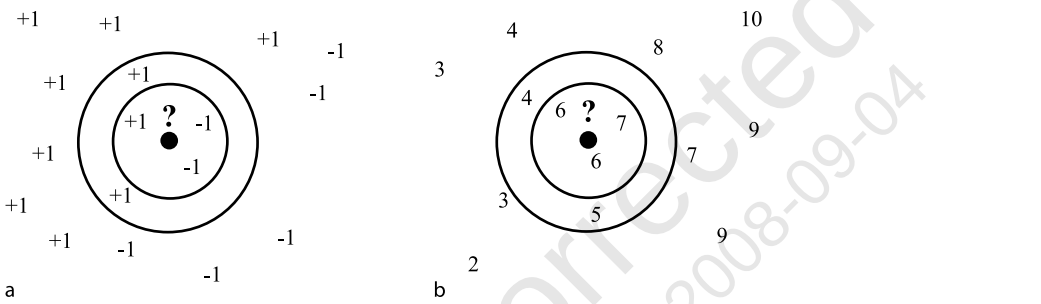
Lazy Learning and *k*-Nearest Neighbors

Lazy learning is a local learning method that keeps in the memory the entire set of training objects, and computed an interpolation model based on the neighborhood on the prediction object Atkeson, 1997 #1955 [6]. The prevalent lazy learning algorithm in SAR and QSAR is the *k*-nearest neighbors [1]. A *k*-NN prediction is based on a descriptor space, which may be the entire set of structural descriptors evaluated in a study or a subset that gives better predictions, a distance function, and a neighborhood *k* of the prediction object. The *k*-NN classification may be applied to two or more classes (Fig. 3a). The prediction object, shown here as a black circle with an unknown class "?", is inserted into the descriptor space and then its *k* nearest neighbors are identified. The class of the prediction object is the class of the majority of its neighbors. In the example from Fig. 3a, the predicted class is -1 if $k = 3$, and $+1$ if $k = 5$. The prediction statistics of the *k*-NN classifier depend on the value of *k*, and its optimum value is usually determined by cross-validation.

A similar procedure is used for *k*-NN regression (Fig. 3b). For regression, each training object has a real number as property value, and the predicted property for the query object is an average value of the property values for its *k* nearest neighbors. For the situation depicted in Fig. 3b, the predicted value for $k = 3$ is $(6+6+7)/3 = 6.3$, whereas the prediction for $k = 5$ is 5.6. A distance weight is usually added to increase the contribution of closer objects and to decrease the influence of more distant objects. For a distance function *d*, a possible weighting of the property values is the inverse distance $1/d$.

Lazy learning applications in drug design are still rare, mainly because the community of practitioners only recently discovered the rich field of machine learning algorithms. Kumar et al used locally linear embedding for nonlinear dimensionality reduction coupled with lazy learning for two QSAR benchmarks, the Selwood dataset and the steroids dataset originally analyzed with CoMFA [85]. Guha et al applied local lazy regression to three QSAR models, namely to 179 artemisinin analogues with antimalarial activity, to 79 platelet-derived growth factor inhibitors, and to 756 inhibitors of dihydrofolate reductase [47]. In all cases, the local models give better predictions compared to global regression models. An automated lazy learning quantitative structure-activity relationship (ALL-QSAR) approach was developed based on locally weighted linear regression models computed from the compounds in training set that are chemically most similar to a test compound [159]. ALL-QSAR was applied with good results for three datasets, namely 48 anticonvulsant compounds, 48 dopamine receptor antagonists, and to model the toxicity of 250 phenols against *Tetrahymena pyriformis*. Sommer and Kramer reported several applications of lazy learning to chemical datasets with a high structural diversity [123]. Collections of noncongeneric compounds are usually difficult to model, but their approach obtained good results for several datasets, namely 739 mutagenic chemicals from the Carcinogenic Potency Database, 342 chemicals tested for biodegradability, and 1481 from the AIDS Antiviral Screen of the NCI Developmental Therapeutics Program.

The fundamental assumptions in *k*-NN applications to SAR and QSAR are that molecules with similar structures are close in the descriptor space, and that similar molecules have similar properties. Basak advocated the use of *k*-NN in QSAR as an alternative to the multiple linear regression [10,11]. In this approach, the chemical compounds are represented with a collection of topological indices, and the Euclidean distance is used to identify the structures most similar with the test molecule. This *k*-NN method was applied with success to model the hydrophobicity of 4067 compounds [10], the mutagenicity of a diverse set of 520 chemicals [11], the boiling temperature of 139 hydrocarbons and the mutagenicity of 95



Drug Design with Machine Learning, Figure 3
Applications of the *k*-NN algorithm to a classification and b regression

aromatic and heteroaromatic amines [9]. Good models are obtained with optimized k values over a range between 1 and 25, as shown in QSAR model for vapor pressure [48]. The descriptors selected in the structural space are essential in establishing a suitable similarity measure. General collections of descriptors usually give good predictions, but property tailored structural spaces can increase significantly the predictivity of the model. Tailored structural spaces include only those descriptors that contribute significantly to the model, as demonstrated for hydrophobicity and boiling temperature [49]. A structural space consisting of electrotopological state indices gives reliable predictions for hydrophobicity, Henry's law constant, vapor pressure, and OH-radical bimolecular rate constant [24].

Tropsha applied the k-NN to numerous QSAR models, demonstrating its value in comparisons with more established algorithms. For a dataset of 29 dopamine antagonists, k-NN predictions are much better than those obtained with CoMFA, which represents the standard 3D QSAR model [55]. k-NN may be used also as a robust method to select those structural descriptors that are important in modeling a particular biological property, as shown for 58 estrogen receptor ligands [161] and for 48 amino acid derivatives with anticonvulsant activity [120]. In a more elaborate implementation, k-NN is used to select descriptors, to find the best value for k, and to optimize the function that weights the contribution of its neighbors [60]. The anticancer activity of 157 epipodophyllotoxin derivatives was modeled with molecular connectivity indices and k-NN [152]. In a comparison with a CoMFA model obtained for the same dataset it was found that k-NN gives better predictions in a cross-validation test. The metabolic stability of 631 drug candidates was evaluated with k-NN and a collection of structural descriptors that included molecular connectivity indices and atom pairs [121]. The prediction in a test set of compounds had accuracy higher than 85%, whereas an external validation set of 107 chemicals was predicted with 83% success rate.

k-NN is also used as a component in more elaborate drug design algorithms. Ajmani et al combined molecular field analysis and k-NN to obtain a robust QSAR model that was tested for three datasets, namely the CoMFA steroids, cyclooxygenase-2 inhibitors, and anticancer compounds [2]. Another QSAR model that incorporates k-NN is Complimentary Ligands Based on Receptor Information (CoLiBRI) [103]. CoLiBRI represents both ligands and receptor binding sites in the same space of universal chemical descriptors. In a test consisting of 800 X-ray protein-ligand receptors from PDB, CoLiBRI ranked the correct ligands in the top 1% chemicals selected, and was able to quickly eliminate improbable ligands. The skin permeability coefficients of 110 compounds were predicted with an ensemble consisting of a collection of k-NN and ridge regression models [99].

The melting temperature of organic compounds is particularly difficult to model, due mainly to a lack of good descriptors for the interactions in the solid and liquid states. Nigsch et al used several types of molecular descriptors to model the melting temperature of 4119 diverse organic molecules with k-NN [101]. The contribution of selected neighbors was evaluated with four weighting functions (arithmetic and geometric average, inverse distance weighting, and exponential weighting), with better results obtained by using the exponential weighting scheme. The optimized model has an average error of 42.2 °C. In a comparison between global and local (k-NN) regression models for the blood-brain distribution of 328 chemicals, it was found that k-NN consistently gives better predictions as measured in cross-validation [84].

Bayesian Methods

Although Bayesian classifiers were only recently applied to drug design and structure-activity studies, they have several characteristics that make them particularly useful in screening large chemical libraries. Bayesian classifiers give dependable predictions, can tolerate noise and errors in experimental data, and can be computed fast compared to other ML algorithms, which is particularly important in screening large collections of molecules. Bayes' theorem describes the relationship between the prior and conditional probabilities of two events A and B, when B has a non-zero probability [12]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} ,$$

where the terms have the following meaning: $P(A)$ is the prior or marginal probability of A, and does not consider any information regarding B; $P(B)$ is the prior (marginal) probability of B; $P(A|B)$ is the conditional (posterior) probability of A given the occurrence of event B; $P(B|A)$ is the conditional (posterior) probability of B given the occurrence of event A. Bayesian classifiers [71,151] evaluate a set of descriptors to predict the activity of a chemical based on the known descriptors and activities of a training set of chemicals. Each descriptor is considered to be statistically independent of all other descriptors. Based on a selected descriptor, the probability that a compound is active is proportional to the ratio of active to inactive compounds that have the same value for that descriptor. The final prediction is obtained by multiplying the descriptor-based probabilities.

A variety of drug discovery and development problems can be solved efficiently with Bayesian classifiers. The virtual screening of large chemical libraries represents a major application of Bayesian classifiers, mainly due to their tolerance to noise in experimental data [146]. Klon and Diller proposed a method to prioritize chemicals for synthesis and screening based on physico-chemical properties [77]. It was found that the Bayesian classifier is superior to the usual pairwise Tanimoto similarity filter. High-throughput docking is an essential tool for structure-based drug design, but the current scoring functions do not provide an accurate ranking of the molecules. Various improvements to the scoring functions were proposed, including Bayesian classifiers [78,79]. Simulations with two biological targets showed that a Bayesian classifier increases the enrichment in active compounds. Finding the potential targets of chemicals is increasingly relevant in drug discovery. Nidhi et al developed a multi-class Bayesian model in which chemicals represented with extended-connectivity fingerprints are evaluated against 964 target classes [100]. The practical utility of the approach is demonstrated by its high success rate, with 77% correct target identification. Adverse drug reactions are investigated during early phases of drug discovery in preclinical safety pharmacology tests for various targets. Multi-class Bayesian classifiers developed for 70 such targets can identify 93% of ligands with a 94% correct classification rate [14]. In a test using chemicals from World Drug Index, the classifier identifies 90% of the adverse drug reactions with a 92% correct classification rate. Adverse reactions of several drugs withdrawn from the market were also predicted with good accuracy.

Watson used 2D pharmacophore feature triplet vectors and Bayesian classifiers to find active compounds in library screening [150]. Then good enrichment in active compounds obtained for several chemogenomic databases demonstrates the practical utility of this approach. Natural products represent an excellent source of starting structures for the design of novel drugs. Ertl trained a Bayesian classifier to provide the natural product-likeness score for a chemical [39]. This score measures the similarity between a molecule and the structural space covered by natural products, and can separate between natural products and synthetic molecules. Usual similarity search in chemical libraries is based on a set of structural descriptors and a similarity (or distance) measure. Bender et al showed that the retrieval rate of active compounds increases considerably by using Bayes affinity fingerprints that describe the ligand bioactivity space [13]. In this space, Bayes scores for ligands from about 1000 activity classes describe the chemical structures. This approach was able to improve with 24% the results from a standard similarity screening.

Bayesian methods are also used in various structure-activity models. Klon et al demonstrated several applications of modified Bayesian classifiers that model continuous numerical data with a Gaussian distribution [80]. Several models developed for absorption, distribution, metabolism and excretion property prediction show that the new classifiers have better performance compared to classical Bayesian classifiers. Multidrug resistance represents the ability of cancer cells to become simultaneously resistant to several drugs. A possible solution to this problem is the use of multidrug resistance reversal (MDDR) agents. Sun found that a Bayesian classifier based on atom types could be used to identify MDDR agents in a set of 424 training compounds [125]. The prediction of MDDR agents in a group of 185 test compounds had a success rate of 82.2%. Phospholipidosis is an adverse effect that is investigated during preclinical studies in animals, and may stop the development of an otherwise promising drug candidate. Pelletier et al proposed an in silico Bayesian classifier to identify phospholipidosis as a viable alternative to more expensive and time-consuming in vivo tests [105]. Based on two simple descriptors, pK_a and ClogP, the model has a success rate of 83% for a dataset of 201 compounds. UGT is an enzyme family that catalyzes the reaction between glucuronic acid and a chemical that has a nucleophilic group. Drug glucuronidation represents an important mechanism for their elimination, but the selectivity of different UGT isoforms is difficult to predict. The Bayesian classifier suggested by Sorich et al predicts the glucuronidation site for eight UGT isoforms based on the partial charge and Fukui function of the nucleophilic atom and the presence of an aromatic ring attached to the nucleophilic atom [124]. The models have a good predictive ability, with a sensitivity and specificity in the range 75–80%. The screening test to identify bitter compounds presented by Rodgers et al uses selected substructural features selected from 649 bitter chemicals and 13,530 random drug-like compounds [111]. A Bayesian classifier trained with these data can predict correctly 72.1% of the bitter compounds. The bitter taste is determined mainly by substructures representing sugar moieties and highly branched carbon skeletons.

Support Vector Machines

Support vector machines represent a major development for SAR and QSAR models, as suggested by the large number of publications that apply SVM and related kernel methods to drug design. To better understand the math-

ematical basis of SVM and the parameters that influence their results, we start this section with a brief theoretical presentation of SVM for classification and regression. Several kernel functions are presented together with SVM plots that demonstrate the complex shapes that may be simulated with these functions. The influence of various kernels on QSAR predictions is shown for a dataset of benzodiazepine receptor ligands, and several applications in drug design are reviewed.

Hard Margin Linear SVM Classification

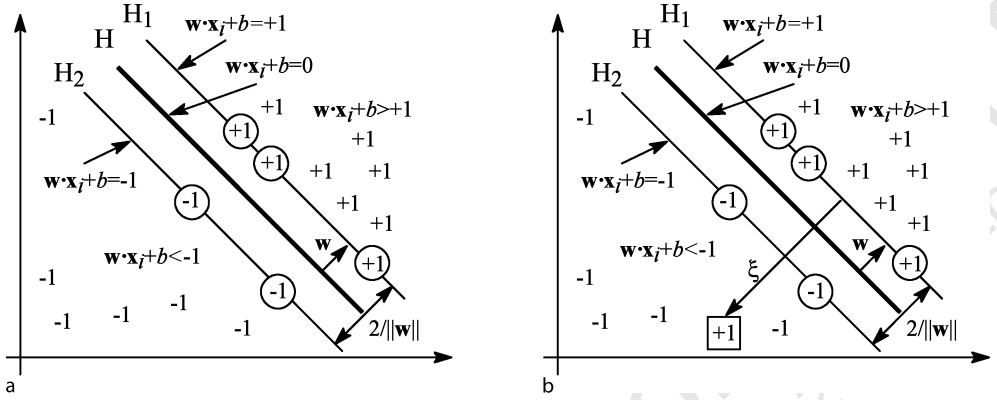
The most simple case of SVM is the classification of two classes of objects that may be separated by a hyperplane (Fig. 4a). It is obvious from this figure that there are many hyperplanes that can separate the two classes of objects, but not all of them give good predictions for novel objects. Intuitively, a hyperplane that maximizes the separation of the two classes, like hyperplane H, is expected to offer the best predictions. The SVM algorithm determines a unique hyperplane H that has the maximum margin, i. e., the maximum distance between hyperplanes H_1 and H_2. Hyperplane H_1, which is determined by three objects from class +1 that are shown inside circles, defines the border with class +1. Hyperplane H_2, which is determined by two objects from class −1 that are shown also inside circles, defines the border with class −1. The objects represented inside circles are called support vectors and they determine the unique solution for the SVM model. By removing all other objects and keeping only the support vectors one obtains the same SVM model. The hyperplane H is defined by $\mathbf{w}\cdot\mathbf{x}+b=0$, where $\mathbf{w}$ is the normal vector to H, the hyperplane H_1 that borders class +1 is defined by $\mathbf{w}\cdot\mathbf{x}+b=+1$, and the hyperplane H_2 that borders class −1 is defined by $\mathbf{w}\cdot\mathbf{x}+b=-1$. The distance between the origin and the hyperplane H is $|b|/\|\mathbf{w}\|$, and the SVM margin (the distance between hyperplanes H_1 and H_2) is $2/\|\mathbf{w}\|$. A molecule i characterized by a vector of structural descriptors $\mathbf{x}_i$ belongs to class +1 if $\mathbf{w}\cdot\mathbf{x}_i+b\geq+1$, or to class −1 if $\mathbf{w}\cdot\mathbf{x}_i+b\leq-1$. Obviously, support vectors from class +1 are determined by $\mathbf{w}\cdot\mathbf{x}_i+b=+1$, whereas support vectors from class −1 are determined by $\mathbf{w}\cdot\mathbf{x}_i+b=-1$.

Based on the formula of the SVM margin, it follows that the maximum separation hyperplane is obtained by maximizing $2/\|\mathbf{w}\|$, which is equivalent to minimizing $\|\mathbf{w}\|^2/2$. The linear SVM model (Fig. 4a) is formulated as:

$$\begin{aligned}&\text{minimize}\quad f(\mathbf{x})=\frac{\|\mathbf{w}\|^2}{2}\\&\text{with the constraints}\quad g_i(\mathbf{x})=y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\geq 0\,,\\&\qquad\qquad i=1,\ldots,m\end{aligned}\tag{2}$$

where y_i is the class (+1 or −1) of the object i. This optimization equation is a quadratic programming that is further transformed with a Lagrangian function into its dual formulation. All SVM formulations are solved for the dual formulation, because the solution obtained for this simple case is easily extended to more complex situations. The minimization problem from Eq. (2) is expressed with a Lagrangian function:

$$\begin{aligned}L_P(\mathbf{w},b,\mathbf{A})&=f(\mathbf{x})+\sum_{i=0}^{m}\alpha_i g_i(\mathbf{x})\\&=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1)\end{aligned}$$



Drug Design with Machine Learning, Figure 4
SVM classification models: a linearly separable data; b linearly non-separable data

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i (\mathbf{w}\cdot\mathbf{x}_i + b) + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i \mathbf{w}\cdot\mathbf{x}_i$$

$$- \sum_{i=1}^{m} \alpha_i y_i b + \sum_{i=1}^{m} \alpha_i \,, \qquad (3)$$

where $\mathbf{A} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ is the set of Lagrange multipliers for the objects that have $\alpha_i \geq 0$, and P in L_P indicates the primal formulation of the problem. The Lagrangian function L_P must be minimized with respect to $\mathbf{w}$ and b, and maximized with respect to α_i, subject to the constraints $\alpha_i \geq 0$. The dual optimization problem L_D is:

$$\begin{aligned} &\text{maximize} \\ &L_D(\mathbf{w}, b, \mathbf{A}) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ &\text{subject to } \alpha_i \geq 0\,, \quad i = 1, \ldots, m \\ &\text{and } \sum_{i=1}^{m} \alpha_i y_i = 0\,. \end{aligned} \qquad (4)$$

The optimization problem is usually solved with the sequential minimal optimization (SMO) proposed by Platt [106]. For a complete formulation of the SVM solution see the books and reviews mentioned in Introduction [70]. The training objects with $\alpha > 0$ constitute the support vectors, whereas the objects with $\alpha = 0$ may be removed from the learning set without affecting the SVM model. The vector $\mathbf{w}$ is obtained as:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \qquad (5)$$

and b is computed as the average value for all support vectors. The SVM model is the optimum separation hyperplane ($\mathbf{w}$, b) that can now be used to predict the class membership for new objects. The class of a molecule k with the structural descriptors $\mathbf{x}_k$ is:

$$\text{class}(k) = \begin{cases} +1 & \text{if } \mathbf{w}\cdot\mathbf{x}_k + b > 0 \\ -1 & \text{if } \mathbf{w}\cdot\mathbf{x}_k + b < 0\,. \end{cases} \qquad (6)$$

The classifier is further simplified by substituting $\mathbf{w}$ with its expression (5):

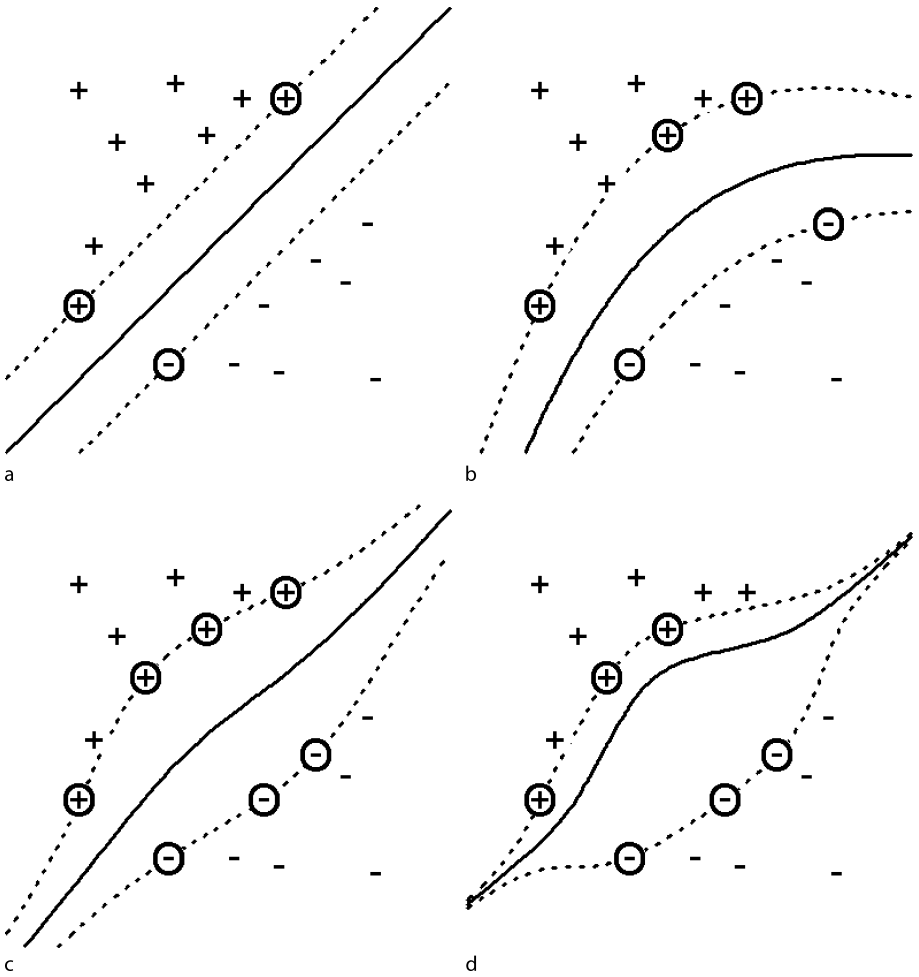
$$\text{class}(k) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_k + b\right)\,, \qquad (7)$$

where the summation goes only over the support vectors. The SVM described in the above equations is called a "hard margin" SVM because it does not allow for classification errors.

The class separation with a linear SVM is simple and intuitive. However, SVM are usually used with nonlinear kernels that have a much more complex shape, which might result in too complicates separation surfaces. To demonstrate the shape of the separation surface for linearly separable data we compare results obtained with a linear kernel and several nonlinear kernels that will be introduced later. All calculations were performed with *R* (http://www.r-project.org/) and the kernlab package. In all figures, class + 1 patterns are represented by "+" and class −1 patterns are represented by "−". The SVM hyperplane is depicted with a continuous line, whereas the margins of the SVM hyperplane are shown with dotted lines. Support vectors from class + 1 are represented as "+" inside a circle, and support vectors from the class −1 are depicted as "−" inside a circle.

The linear kernel gives an SVM model that has an optimum separation of the two classes (Fig. 5a) with three support vectors, namely two from class + 1 and one from class −1. The hyperplane has the maximum width, and no object is situated inside the margins (represented with dotted lines). The classification of new objects is made by applying Eq. (7) to the three support vectors. The same dataset is modeled with a degree 2 polynomial kernel (Fig. 5b) in which case the SVM model has five support vectors, namely three for class + 1 and two for class −1. The margin width varies, being larger in the middle and smaller towards the extremes. The hyperplane topology is different from that obtained with a linear kernel, and obviously for some objects the two SVM models will give different predictions. The SVM classifier obtained with a degree 3 polynomial kernel (Fig. 5c) has four support vectors from class + 1 and three support vectors from class −1, and the margin becomes smaller towards the ends. Higher order polynomial kernels produce a complex separation hyperplane, as shown also for a degree 10 polynomial kernel (Fig. 5d), in which case the margin almost vanished towards the extremes.

The Gaussian radial basis function (RBF) kernel is the most common option for SVM, and in Fig. 6 we present four SVM models obtained for different σ values. For low σ values the RBF kernel (Fig. 6a, $\sigma = 0.01$) approximates a linear kernel (Fig. 5a) and the two hyperplanes are very similar. As σ increases the nonlinear behavior of the kernel becomes apparent (Fig. 6b, $\sigma = 0.1$), but the SVM model still has a good separation of the data. As σ increases to 1 (Fig. 6c) and then to 5 (Fig. 6d) the nonlinearity of the SVM model becomes more apparent, and the number of support vectors increases. In the last case all data points are



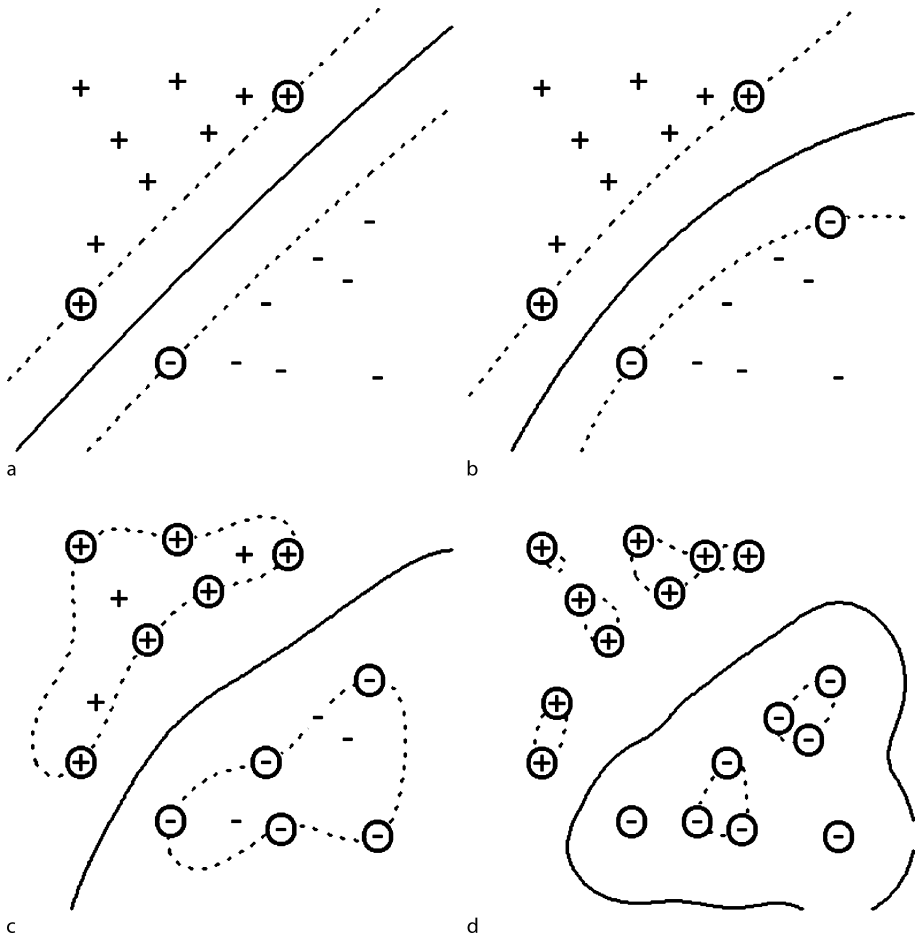
Drug Design with Machine Learning, Figure 5
SVM classification models for linearly separable data: a dot kernel (*linear*); b polynomial kernel, degree 2; c polynomial kernel, degree 3; d polynomial kernel, degree 10

support vectors, and the SVM model has a very complex topology. The examples presented here show that the RBF kernel may generate too complex hyperplanes that do not properly reflect the structure of the data.

Another important kernel is the hyperbolic tangent (tanh), which is also used in artificial neural networks. Some combinations of parameters result in SVM models almost identical with those obtained with the linear kernel (Fig. 7a) or similar (Fig. 7b). Higher values for the two parameters result in SVM models with smaller margins and nonlinear separation of the two classes (Fig. 7c,d). The SVM models demonstrated here for linearly separable data show that nonlinear kernels, such as RBF and tanh, may give SVM models similar to those obtained with linear kernels, for certain values of their parameters. However, nonlinear kernels may also give SVM models that are too complex and that do not follow the structure of the data. As a general rule, the SVM computed with nonlinear kernels should be compared with a linear kernel SVM.

Soft Margin Linear SVM Classification

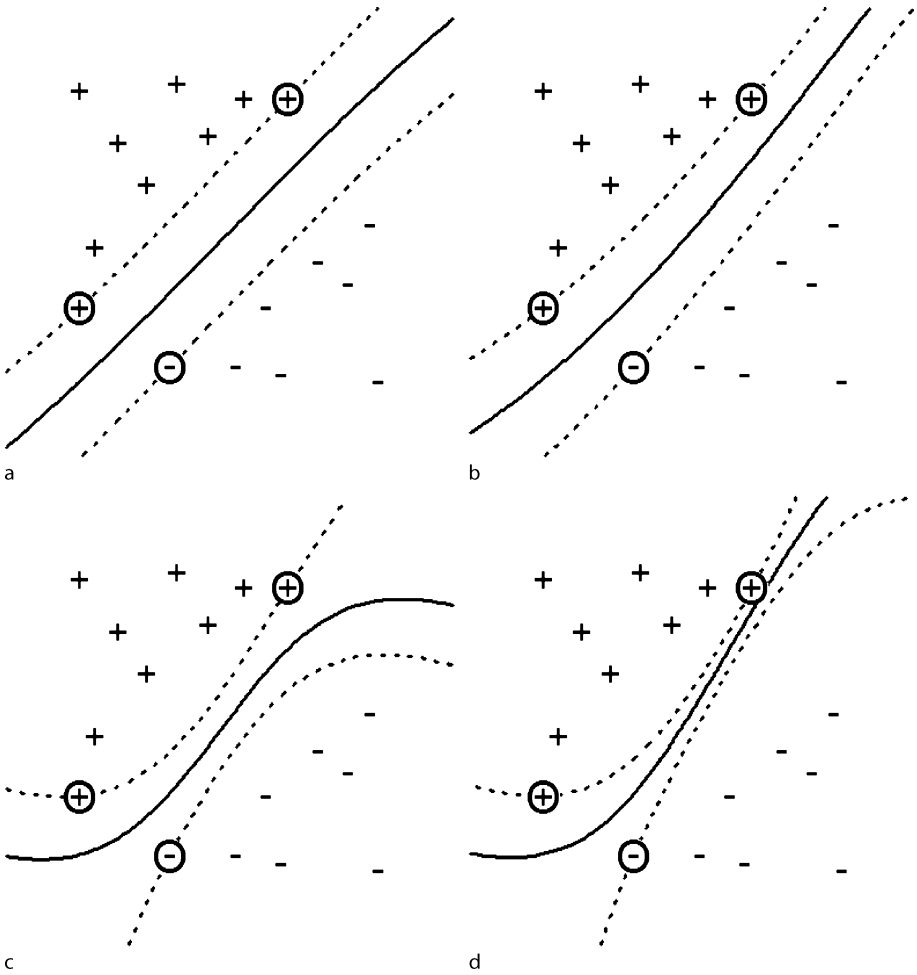
The classification problem solved in the previous section considered a two-class situation with classes separated with a linear classifier. However, such cases are rare in drug design applications, and most often the classes of compounds have regions where they superpose. There are several reasons why classes of chemicals cannot be separated with a linear classifier. The most common problem is the identification of the structural descriptors that determine the property. There are thousands of structural descriptors proposed in the literature, but it is difficult to



Drug Design with Machine Learning, Figure 6
SVM classification models for linearly separable data with the Gaussian RBF kernel: a $\sigma = 0.01$; b $\sigma = 0.1$; c $\sigma = 1$; d $\sigma = 5$

try them all in a SAR or QSAR study. It is always a possibility that the proper type of descriptors is not even discovered. The second problem is to identify the subset of descriptors that give the best predictions, which is a time-consuming task that is usually solved with heuristic algorithms that give a sub-optimal solution. The mapping between the descriptor space and the property might be nonlinear, and obviously a linear classifier fails to model the data. As a final point we have to mention that experimental data may be affected by noise, measurement errors, interference with other biological targets, or unaccounted factors that determine the outcome of the experiments. Although the hard margin linear SVM is limited in its abilities to model real data, the formalism presented in the previous section is the basis for developing more complex SVM models. The linear SVM with soft margin is such an extension that allows classification errors.

The classification problem shown in Fig. 4b is very similar with the linear separable case from Fig. 4a, because all but one object can be separated with a linear hyperplane H. The exception is a object from class +1 that is situated in the −1 space region (shown inside a square). A penalty term is introduced for misclassified data, with the property that it is zero for objects correctly, and it has a positive value for objects that are not situated on the correct side of the classifier. The value of the penalty increases when the distance of the corresponding hyperplane increases. The penalty is called a slack variable and it is denoted with ξ. For the situation depicted in Fig. 4b the penalty for the misclassified object is measured starting from hyperplane H_1 because this hyperplane border the + 1 region. For + 1 molecules situated in the buffer zone between H and H_1, and for −1 molecules situated in the buffer zone between H and H_2, the slack variable takes values between 0 and 1.



Drug Design with Machine Learning, Figure 7
SVM classification models for linearly separable data with the hyperbolic tangent (tanh) kernel: a $a = 0.1, b = 0$; b $a = 0.1, b = 0.5$; c $a = 0.5, b = 0$; d $a = 0.5, b = 0.5$

Such patterns are not considered to be misclassified, but have a penalty added to the objective function. The constraints for the objective function for soft margin linear SVM are:

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 - \xi_i & \text{if } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i & \text{if } y_i = -1 \\ \xi_i > 0\,, \quad \forall i\,. \end{cases} \tag{8}$$

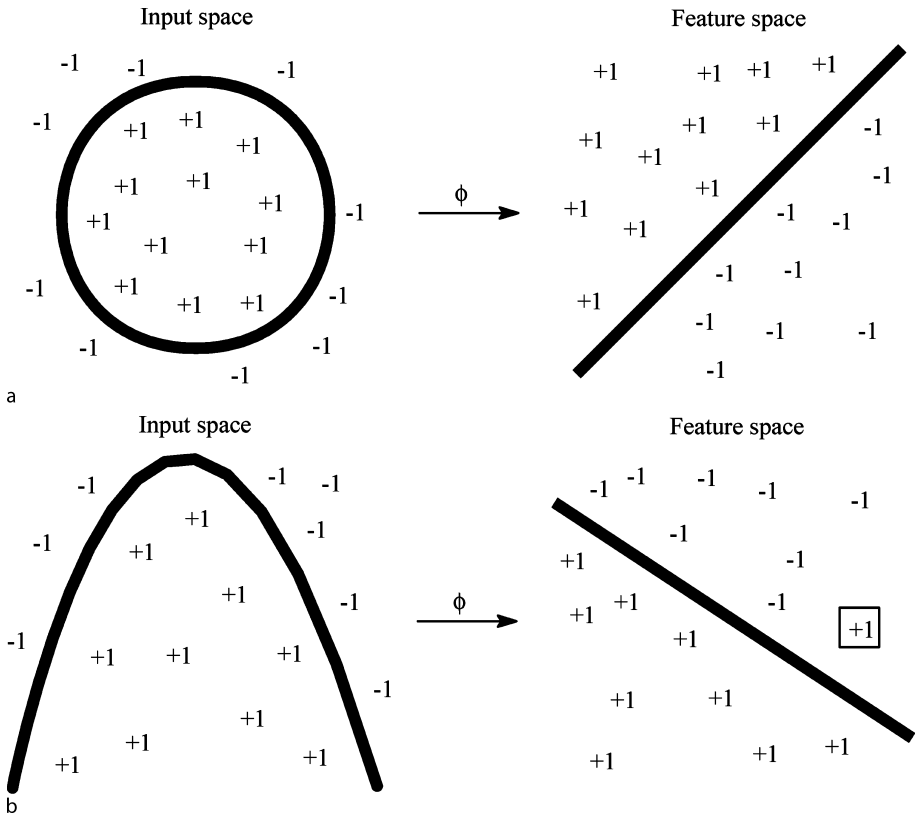
The soft margin SVM should balance two opposite conditions, namely a maximum margin and a minimum of errors that is equivalent with minimizing a sum of slack variables. The optimization problem for a linear SVM with classification errors is;

$$\text{minimize } \frac{\|\mathrm{w}\|^2}{2} + C \sum_{i=1}^{m} \xi_i$$

with the constraints (9)

$$\begin{aligned} y_i(\mathrm{w} \cdot \mathrm{x}_i + b) &\geq +1 - \xi_i\,, \quad i = 1, \ldots, m \\ \xi_i &\geq 0\,, \quad i = 1, \ldots, m \end{aligned}$$

where C is called capacity and is a parameter that weights the penalty for classification errors. A large value for the capacity parameter C represents a high penalty for classification errors, which forces the SVM solution to minimize the number of misclassified objects by reducing the margin of the classifier. A small capacity C lowers the value of the penalty term and maximizes the margin, which makes



Drug Design with Machine Learning, Figure 8
Patterns that are nonlinear separable in the input space are linear separable in the feature space: a the class separation is complete; b a pattern from class + 1 (depicted in a *square*) is situated in the −1 region of the hyperplane

the solution less sensitive to classification errors. The optimization problem is solved with Lagrange multipliers, similar with the procedure outlined for the hard margin linear SVM, when the primal Lagrangian expression is:

$$L_P(\mathbf{w}, b, \mathbf{A}, \mathbf{M}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\mu_i\xi_i \quad (10)$$

where the Lagrange multipliers $\mathbf{A} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ are assigned to each constraint $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq +1 - \xi_i$, and the Lagrange multipliers $\mathbf{M} = (\mu_1, \mu_2, \ldots, \mu_m)$ are assigned to each constraint $\xi_i \geq 0, \forall i = 1, \ldots, m$. The dual optimization problem is:

maximize

$$L_D(\mathbf{w}, b, \mathbf{A}, \mathbf{M}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j$$

subject to $0 \leq \alpha_i \leq C\,, \quad i = 1, \ldots, m$

and $$\sum_{i=1}^{m}\alpha_i y_i = 0\,. \quad (11)$$

The expression for the vector $\mathbf{w}$ is identical with that obtained for hard margin SVM:

$$\mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\,. \quad (12)$$

The SVM classifier, which is determined by the pair $(\mathbf{w}, b)$, may be used for the classification of new molecules. The class of a molecule k with the structural descriptors $\mathbf{x}_k$ is:

$$\text{class}(k) = \text{sign}\left(\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\cdot\mathbf{x}_k + b\right) \quad (13)$$

where the summation goes only over the support vectors.

Nonlinear Classification with Kernels

Nonlinear relationships between structural descriptors and drug properties may be modeled with SVM equations that replace the linear classifier with nonlinear kernel functions. The SVM extension to nonlinear relationships is based on the property of feature functions $\phi(\mathbf{x})$ that transform the input spaces into a feature space of higher dimensionality, in which a linear separation of the classes is possible. This transformation is depicted in Fig. 8a. In the input space, which is represented by the structural descriptors, a circle can discriminate the two classes of objects. Using a proper feature function ϕ the input space is transformed into a feature space in which the two classes can be separated with a linear classifier. Figure. 8b illustrates another situation in which another nonlinear function can discriminate the classes in input space, which is then transformed by the function ϕ into a linear classifier in feature space. The challenge is to find that feature function ϕ that maps the objects into a feature space in which the classes are linearly separable. In real-life applications it is not even desirable to achieve a perfect separation, because the function may map noise or errors in data (see Fig. 8b, where a +1 object, depicted in a square, is situated in the −1 region of the hyperplane). Due to the fact that in feature space the objects may be separated with a linear classifier, it is possible to use the results for linear SVM presented in the previous sections, which is a fundamental property of SVM.

The mapping from the input space to the feature space may be expressed as:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_h(\mathbf{x})) \,. \quad (14)$$

A soft margin SVM in the feature space is obtained by simply substituting the input vectors $\mathbf{x}$ with the feature vectors $\phi(\mathbf{x})$. We have to point here that a good linear classifier in feature space depends on a proper form of the function ϕ. The class of a molecule k with the structural descriptors $\mathbf{x}_k$ is:

$$\begin{aligned} \text{class}(k) &= \text{sign}[\mathbf{w} \cdot \phi(\mathbf{x}_k) + b] \\ &= \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_k) + b\right) \end{aligned} \quad (15)$$

The prediction for a molecule k requires the computation of the dot product $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_k)$ for all support vectors $\mathbf{x}_i$. This is an important observation showing that we do not have to know the actual expression of the feature function ϕ. Furthermore, nonlinear SVM use kernels, which are a special class of functions that compute the dot product $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_k)$ in the input space:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \,. \quad (16)$$

Several popular kernel functions are presented below. The inner product of two vectors defines the linear (dot) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j \,. \quad (17)$$

The dot kernel is a linear classifier, and should be used as a reference to demonstrate an eventual improvement of the classification with nonlinear kernels. The polynomial kernel is useful mainly for lower degrees, because for higher degrees it has the tendency to overfit the data:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^d \,. \quad (18)$$

The Gaussian radial basis functions (RBF) kernel is very flexible, and depending on the values of the parameter σ it can display a wide range of nonlinearity:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma \|x - y\|^2) \,. \quad (19)$$

The hyperbolic tangent (tanh) function has a sigmoid shape and it is the most used transfer function for artificial neural networks. The corresponding kernel has the formula:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(a\mathbf{x}_i \cdot \mathbf{x}_j + b) \,. \quad (20)$$

The nonlinearity of the anova kernel is controlled by the parameters γ and d:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_i \exp(-\gamma(\mathbf{x}_i - \mathbf{x}_j))\right)^d \,. \quad (21)$$

Hard Margin Nonlinear SVM Classification

The formulation of the hard margin nonlinear SVM classification is obtained from the linear SVM by substituting input vectors $\mathbf{x}$ with feature functions $\phi(\mathbf{x})$. The dual problem is:

maximize

$$L_D(\mathbf{w}, b, \mathbf{A}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

subject to $\alpha_i \geq 0 \,, \quad i = 1, \ldots, m$

and $\sum_{i=1}^{m} \alpha_i y_i = 0 \,.$ (22)

The optimum separation hyperplane is determined by the vector **w**:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i) \,. \tag{23}$$

In the final expression of the classifier, the dot product for two feature functions $\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_k)$ is replaced with a kernel function $K(\mathbf{x}_i, \mathbf{x}_k)$:

$$\text{class}(k) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + b\right) \,. \tag{24}$$

The structure of a nonlinear SVM may be represented in a network format (Fig. 9). The network input is represented by a set of support vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a prediction object $\mathbf{x}_p$. The input space is transformed with a feature function ϕ into the feature space, and the dot product of feature functions is computed and multiplied with the Lagrangian multipliers α. The output is Eq. (24) in which dot product of feature functions is substituted with a kernel function K.

Soft Margin Nonlinear SVM Classification

The expression of the soft margin nonlinear SVM classification is obtained from the corresponding linear SVM formula by substituting input vectors **x** with feature functions $\phi(\mathbf{x})$. The dual problem is:

$$\begin{aligned} &\text{maximize} \\ &L_D(\mathbf{w}, b, \mathbf{A}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j) \\ &\text{subject to } \; 0 \le \alpha_i \le C \,, \quad i = 1, \ldots, m \\ &\text{and } \; \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{25}$$

with a solution identical with that for hard margin nonlinear SVM classification from Eq. (24).

In Fig. 10 we demonstrate the nonlinear SVM classification for the Gaussian RBF kernel. The best separation of the two classes is obtained for a low σ value (Fig. 10a, $\sigma = 0.1$). As the value of σ increases the kernel nonlinearity increases and the separation surface becomes more complex. It is clear from this example that the kernel nonlinearity must match the structure of the data. Also, highly nonlinear functions do not translate in better SVM models. In practical applications the parameters that determine the kernel shape must be optimized to provide the best predictions in a cross-validation test.

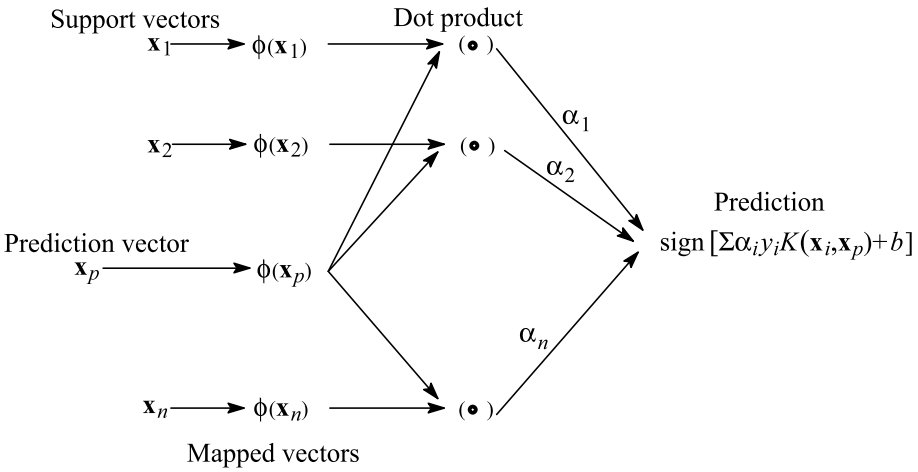
SVM Regression

Vapnik extended the SVM algorithm for regression [139] by using an ε-insensitive loss function (Fig. 11). The SVM regression (SVMR) determines a function $f(\mathbf{x})$ with the property that for all learning objects **x** it has a maximum deviation ε from the target (experimental) values y, and it has a maximum margin. Starting from the training objects, SVMR computes a model representing a tube with radius ε fitted to the data. All object situated inside the regression tube have an error equal to zero. For the hard margin SVMR no object is allowed outside the tube, whereas for the soft margin SVMR uses positive slack variables to penalize the objected situated outside the tube [44,45,93,94,122]. If Fig. 11 the objects situated inside the regression tube are shown as white circles, whereas the objects outside the regression tube are depicted as black circles. The slack variable ξ introduces a penalty proportional with the distance between the object and the margin of the regression tube. Support vectors are selected only from objects situated outside the tube.

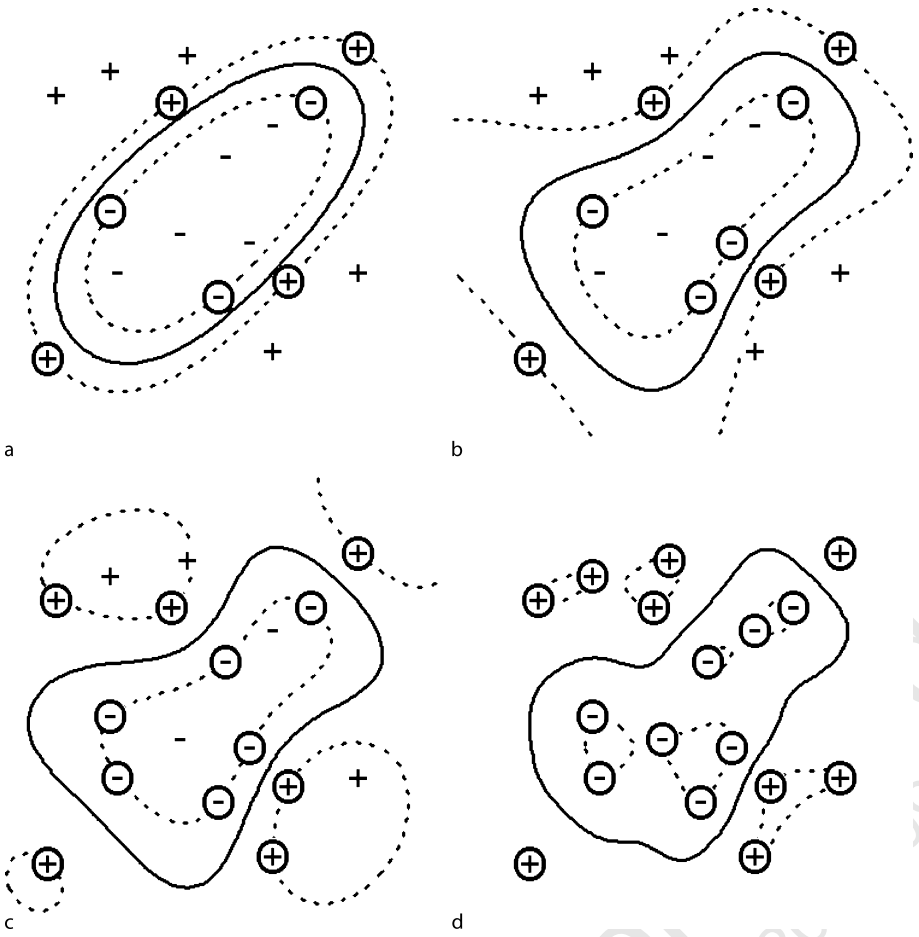
To demonstrate the kernel influence on the SVM regression we model the inhibition constant K_i for bovine milk xanthine oxidase [51]. The structural descriptor used is ClogP, the computed hydrophobicity. To model the nonlinear dependence between ClogP and K_i we apply a degree 2 polynomial kernel (Fig. 12) with different values for the ε-insensitive parameter. The molecules situated inside the regression tube are indicated with black triangles, whereas those located outside the regression tube are represented as black dots (support vectors are shown as black dots inside a circle). A low value for ε produces a SVMR model with a larger number of support vectors, but an overall good shape for the relationship between ClogP and K_i (Fig. 12a, $\varepsilon = 0.1$). As ε increases the diameter of the regression tube includes almost all molecules, and thus the model depends on a small number of support vectors because objects situated inside the tube do not influence the regression model.

We explore further two higher order polynomials, namely a degree 5 polynomial kernel (Fig. 13a) and a degree 8 polynomial kernel (Fig. 13b). Both kernels overfit the data, and have shapes that are too complex. The Gaussian RBF kernel with $\sigma = 0.2$ is a good fit for the K_i data (Fig. 13c), but an increase in σ results in a regression that start to fit the errors in the experimental data (Fig. 13d).

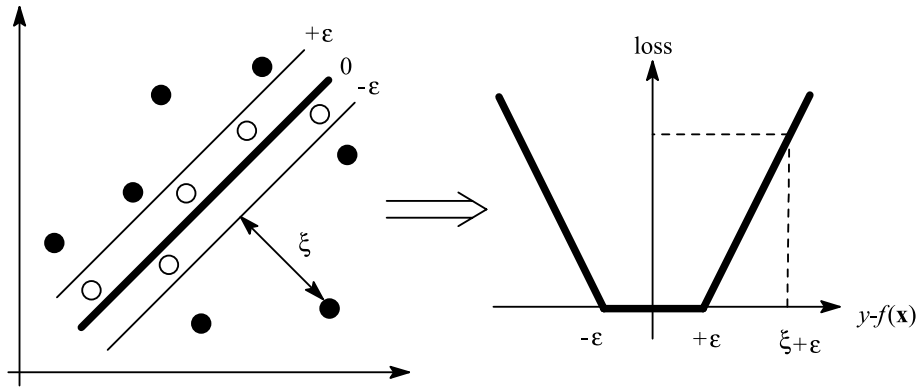
The next set of SVM regression QSAR evaluates the tanh kernel (Fig. 14). The first combination of parameters is insensitive to the relationship between ClogP and K_i (Fig. 14a). Although the K_i data are best represented by a downward parabola, the second tanh QSAR has the



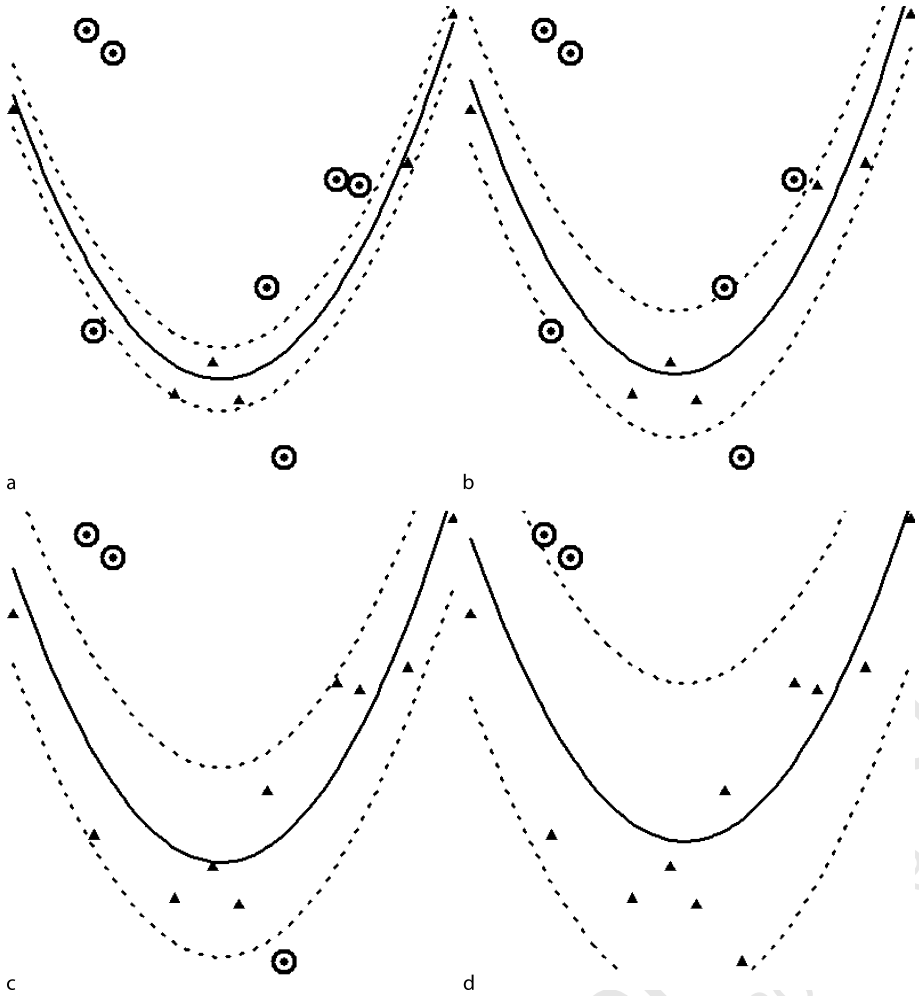
Drug Design with Machine Learning, Figure 9
Network representation of support vector machines



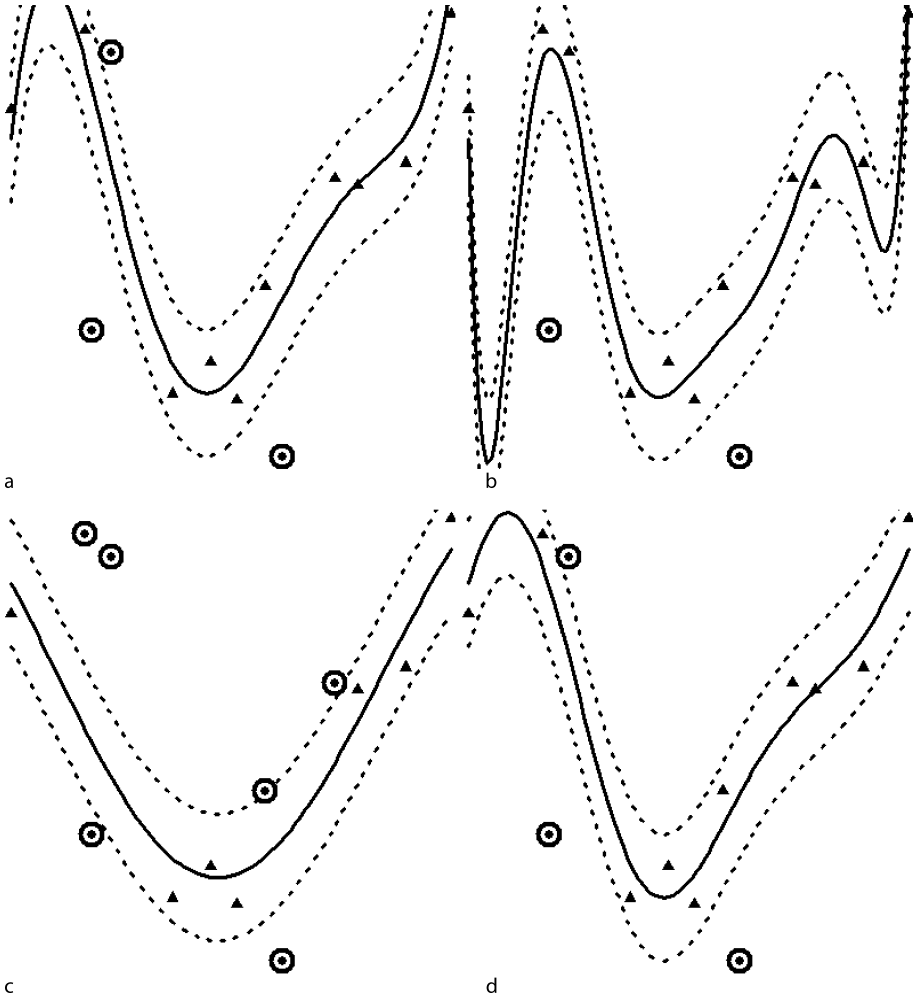
Drug Design with Machine Learning, Figure 10
SVM classification models for linearly non-separable data with the Gaussian RBF kernel: a $\sigma = 0.1$; b $\sigma = 0.5$; c $\sigma = 1$; d $\sigma = 5$



Drug Design with Machine Learning, Figure 11
SVM regression with ε-insensitive loss function



Drug Design with Machine Learning, Figure 12
SVM regression models with a degree 2 polynomial kernel ($C = 1000$): a $\varepsilon = 0.1$; b $\varepsilon = 0.2$; c $\varepsilon = 0.3$; d $\varepsilon = 0.5$



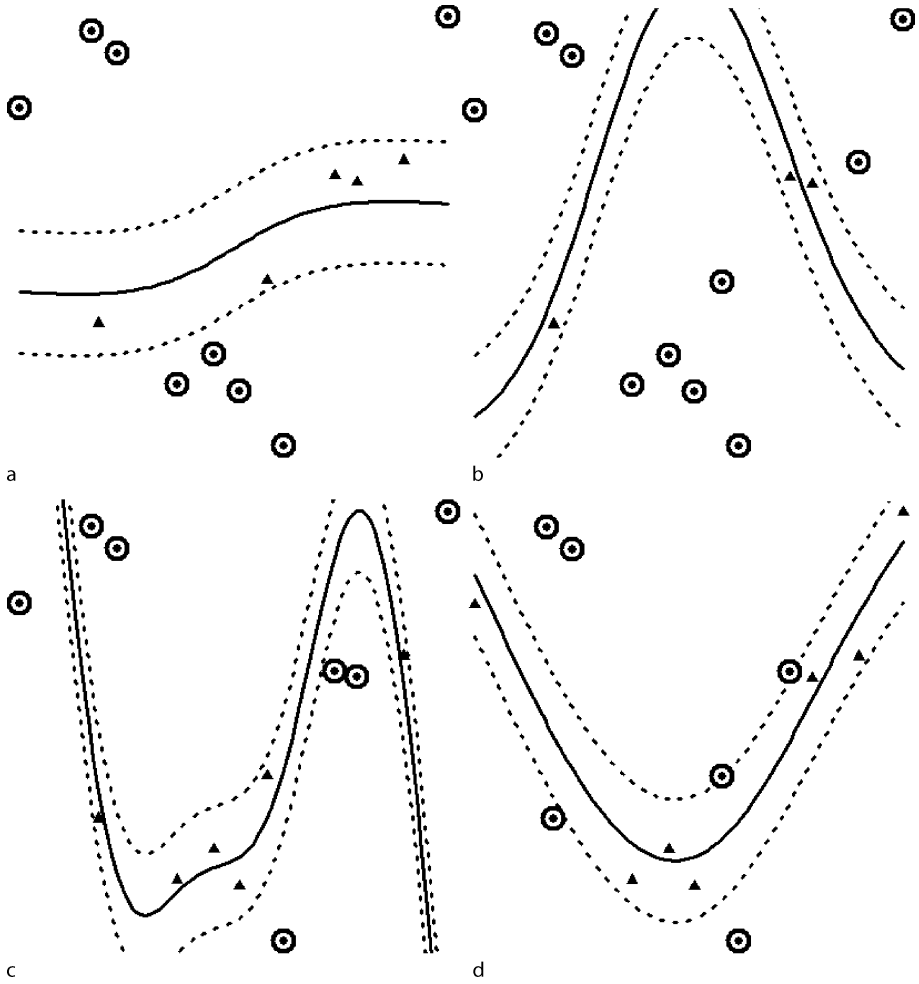
Drug Design with Machine Learning, Figure 13
SVM regression models ($C = 1000$, $\varepsilon = 0.2$): a degree 5 polynomial kernel; b degree 8 polynomial kernel; c Gaussian RBF kernel, $\sigma = 0.1$; d Gaussian RBF kernel, $\sigma = 0.5$

shape of an upward parabola, and misses the most apparent correlation between ClogP and K_i (Fig. 14b). The third tanh QSAR is overfitted, mainly in its right side (Fig. 14c). The last combination of parameters for the tanh QSAR shows a very good fit of the data and obviously it is a good candidate for the prediction of new molecules (Fig. 14d). Considering the large variation in shape of the tanh QSAR it is apparent that its parameters should be varied over a large range to find the best regression model.

Since the dataset considered is a real QSAR problem, it is interesting to examine the behavior of other, less common, kernels. The anova RBF kernel shows an overfit of the data, which is more prominent in its left side (Fig. 15a). The spline kernel has a too low nonlinearity, and the model has no value in predicting new chemicals (Fig. 15b). The degree 1 Bessel kernel is a very good fit for the correlation between ClogP and K_i (Fig. 15c), whereas a degree 5 Bessel kernel overfits the data in the left side (Fig. 15d). The results presented here are only to illustrate the shape of various kernels, and to demonstrate their ability to model nonlinear relationships. Other combinations of parameters and capacity C may give better or worse correlations, and the most predictive QSAR may be identified only by optimizing the kernel parameters.

Kernel Comparison in QSAR

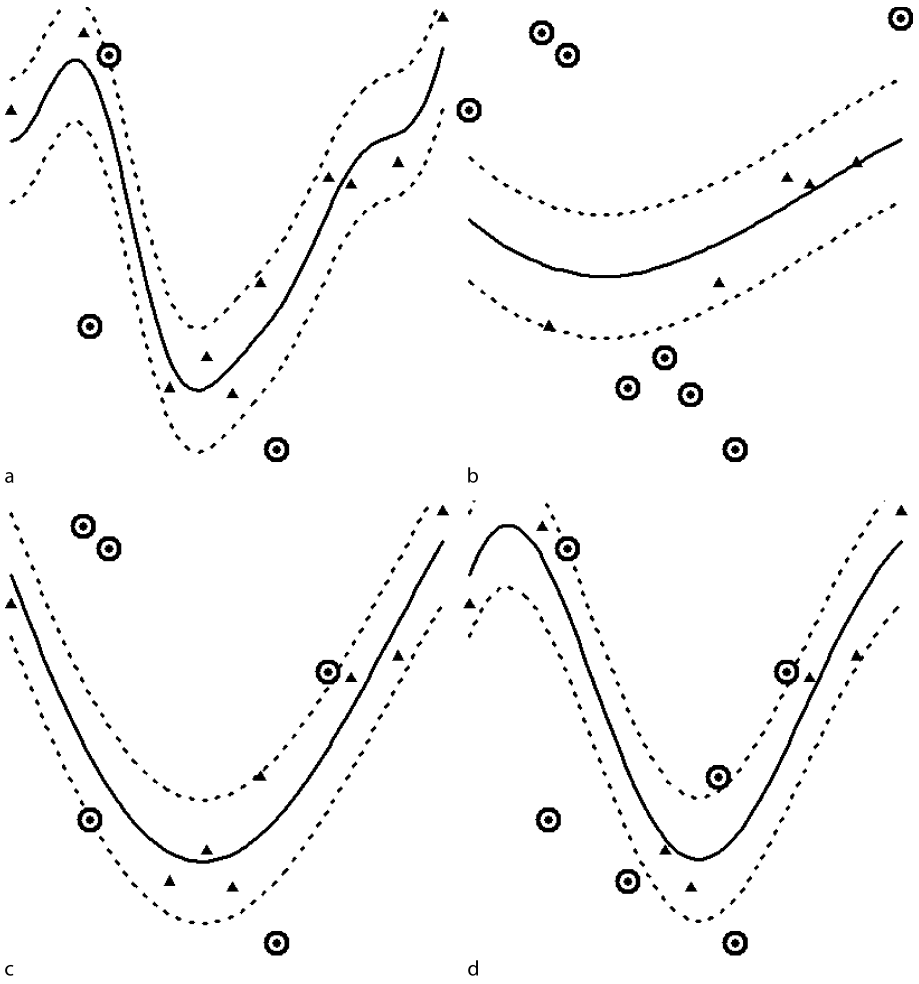
Several kernels have the property to give SVM models that are universal approximators to arbitrary functions, namely the SVM model can approximate any function to any level



Drug Design with Machine Learning, Figure 14
SVM regression models with a hyperbolic tangent (tanh) kernel ($\varepsilon = 0.2$): a $a = 1, b = 0, C = 1$; b $a = 1, b = 1, C = 1$; c $a = 0.9, b = -1, C = 10$; d $a = 0.5, b = -0.7, C = 10$

of precision. However, the theory cannot guarantee that such a model will be also optimally predictive. The essential requirement for a SAR or QSAR model is its ability to provide reliable predictions for new molecules. The simple data fitting ability with a low error is not enough for an SVM model, because such models are usually overfitted and give bad predictions. The experimental data used in SAR and QSAR are affected by experimental errors and sometimes by the fact that good structural descriptors are missing from the model. Therefore it is important to evaluate a number of kernels in order to find the most predictive SVM model and to avoid fitting the noise or the errors from the data. To offer a comparative evaluation of several kernels we review here a number of SVM classification and regression models in which five kernels are compared, namely linear, polynomial, Gaussian RBF, tanh, and anova. Each kernel was evaluated for a large range of parameters, and all results are compared for cross-validation. The SVM models were computed with mySVM, by Rüping (http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html).

The carcinogenic activity of a group of 46 methylated and 32 non-methylated polycyclic aromatic hydrocarbons was modeled with semiempirical quantum indices [61]. Form the set of 78 chemicals, 34 are carcinogenic and 44 are non-carcinogenic. The accuracy for leave-10%-out (L10%O) cross-validation tests shows that the best predictions are obtained with the Gaussian RBF kernel: RBF, $\sigma = 0.5$, $AC = 0.86$; anova, $\gamma = 0.5$, $d = 1$, $AC = 0.84$; degree 2 polynomial, $AC = 0.82$; linear, $AC = 0.76$; tanh, $a = 2$, $b = 0$, $AC = 0.66$. All SVM classification models were obtained with $C = 10$. The increase in prediction ac-



Drug Design with Machine Learning, Figure 15
SVM regression models ($\varepsilon = 0.2$): a anova RBF kernel, $\sigma = 0.2$, degree $= 5$, $C = 1000$; b spline kernel, $C = 0.05$; c Bessel kernel, $\sigma = 1$, order $= 1$, degree $= 1$, $C = 1000$; c Bessel kernel, $\sigma = 0.6$, order $= 1$, degree $= 5$, $C = 1000$

curacy for RBF and anova kernel, compared with the linear SVM, indicates that there is a nonlinear relationship between the quantum indices and hydrocarbon carcinogenicity.

A structure-odor classification was performed for 98 tetra-substituted pyrazines representing three odor classes, namely 32 green, 23 nutty, and 43 bell-pepper [62]. L10%O predictions were obtained for three classification tests, in which one odor type is selected as class +1 and the remaining two odor types form class −1. For green aroma compounds the best predictions are obtained with a polynomial kernel: degree 2 polynomial, $C = 1000$, $AC = 0.86$; anova, $C = 100$, $\gamma = 0.5$, $d = 1$, $AC = 0.84$; linear, $C = 10$, $AC = 0.80$; RBF, $C = 10$, $\sigma = 0.5$, $AC = 0.79$; tanh, $C = 10$, $a = 0.5$, $b = 0$, $AC = 0.73$. For nutty aroma compounds the anova kernel has slightly better predictions: anova, $C = 100$, $\gamma = 0.5$, $d = 1$, $AC = 0.92$; linear, $C = 10$, $AC = 0.89$; degree 2 polynomial, $C = 10$, $AC = 0.89$; RBF, $C = 10$, $\sigma = 0.5$, $AC = 0.89$; tanh, $C = 100$, $a = 0.5$, $b = 0$, $AC = 0.79$. The SVM models for bell-pepper aroma compounds show that three kernels (RBF, polynomial, and anova) give good predictions: RBF, $C = 10$, $\sigma = 0.5$, $AC = 0.89$; degree 2 polynomial, $C = 10$, $AC = 0.88$; anova, $C = 10$, $\gamma = 0.5$, $d = 1$, $AC = 0.87$. linear, $C = 10$, $AC = 0.74$; tanh, $C = 100$, $a = 2$, $b = 1$, $AC = 0.68$. In all three SVM experiments the best model outperformed the linear kernel, showing that nonlinear relationships are useful in structure-aroma studies.

The relationships between the structure of a chemical

compounds and their aqueous toxicity is highly relevant because many chemicals are environmental pollutants. The mechanism of toxic action (MOA) of organic chemicals may be predicted from hydrophobicity and experimental toxicity against *Pimephales promelas* and *Tetrahymena pyriformis* [67]. SVM classification was used to discriminate between 126 nonpolar narcotics and 211 chemicals that have other MOA. The RBF, anova, linear and polynomial kernels have similar statistics: RBF, $\sigma = 1$, $AC = 0.80$; anova, $\gamma = 0.5$, $d = 2$, $AC = 0.80$; linear, $AC = 0.79$; degree 2 polynomial, $AC = 0.79$; tanh, $a = 0.5$, $b = 0$, $AC = 0.53$. These results indicate that there is a linear relationship between descriptors and MOA. A similar conclusion was obtained for the MOA classification of polar and nonpolar narcotic compounds [66]. In a related study for the classification of MOA for narcotic and reactive compounds [63], the degree 2 polynomial offered the best predictions: degree 2 polynomial, $C = 10$, $AC = 0.92$; anova, $C = 10$, $\gamma = 0.5$, $d = 1$, $AC = 0.87$; linear, $C = 1000$, $AC = 0.86$; RBF, $C = 100$, $\gamma = 0.5$, $AC = 0.83$; tanh, $C = 10$, $a = 0.5$, $b = 0$, $AC = 0.78$.

SVM regression is demonstrated for QSAR models obtained for 52 benzodiazepine receptor ligands [68]. Five structural descriptors were used as SVM input, and the prediction was evaluated with leave-5%-out (L5%O) and L10%O cross-validation. A number of 34 experiments were conducted with five kernels (Tab. 1). The predictions obtained with the linear kernel L are used as baseline to compare the results provided by the other kernels. All four polynomial kernels P give very bad predictions, as indicated by the statistical indices. The RBF kernel R has better predictions than L only for $\sigma = 0.25$ (experiment 6), whereas all the other RBF models are worse than L. Overall, the best results are obtained with the tanh kernel T for $a = 0.5$ and $b = 0$ (experiment 11) which is a clear improvement compared to the linear kernel. Finally, the anova kernel A has uneven predictions. Some are good (experiments 22 and 20) whereas experiments 25–34 have very bad predictions. The SVM regression experiments presented in Tab. 1 show that kernel parameters are of paramount importance in obtaining a predictive QSAR.

In this section we presented comparative SAR and QSAR models obtained with five kernels. In developing SVM models it is important to include the linear kernel as a baseline comparison. If nonlinear kernels do not improve the predictions of the linear kernel, than the structure-activity relationship is linear and a linear SVM should be used. The parameters of nonlinear kernels should be optimized in order to obtain the best prediction. The most predictive SAR and QSAR model may be obtained with anyone of the five kernels investigated, and there is no apparent rule that can predict which kernel is the best. Therefore, it is important to experiment with a large diversity of kernels in order to find a predictive SVM model.

Applications in Drug Design

Torsade de pointes (TdP) is an important adverse drug reaction that is responsible for almost one-third of all drug failures during drug development, and resulted in several drugs being withdrawn from the market. Yap et al compared several ML for their ability to separate TdP compounds from those that do not induce TdP [156]. The prediction accuracy shows that SVM is the most reliable ML: SVM with Gaussian RBF kernel 91.0%, *k*-NN 88.5%, probabilistic neural network 78.2%, and C4.5 decision tree 65.4%. Inhibitors of the hERG (human ether-a-go-go-related gene) potassium channel can lead to a prolongation of the QT interval that can trigger TdP, an atypical ventricular tachycardia. A dataset of 39 hERG inhibitors (26 for learning and 13 for test) was evaluated by a pharmacophore ensemble coupled with SVM regression [87]. The pharmacophore ensemble encodes the protein conformational flexibility, and the SVM brings the capability of modeling nonlinear relationships. The QSAR model has very good statistics, with $q^2 = 0.89$ and $r^2 = 0.94$ for the test set. A structure-activity SVM study for hERG inhibitors was performed by Tobita et al for 73 drugs [132]. Using a combination of structural descriptors and molecular fragments, the SVM model had a prediction accuracy higher than 90%. Selecting an optimum group of descriptors is an important phase in computing a predictive structure-activity model. The SVM recursive feature elimination algorithm was evaluated by Xue et al for three SAR datasets, namely *P*-glycoprotein substrates, human intestinal absorption, and compounds that cause TdP [153]. The prediction statistics show that a proper descriptor selection may significantly increase the SVM performance.

The drug metabolism by human cytochrome P450 isoforms 3A4, 2D6, and 2C9 was studied by Terfloth et al for 379 drugs and drug analogs [131]. The best SVM model has a cross-validation accuracy of 89% and 83% accuracy for 233 validation compounds. Unbalanced data sets, with a majority of inactive compounds and only a few active chemicals, represent the usual situation in library screening. To overcome the ML bias toward the larger class, Eitrich et al used SVM with oversampling to identify cytochrome P450 2D6 inhibitors [37]. An ensemble of SVM models was presented by Yap and Chen as suitable classifiers for inhibitors and substrates of cytochromes P450 3A4, 2D6, and 2C9 [155]. Descriptors computed

Drug Design with Machine Learning, Table 1
Support vector regression statistics for benzodiazepine receptor ligands

Exp	Kernel	p_1	p_2	$q^2_{L5\%O}$	$RMSE_{L5\%O}$	$q^2_{L10\%O}$	$RMSE_{L10\%O}$
1	L			0.261	0.98	0.273	0.97
2	P	2		< -100	> 10	< -100	> 10
3	P	3		< -100	> 10	< -100	> 10
4	P	4		< -100	> 10	< -100	> 10
5	P	5		< -100	> 10	< -100	> 10
6	R	0.25		0.368	0.91	0.370	0.91
7	R	0.5		0.226	1.00	0.324	0.94
8	R	1.0		0.221	1.01	0.310	0.95
9	R	1.5		0.236	1.00	0.258	0.98
10	R	2.0		0.208	1.01	0.205	1.02
11	T	0.5	0.0	0.453	0.84	0.498	0.81
12	T	1.0	0.0	0.416	0.87	0.411	0.87
13	T	2.0	0.0	0.396	0.89	0.394	0.89
14	T	0.5	1.0	0.070	1.10	0.120	1.07
15	T	1.0	1.0	0.389	0.89	< -10	5.36
16	T	2.0	1.0	0.297	0.96	0.348	0.92
17	T	0.5	2.0	−0.365	1.33	−0.405	1.35
18	T	1.0	2.0	0.106	1.08	0.243	0.99
19	T	2.0	2.0	0.345	0.92	0.376	0.90
20	A	0.25	1	0.397	0.89	0.377	0.90
21	A	0.5	1	0.324	0.94	0.331	0.93
22	A	1.0	1	0.398	0.88	0.412	0.87
23	A	1.5	1	0.299	0.95	0.374	0.90
24	A	2.0	1	0.293	0.96	0.339	0.93
25	A	0.25	2	−1.289	1.73	−0.921	1.58
26	A	0.5	2	−1.375	1.76	−0.873	1.56
27	A	1.0	2	−0.606	1.44	−0.465	1.38
28	A	1.5	2	−0.241	1.27	−0.195	1.25
29	A	2.0	2	−0.071	1.18	−0.060	1.17
30	A	0.25	3	−2.998	2.28	−1.934	1.95
31	A	0.5	3	−1.282	1.72	−0.983	1.61
32	A	1.0	3	−0.060	1.17	−0.094	1.19
33	A	1.5	3	0.060	1.11	−0.097	1.19
34	A	2.0	3	−0.062	1.18	−0.073	1.18

with Dragon were used to develop predictive models, with MCC (Matthews correlation coefficient) statistics, namely 899 for 3A4, 0.884 for 2D6, and 0.872 for 2C9. The classification of of cytochrome P450 3A4 inhibitors and non-inhibitors was investigated by Arimoto et al with SVM, *k*-NN, recursive partitioning, logistic regression, and Bayesian classifier [7]. The chemical structure of the 4470 compounds was encoded with fingerprints and MolconnZ topological indices, and the best predictions were obtained with SVM trained with fingerprints. A comparison of single and ensemble classifiers was made by Merkwirth et al for SVM, *k*-NN, and ridge regression [95]. The tests conducted for two libraries of chemicals, one for unspecific protein inhibition and another one for inhibitors of cytochrome P450 3A4, showed that single and ensemble SVM models give similar results, with better predictions than *k*-NN and ridge regression.

Du et al predicted the genotoxicity of 140 thiophene derivatives with SVM and linear discriminant analysis (LDA) [34]. Seven structural descriptors were used as input data for the classification models, and the SVM parameters were identified by grid search. The accuracy of the SVM model (92.9% in training and 92.6% in prediction) is significantly higher than the LDA accuracy (81.4% in

training and 85.2% in prediction). Recursive partitioning and SVM were compared in a test for mutagenicity prediction from substructure descriptors [92]. The prediction accuracy for 2199 mutagens is similar for SVM (81.4%) and recursive partitioning (80.2%).

SVM classification was compared with linear discriminant analysis in predicting drug bioavailability [149]. The learning set comprises 167 compounds characterized by five structural descriptors. The SVM parameters were optimized with grid search. The classification results indicate that SVM (accuracy 85.6%) provides better models for bioavailability compared to LDA (accuracy 72.4%). Sakiyama et al studied SAR models for the metabolic stability of drug candidates [113]. The structure-activity models were obtained with random forest, support vector machine, logistic regression, and recursive partitioning. Classification models were obtained for a collection of 1952 compounds characterized by 193 descriptors. All ML models had accuracy > 0.8, with slightly higher results for random forest and SVM.

The identification of drug-like compounds based on atom type counts was investigated with SVM, linear programming machines, linear discriminant analysis, bagged k-nearest neighbors, and bagged decision trees C4.5 [97]. The classification dataset consisted of drug-like compounds from World Drug Index and non-drug compounds from Available Chemicals Directory. The smallest errors were obtained with SVM trained with polynomial and Gaussian RBF kernels. A computational filter for chemical compounds with antidepressant activity was developed by Lepp et al based on 21 biological targets related to depression [88]. An SVM model obtained with atom type descriptors gave high enrichment levels, and showed that a compound selected as active interacts on average with 2.3 targets. SVM, artificial neural networks, and multiple linear regression were compared in a model for apoptosis induction by 4-aryl-4-H-chromenes [40]. A leave-one-out cross-validation test showed that SVM gives the best predictions. Briem and Günther used a dataset of 565 kinase inhibitors and 7194 inactive compounds to compare four machine learning algorithms, namely SVM with a Gaussian RBF kernel, artificial neural networks, k-nearest neighbors, and recursive partitioning [23]. The best predictions were obtained with SVM, followed closely by k-NN. The SVM ability to identify active compounds from a chemical library was investigated by Jorissen and Gilson for sets of molecules that interact with the α_{1A} adrenoceptor, and cyclin-dependent kinase, cyclooxygenase-2, factor Xa, and phosphodiesterase-5 [72]. A comparison over ten QSAR datasets was performed to compare the stochastic gradient boosting method (SGB, an ensemble of classification and regression trees) with decision tree, random forest, partial least squares, k-NN, Bayes classifier, and SVM [129]. The best results predictions were obtained with random forest, followed by SVM and SGB.

The enzyme-substrate interaction is very specific, which points to the existence of recognition elements located on the enzyme interface. A surface patch ranking (SPR) method was proposed for the identification of those protein residues that determine the substrate specificity [157]. SPR incorporates an SVM module that can highlight the residues important in ligand binding. The method was applied with good results for several homologous enzymes, namely guanylyl/adenylyl cyclases, lactate/malate dehydrogenases, and trypsin/chymotrypsin. Potential applications are residue selection for mutagenesis experiments and functional annotation of proteins. Single base DNA mutations that result in an amino acid substitution are a common cause of monogenic disease. These substitutions reduce the protein stability, as measured by reduction in hydrophobic area, overpacking, backbone strain, and loss of electrostatic interactions [158]. An SVM classifier was trained with a set of mutations causative of disease and another set of non-disease causing mutations. The cross-validation tests show that the SVM identifies 74% of disease mutations, and confirms that loss of protein stability is a deteminant factor in monogenic disease.

Comparative Studies

SAR and QSAR models have two major components, namely a set of structural descriptors and a machine learning or ensemble of ML algorithms. The studies reviewed in previous sections show that there is no universal family of descriptors that gives the best predictions for any molecular property, which justifies the continuous development of novel structural descriptors. Similarly, no ML algorithm performs optimally for any property and any set of descriptors. Extensive computational experiments show that for a given property, the best model can be identified only by an empirical comparison of a large number of ML methods. In this section we review several studies that sought to compare the predictive abilities of ML models.

The aqueous solubility of drugs and drug-related compounds was modeled by Schroeter et al with Gaussian process, random forest, SVM, and ridge regression [118]. The domain of applicability for each model was estimate by computing error levels. The aqueous solubility of 988 organic chemicals was modeled with four regression algorithms, namely partial least squares, random forest, SVM, and MLF artificial neural networks [104]. The best predic-

tions were obtained with random forest, which also gave good results for an external validation set of 330 chemicals. The melting temperatures of ionic liquids were modeled with SVM, associative neural networks, *k*-NN, partial least squares, multilayer feedforward (MLF) artificial neural networks, and multiple linear regression [145]. Slightly better results were obtained with SVM, associative neural networks, and MLF artificial neural networks.

The human serum protein binding of 808 compounds was simulated with multiple linear regression, MLF artificial neural networks, *k*-NN, and SVM [148]. An ensemble of MLF artificial neural networks provided the best predictions for an external validation set of 200 compounds. The blood-brain barrier permeability of 415 chemicals was modeled by Li et al with logistic regression, linear discriminate analysis, *k*-NN, C4.5 decision tree, probabilistic neural network, and SVM [89]. It was found that all ML models are improved by using a descriptor set selected with recursive feature elimination. Plewczynski et al compared SAR models for five biological targets obtained with SVM, random forest, artificial neural networks, *k*-NN, trend vectors, Bayesian classifier, and decision tree [107]. There are significant differences in performance depending on the target and objective, i. e., high enrichment or maximum number of actives. The toxicity risk assessment of organic chemicals was evaluated with SVM, *k*-NN, Bayesian classifier, and self-organizing maps (SOM) [147]. Both SOM and SVM can separate toxic from nontoxic compounds based on structural fragments. A new ML algorithm that shows promising results is kScore, which in several tests performed better than SVM, *k*-NN, recursive partitioning, artificial neural networks, Gaussian process, and Bayesian classifier [102].

A comparison of 21 machine learning algorithms is presented for data selected from the National Cancer Institute 60-cell line screening panel (NCI-60) [69]. The SMILES codes of the chemical compounds [130] were used to compute fingerprints with Open Babel (http://openbabel.org/). The descriptor selection (based on CfsSubsetEval and BestFirst) and all machine learning models were computed with Weka [41,151], and all prediction results are for 10-fold (leave-10%-out) cross-validation. The machine learning algorithms used are briefly listed here, using their notation in Weka: BayesNet, Bayesian network; NaiveBayes, naïve Bayesian classifier [71]; NaiveBayesUpdateable, naïve Bayesian classifier with estimator classes [71]; Logistic, logistic regression with a ridge estimator [86]; RBFNetwork, Gaussian radial basis function network; IBk, *k*-NN classifier with distance weight (NoW – no distance weight, $W(1/d)$ – weighted with $1/d$, $W(1-d)$ – weighted with $(1-d)$) and k an odd number between 1 and 9 [1]; KStar, K^* lazy learner with entropy-based distance function [26]; ADTree, alternating decision tree (search type: Eap, expand all paths; Ehp, expand the heaviest path; Ezp, expand the best z-pure path; Erp, expand a random path) [42]; DecisionStump, one-level binary decision tree with categorical or numerical class label; J48, C4.5 decision tree [108]; NBTree, decision tree with naïve Bayes classifiers at the leaves [82]; RandomForest, random forest (the number of random trees is between 10 and 50) [22]; RandomTree, a tree that considers k randomly chosen attributes at each node; REPTree, fast decision tree learner; ConjunctiveRule, conjunctive rule learner; DecisionTable, decision table majority classifier [81]; JRip, a propositional rule learner based on RIPPER [27]; OneR, rule classifier that uses the minimum-error attribute for prediction [56]; PART, a PART decision list that builds a partial C4.5 decision tree in each iteration and transforms the best leaf into a rule ; Ridor, a RIpple–DOwn Rule learner [43]; SVM, LibSVM. More details for each ML may be found in Weka.

The first series of SAR models for anticancer compounds considers the cell line NCI-H460 (lung large cell carcinoma) with 3599 chemicals (2049 in class + 1, 1550 in class −1) and 30 descriptors (Tab. 2). For an easier comparison of the ranking of all ML methods, the results are ordered after the Matthews correlation coefficient MCC. The prediction statistics show that *k*-NN, in its several variants evaluated here, constantly gives the best predictions. Three other ML with good predictions are KStar, SVM, and RandomForest.

The second group of structure-activity models is obtained for the cell line SF-268 (glioma) with 3721 chemicals (2020 in class + 1, 1701 in class −1) and 36 descriptors (Tab. 3). A *k*-NN classifier is in the first position, followed by SVM and other *k*-NN variants. In decreasing order of the predictions, the list continues with another lazy learning method, KStar, followed by RandomForest and J48.

The third set of experiments considers the cell line SK-MEL-5 (melanoma) with 3685 chemicals (2034 in class + 1, 1651 in class −1) and 30 descriptors (Tab. 4). All *k*-NN give good predictions, including the top five results. RandomForest ranks better compared with the previous two cell lines, and it is followed by SVM and KStar. As a conclusion of these three sets of experiments, we find that a small group of ML algorithms gives constantly the best predictions. The top results are obtained with *k*-NN, which has good statistics for the whole range of parameters tested here. This is an important result, because *k*-NN is a robust method that is much easier to compute than SVM. Three other ML algorithms have good predictions, namely SVM, KStar, and RandomForest. Although KStar

Drug Design with Machine Learning, Table 2
Machine learning anticancer SAR for lung large cell carcinoma NCI-H460

Exp	Model	TP_p	FN_p	TN_p	FP_p	Ac_p	MCC_p
1	IBk $k = 9W(1 - d)$	1386	663	1035	515	0.6727	0.3414
2	IBk $k = 5W(1 - d)$	1410	639	1011	539	0.6727	0.3383
3	IBk $k = 9$ NoW	1404	645	1015	535	0.6721	0.3378
4	IBk $k = 7W(1 - d)$	1386	663	1027	523	0.6705	0.3364
5	IBk $k = 9W(1/d)$	1399	650	1012	538	0.6699	0.3334
6	IBk $k = 5$ NoW	1433	616	983	567	0.6713	0.3324
7	KStar	1364	685	1034	516	0.6663	0.3299
8	IBk $k = 7$ NoW	1404	645	1001	549	0.6682	0.3290
9	SVM RBF $\sigma = 0.01$	1494	555	926	624	0.6724	0.3286
10	IBk $k = 7W(1/d)$	1396	653	1003	547	0.6666	0.3263
11	IBk $k = 3W(1 - d)$	1420	629	982	568	0.6674	0.3252
12	RandomForest $T = 40$	1509	540	902	648	0.6699	0.3217
13	IBk $k = 3$ NoW	1446	603	954	596	0.6669	0.3210
14	IBk $k = 5W(1/d)$	1408	641	985	565	0.6649	0.3210
15	RandomForest $T = 50$	1499	550	903	647	0.6674	0.3171
16	RandomForest $T = 30$	1499	550	902	648	0.6671	0.3164
17	IBk $k = 3W(1/d)$	1428	621	955	595	0.6621	0.3125
18	RandomForest $T = 20$	1487	562	889	661	0.6602	0.3021
19	IBk $k = 1$ NoW	1457	592	911	639	0.6580	0.3000
20	IBk $k = 1W(1/d)$	1457	592	911	639	0.6580	0.3000
21	IBk $k = 1W(1 - d)$	1457	592	911	639	0.6580	0.3000
22	RandomForest $T = 10$	1494	555	878	672	0.6591	0.2990
23	JRip	1568	481	810	740	0.6607	0.2972
24	J48	1481	568	884	666	0.6571	0.2959
25	ADTree Ehp	1536	513	836	714	0.6591	0.2956
26	ADTree Ezp	1536	513	836	714	0.6591	0.2956
27	ADTree Erp	1580	469	791	759	0.6588	0.2922
28	NaiveBayes	1130	919	1147	403	0.6327	0.2919
29	NaiveBayesUpdateable	1130	919	1147	403	0.6327	0.2919
30	ADTree Eap	1563	486	806	744	0.6582	0.2919
31	Logistic	1514	535	848	702	0.6563	0.2911
32	BayesNet	1256	793	1054	496	0.6418	0.2903
33	NBTree	1406	643	930	620	0.6491	0.2857
34	RBFNetwork	1578	471	769	781	0.6521	0.2774
35	PART	1404	645	917	633	0.6449	0.2766
36	REPTree	1551	498	785	765	0.6491	0.2723
37	DecisionTable	1549	500	752	798	0.6393	0.2507
38	RandomTree	1384	665	885	665	0.6305	0.2464
39	Ridor	1617	432	642	908	0.6277	0.2201
40	OneR	1745	304	427	1123	0.6035	0.1565
41	ConjunctiveRule	1899	150	163	1387	0.5729	0.0562
42	DecisionStump	2049	0	0	1550	0.5693	0.0000

is not used in drug design, the results reported here indicate that this ML algorithm should be added to the toolbox of structure-activity methods. We have to note the absence of the Bayesian classifiers, although they are the preferred method in some library screening experiments.

The studies reviewed in this section show that it is difficult to perform comprehensive comparisons for a large number of machine learning algorithms. To address this

Drug Design with Machine Learning, Table 3
Machine learning anticancer SAR for glioma SF-268

Exp	Model	TP_p	FN_p	TN_p	FP_p	Ac_p	MCC_p
1	IBk $k = 9W(1/d)$	1338	682	1177	524	0.6759	0.3530
2	SVM RBF $\sigma = 0.01$	1403	617	1118	583	0.6775	0.3513
3	IBk $k = 7W(1/d)$	1344	676	1165	536	0.6743	0.3490
4	IBk $k = 5W(1/d)$	1351	669	1152	549	0.6727	0.3449
5	IBk $k = 7W(1 - d)$	1325	695	1173	528	0.6713	0.3443
6	IBk $k = 9W(1 - d)$	1301	719	1185	516	0.6681	0.3395
7	IBk $k = 5W(1 - d)$	1330	690	1160	541	0.6692	0.3391
8	IBk $k = 5$ NoW	1372	648	1120	581	0.6697	0.3368
9	IBk $k = 7$ NoW	1353	667	1135	566	0.6686	0.3360
10	KStar	1307	713	1168	533	0.6651	0.3325
11	IBk $k = 3$ NoW	1396	624	1089	612	0.6678	0.3311
12	IBk $k = 3W(1 - d)$	1358	662	1121	580	0.6662	0.3304
13	IBk $k = 9$ NoW	1323	697	1150	551	0.6646	0.3298
14	RandomForest $T = 40$	1408	612	1074	627	0.6670	0.3287
15	RandomForest $T = 50$	1408	612	1072	629	0.6665	0.3275
16	RandomForest $T = 30$	1418	602	1062	639	0.6665	0.3269
17	IBk $k = 3W(1/d)$	1363	657	1108	593	0.6641	0.3254
18	J48	1376	644	1082	619	0.6606	0.3169
19	RandomForest $T = 20$	1403	617	1045	656	0.6579	0.3095
20	Logistic	1404	616	1028	673	0.6536	0.3003
21	NBTree	1240	780	1164	537	0.6461	0.2974
22	RandomForest $T = 10$	1411	609	1015	686	0.6520	0.2965
23	BayesNet	1220	800	1176	525	0.6439	0.2948
24	IBk $k = 1$ NoW	1396	624	1020	681	0.6493	0.2916
25	IBk $k = 1W(1/d)$	1396	624	1020	681	0.6493	0.2916
26	IBk $k = 1W(1 - d)$	1396	624	1020	681	0.6493	0.2916
27	ADTree Erp	1407	613	993	708	0.6450	0.2819
28	RBFNetwork	1519	501	884	817	0.6458	0.2800
29	NaiveBayes	1064	956	1272	429	0.6278	0.2790
30	NaiveBayesUpdateable	1064	956	1272	429	0.6278	0.2790
31	ADTree Ehp	1406	614	986	715	0.6428	0.2774
32	ADTree Ezp	1406	614	986	715	0.6428	0.2774
33	REPTree	1380	640	1009	692	0.6420	0.2771
34	JRip	1418	602	960	741	0.6391	0.2689
35	ADTree Eap	1443	577	936	765	0.6393	0.2684
36	PART	1328	692	1024	677	0.6321	0.2593
37	DecisionTable	1438	582	915	786	0.6324	0.2538
38	RandomTree	1357	663	984	717	0.6291	0.2510
39	Ridor	1705	315	630	1071	0.6275	0.2454
40	DecisionStump	674	1346	1414	287	0.5611	0.1877
41	ConjunctiveRule	665	1355	1419	282	0.5601	0.1869
42	OneR	1257	763	932	769	0.5883	0.1702

issue, we organized a competition for the ML evaluation in blind predictions, CoEPrA 2006 (Comparative Evaluation of Prediction Algorithms, http://www.coepra.org/). For each prediction task the participants received a training dataset (structure, descriptors, and class attribution or activity) and a prediction dataset (structure and descriptors). All predictions received before deadline were compared with the experimental results, and then the ML models

Drug Design with Machine Learning, Table 4
Machine learning anticancer SAR for melanoma SK-MEL-5

Exp	Model	TP_p	FN_p	TN_p	FP_p	Ac_p	MCC_p
1	IBk $k = 9W(1-d)$	1379	655	1119	532	0.6779	0.3541
2	IBk $k = 9$ NoW	1401	633	1099	552	0.6784	0.3532
3	IBk $k = 7$ NoW	1422	612	1081	570	0.6792	0.3531
4	IBk $k = 5W(1-d)$	1401	633	1095	556	0.6773	0.3508
5	IBk $k = 7W(1-d)$	1390	644	1104	547	0.6768	0.3506
6	RandomForest $T = 50$	1487	547	1019	632	0.6801	0.3504
7	IBk $k = 7W(1/d)$	1414	620	1079	572	0.6765	0.3479
8	IBk $k = 5W(1/d)$	1417	617	1074	577	0.6760	0.3465
9	IBk $k = 9W(1/d)$	1404	630	1084	567	0.6752	0.3458
10	RandomForest $T = 30$	1478	556	1019	632	0.6776	0.3456
11	IBk $k = 5$ NoW	1427	607	1063	588	0.6757	0.3451
12	RandomForest $T = 20$	1482	552	1013	638	0.6771	0.3443
13	RandomForest $T = 40$	1481	553	1013	638	0.6768	0.3437
14	SVM RBF $\sigma = 0.03$	1414	620	1061	590	0.6716	0.3373
15	KStar	1326	708	1119	532	0.6635	0.3279
16	RandomForest $T = 10$	1481	553	976	675	0.6668	0.3222
17	IBk $k = 3W(1-d)$	1390	644	1053	598	0.6630	0.3204
18	IBk $k = 3$ NoW	1416	618	1025	626	0.6624	0.3171
19	IBk $k = 3W(1/d)$	1393	641	1044	607	0.6613	0.3166
20	IBk $k = 1$ NoW	1427	607	987	664	0.6551	0.3005
21	IBk $k = 1W(1/d)$	1427	607	987	664	0.6551	0.3005
22	IBk $k = 1W(1-d)$	1427	607	987	664	0.6551	0.3005
23	BayesNet	1232	802	1132	519	0.6415	0.2901
24	PART	1348	686	1033	618	0.6461	0.2875
25	RBFNetwork	1541	493	857	794	0.6507	0.2856
26	NBTree	1313	721	1055	596	0.6426	0.2832
27	J48	1378	656	1000	651	0.6453	0.2831
28	Logistic	1426	608	958	693	0.6469	0.2830
29	JRip	1483	551	899	752	0.6464	0.2785
30	RandomTree	1401	633	963	688	0.6415	0.2731
31	NaiveBayes	1068	966	1224	427	0.6220	0.2698
32	NaiveBayesUpdateable	1068	966	1224	427	0.6220	0.2698
33	DecisionTable	1494	540	863	788	0.6396	0.2634
34	REPTree	1431	603	918	733	0.6374	0.2622
35	ADTree Erp	1436	598	889	762	0.6309	0.2478
36	ADTree Ehp	1429	605	892	759	0.6299	0.2459
37	ADTree Ezp	1429	605	892	759	0.6299	0.2459
38	ADTree Eap	1323	711	981	670	0.6252	0.2441
39	Ridor	1469	565	817	834	0.6204	0.2230
40	DecisionStump	1279	755	932	719	0.6000	0.1930
41	OneR	1279	755	932	719	0.6000	0.1930
42	ConjunctiveRule	1404	630	770	881	0.5900	0.1605

were ranked after different statistical indices (Tab. 5). The competition had four classification (C in Tab. 5) tasks and four regression (R in Tab. 5) tasks. All datasets consisted of peptides evaluated for their binding affinity to the major histocompatibility complex (MHC) proteins. To evaluate the general performance of the algorithms tested in competition, for each task we present the ML from the top five positions. In general, each task has a different win-

Drug Design with Machine Learning, Table 5
Machine learning algorithms with best results in the CoEPrA 2006 (comparative evaluation of prediction algorithms, http://www.coepra.org/) competition

Task	Rank	Machine Learning	Task	Rank	Machine Learning
C1	1	naïve Bayes	R1	1	LS-SVM linear kernel
	2	LS-SVM RBF kernel		2	random forest
	3	CART		3	Gaussian process
	4	C4.5		4	kernel PLS
	5	SVM string kernel		5	SVM RBF kernel
C2	1	SVM linear kernel	R2	1	kernel PLS
	2	kScore		2	kNN
	3	SVM RBF kernel		3	PLS
	4	ClassificationViaRegression		4	SVM string kernel
	5	random forest		5	kScore
C3	1	SVM linear kernel	R3	1	random forest
	2	LS-SVM RBF kernel		2	kernel PLS
	3	SVM		3	SVM string kernel
	4	C4.5		4	PLS
	5	random forest		5	Gaussian process
C4	1	LS-SVM quadratic kernel	R3D2	1	kernel PLS
	2	SVM with AA sequence		2	SVM string kernel
	3	SVM binary encoding of AA		3	random forest
	4	kScore		4	Gaussian process
	5	random forest		5	SVM binary encoding of AA

ner and order for the top ML. The most frequent ML is SVM, in several variants. The least squares SVM (LS-SVM) is particularly robust, insensitive to noise, and with good performance. Currently, LS-SVM is not used in virtual screening, but the results from the competition show that it should be included in the set of preferred methods. Other predictive ML methods are random forest, Gaussian process, and kScore. The CoEPrA offered interesting conclusions with direct implications in drug design, SAR, and QSAR. Similar experiments are necessary to evaluate datasets consisting of drugs and drug-like chemicals.

Future Directions

For a long period of time the machine learning selection available to the computational chemist was restricted to a small number of algorithms. However, recent publications explore a much larger diversity of statistical models, with the intention to help the drug discovery process with reliable predictions. As an example of the current trend, one should consider the fast transition of the support vector machines from small-scale experiments to the preferred method in many industrial applications. Following the SVM success, other kernel methods were adopted in chemoinformatics and drug design. The results reviewed here show that it is not possible to predict which combination of structural descriptors and machine learning will give the most reliable predictions for novel chemicals. Therefore, in order to identify good models, it is always a good idea to compare several types of descriptors and as many ML algorithms as computationally possible. Equally important is a good plan for the experiments, that includes a proper validation and whenever possible tests with external data that were not used in training or cross-validation. Of great importance is an objective and comprehensive evaluation of ML algorithms, which can be performed in settings similar with the CoEPrA competition. Such blind predictions may offer an unbiased ranking of the machine learning algorithms.

Bibliography

1. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6:37–66
2. Ajmani S, Jadhav K, Kulkarni SA (2006) Three-dimensional QSAR using the *k*-nearest neighbor method and its interpretation. J Chem Inf Model 46:24–31
3. Andres C, Hutter MC (2006) CNS permeability of drugs predicted by a decision tree. QSAR Comb Sci 25:305–309
4. Alpaydin E (2004) Introduction to machine learning. MIT Press, Cambridge, p 445

5. Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning. Artif Intell Rev 11:11–73
6. Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning for control. Artif Intell Rev 11:75–113
7. Arimoto R, Prasad MA, Gifford EM (2005) Development of CYP3A4 inhibition models: comparisons of machine-learning techniques and molecular descriptors. J Biomol Screen 10:197–205
8. Balaban AT, Ivanciuc O (1999) Historical development of topological indices. In: Devillers J, Balaban AT (eds) Topological indices and related descriptors in QSAR and QSPR. Gordon & Breach Science Publishers, Amsterdam, pp 21–57
9. Basak SC, Grunwald GD (1995) Molecular similarity and estimation of molecular properties. J Chem Inf Comput Sci 35:366–372
10. Basak SC, Bertelsen S, Grunwald GD (1994) Application of graph theoretical parameters in quantifying molecular similarity and structure-activity relationships. J Chem Inf Comput Sci 34:270–276
11. Basak SC, Bertelsen S, Grunwald GD (1995) Use of graph theoretic parameters in risk assessment of chemicals. Toxicol Lett 79:239–250
12. Bayes T (1763) An essay towards solving a problem in the doctrine of chances. Philos Trans Roy Soc London 53:370–418
13. Bender A, Jenkins JL, Glick M, Deng Z, Nettles JH, Davies JW (2006) "Bayes affinity fingerprints" improve retrieval rates in virtual screening and define orthogonal bioactivity space: when are multitarget drugs a feasible concept? J Chem Inf Model 46:2445–2456
14. Bender A, Scheiber J, Glick M, Davies JW, Azzaoui K, Hamon J, Urban L, Whitebread S, Jenkins JL (2007) Analysis of pharmacology data and the prediction of adverse drug reactions and off-target effects from chemical structure. Chem Med Chem 2:861–873
15. Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin, p 740
16. Bishop CM (1996) Neural networks for pattern recognition. Oxford University Press, Oxford, p 504
17. Boid DB (2007) How computational chemistry became important in the pharmaceutical industry. In: Lipkowitz KB, Cundari TR (eds) Reviews in computational chemistry, vol 23. Wiley, Weinheim, pp 401–451
18. Bonchev D (1983) Information theoretic indices for characterization of chemical structure. Research Studies Press, Chichester
19. Bonchev D, Rouvray DH (eds) (1991) Chemical graph theory. Introduction and fundamentals. Abacus Press/Gordon & Breach Science Publishers, New York
20. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proc of the 5th annual ACM workshop on computational learning theory. ACM Press, Pittsburgh, pp 144–152
21. Bottou L, Chapelle O, DeCoste D, Weston J (2007) Large-scale kernel machines. MIT Press, Cambridge, p 416
22. Breiman L (2001) Random forests. Mach Learn 45:5–32
23. Briem H, Günther J (2005) Classifying "kinase inhibitor-likeness" by using machine-learning methods. Chem Bio Chem 6:558–566
24. Cash GG (1999) Prediction of physicochemical properties from Euclidean distance methods based on electrotopological state indices. Chemosphere 39:2583–2591
25. Chapelle O, Haffner P, Vapnik VN (1999) Support vector machines for histogram-based image classification. IEEE Trans Neural Netw 10:1055–1064
26. Cleary JG, Trigg LE (1995) K^*: an instance-based learner using and entropic distance measure. In: Prieditis A, Russell SJ (eds) Proc of the 12th international conference on machine learning. Morgan Kaufmann, Tahoe City, pp 108–114
27. Cohen WW (1995) Fast effective rule induction. In: Prieditis A, Russell SJ (eds) Proc of the 12th international conference on machine learning. Morgan Kaufmann, Tahoe City, pp 115–123
28. Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20:273–297
29. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines. Cambridge University Press, Cambridge
30. Deconinck E, Zhang MH, Coomans D, Vander Heyden Y (2006) Classification tree models for the prediction of blood-brain barrier passage of drugs. J Chem Inf Model 46:1410–1419
31. Deng Z, Chuaqui C, Singh J (2006) Knowledge-based design of target-focused libraries using protein-ligand interaction constraints. J Med Chem 49:490–500
32. Doddareddy MR, Cho YS, Koh HY, Kim DH, Pae AN (2006) In silico renal clearance model using classical Volsurf approach. J Chem Inf Model 46:1312–1320
33. Drucker H, Wu DH, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10:1048–1054
34. Du H, Wang J, Watzl J, Zhang X, Hu Z (2008) Classification structure-activity relationship (CSAR) studies for prediction of genotoxicity of thiophene derivatives. Toxicol Lett 177: 10–19
35. Duda RO, Hart PE, Stork DG (2000) Pattern classification. 2nd edn. Wiley, New York
36. Ehrman TM, Barlow DJ, Hylands PJ (2007) Virtual screening of chinese herbs with random forest. J Chem Inf Model 47:264–278
37. Eitrich T, Kless A, Druska C, Meyer W, Grotendorst J (2007) Classification of highly unbalanced CYP450 data of drugs using cost sensitive machine learning techniques. J Chem Inf Model 47:92–103
38. Ekins S, Balakin KV, Savchuk N, Ivanenkov Y (2006) Insights for human ether-a-go-go-related gene potassium channel inhibition using recursive partitioning and Kohonen and Sammon mapping techniques. J Med Chem 49:5059–5071
39. Ertl P, Roggo S, Schuffenhauer A (2008) Natural product-likeness score and its application for prioritization of compound libraries. J Chem Inf Model 48:68–74
40. Fatemi MH, Gharaghani S (2007) A novel QSAR model for prediction of apoptosis-inducing activity of 4-aryl-4-H-chromenes based on support vector machine. Bioorg Med Chem 15:7746–7754
41. Frank E, Hall M, Trigg L, Holmes G, Witten IH (2004) Data mining in bioinformatics using Weka. Bioinformatics 20:2479–2481
42. Freund Y, Mason L (1999) The alternating decision tree learning algorithm. In: Bratko I, Dzeroski S (eds) Proc of the 16th international conference on machine learning (ICML (1999)). Morgan Kaufmann, Bled, pp 124–133
43. Gaines BR, Compton P (1995) Induction of ripple-down rules

applied to modeling large databases. Intell J Inf Syst 5:211–228
44. Gao JB, Gunn SR, Harris CJ (2003) SVM regression through variational methods and its sequential implementation. Neurocomputing 55:151–167
45. Gao JB, Gunn SR, Harris CJ (2003) Mean field method for the support vector machine regression. Neurocomputing 50:391–405
46. Gepp MM, Hutter MC (2006) Determination of hERG channel blockers using a decision tree. Bioorg Med Chem 14:5325–5332
47. Guha R, Dutta D, Jurs PC, Chen T (2006) Local lazy regression: making use of the neighborhood to improve QSAR predictions. J Chem Inf Model 46:1836–1847
48. Gute BD, Basak SC (2001) Molecular similarity-based estimation of properties: a comparison of three structure spaces. J Mol Graph Modell 20:95–109
49. Gute BD, Basak SC, Mills D, Hawkins DM (2002) Tailored similarity spaces for the prediction of physicochemical properties. Internet Electron J Mol Des 1:374–387
50. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
51. Hansch C, Garg R, Kurup A, Mekapati SB (2003) Allosteric interactions and QSAR: on the role of ligand hydrophobicity. Bioorg Med Chem 11:2075–2084
52. Hastie T, Tibshirani R, Friedman JH (2003) The elements of statistical learning. Springer, Berlin, p 552
53. Herbrich R (2002) Learning kernel classifiers. MIT Press, Cambridge
54. Hert J, Willett P, Wilton DJ, Acklin P, Azzaoui K, Jacoby E, Schuffenhauer A (2006) New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. J Chem Inf Model 46:462–470
55. Hoffman B, Cho SJ, Zheng W, Wyrick S, Nichols DE, Mailman RB, Tropsha A (1999) Quantitative structure-activity relationship modeling of dopamine D_1 antagonists using comparative molecular field analysis, genetic algorithms-partial least-squares, and *K*-nearest neighbor methods. J Med Chem 42:3217–3226
56. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11:63–90
57. Hou T, Wang J, Zhang W, Xu X (2007) ADME evaluation in drug discovery. 7. Prediction of oral absorption by correlation and classification. J Chem Inf Model 47:208–218
58. Huang T-M, Kecman V, Kopriva I (2006) Kernel based algorithms for mining huge data sets. Springer, Berlin, p 260
59. Hudelson MG, Ketkar NS, Holder LB, Carlson TJ, Peng C-C, Waldher BJ, Jones JP (2008) High confidence predictions of drug-drug interactions: predicting affinities for cytochrome P450 2C9 with multiple computational methods. J Med Chem 51:648–654
60. Itskowitz P, Tropsha A (2005) *k*-nearest neighbors QSAR modeling as a variational problem: theory and applications. J Chem Inf Model 45:777–785
61. Ivanciuc O (2002) Support vector machine classification of the carcinogenic activity of polycyclic aromatic hydrocarbons. Internet Electron J Mol Des 1:203–218
62. Ivanciuc O (2002) Structure-odor relationships for pyrazines with support vector machines. Internet Electron J Mol Des 1:269–284
63. Ivanciuc O (2002) Support vector machine identification of the aquatic toxicity mechanism of organic compounds. Internet Electron J Mol Des 1:157–172
64. Ivanciuc O (2003) Graph theory in chemistry. In: Gasteiger J (ed) Handbook of chemoinformatics, vol 1. Wiley, Weinheim, pp 103–138
65. Ivanciuc O (2003) Topological indices. In: Gasteiger J (ed) Handbook of chemoinformatics, vol 3. Wiley, Weinheim, pp 981–1003
66. Ivanciuc O (2003) Aquatic toxicity prediction for polar and nonpolar narcotic pollutants with support vector machines. Internet Electron J Mol Des 2:195–208
67. Ivanciuc O (2004) Support vector machines prediction of the mechanism of toxic action from hydrophobicity and experimental toxicity against pimephales promelas and tetrahymena pyriformis. Internet Electron J Mol Des 3:802–821
68. Ivanciuc O (2005) Support vector regression quantitative structure-activity relationships (QSAR) for benzodiazepine receptor ligands. Internet Electron J Mol Des 4:181–193
69. Ivanciuc O (2005) Machine learning applied to anticancer structure-activity relationships for NCI human tumor cell lines. Internet Electron J Mol Des 4:948–958
70. Ivanciuc O (2007) Applications of support vector machines in chemistry. In: Lipkowitz KB, Cundari TR (eds) Reviews in computational chemistry, vol 23. Wiley, Weinheim, pp 291–400
71. John GH, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Besnard P, Hanks S (eds) UAI '95: Proc of the 11th annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, Montreal, pp 338–345
72. Jorissen RN, Gilson MK (2005) Virtual screening of molecular databases using a support vector machine. J Chem Inf Model 45:549–561
73. Jurs P (2003) Quantitative structure-property relationships. In: Gasteiger J (ed) Handbook of chemoinformatics, vol 3. Wiley, Weinheim, pp 1314–1335
74. Kier LB, Hall LH (1976) Molecular connectivity in chemistry and drug research. Academic Press, New York
75. Kier LB, Hall LH (1986) Molecular connectivity in structure-activity analysis. Research Studies Press, Letchworth
76. Kier LB, Hall LH (1999) Molecular structure description. The electrotopological state. Academic Press, San Diego
77. Klon AE, Diller DJ (2007) Library fingerprints: a novel approach to the screening of virtual libraries. J Chem Inf Model 47:1354–1365
78. Klon AE, Glick M, Davies JW (2004) Combination of a naive Bayes classifier with consensus scoring improves enrichment of high-throughput docking results. J Med Chem 47:4356–4359
79. Klon AE, Glick M, Thoma M, Acklin P, Davies JW (2004) Finding more needles in the haystack: a simple and efficient method for improving high-throughput docking results. J Med Chem 47:2743–2749
80. Klon AE, Lowrie JF, Diller DJ (2006) Improved naïve Bayesian modeling of numerical data for absorption, distribution, metabolism and excretion (ADME) property prediction. J Chem Inf Model 46:1945–1956
81. Kohavi R (1995) The power of decision tables. In: Lavrac N, Wrobel S (eds) ECML-95 8th european conference on machine learning, vol 912. Springer, Heraclion, pp 174–189

CE2 Please provide publishing location.

Editor's or typesetter's annotations (will be removed before the final TEX run)

82. Kohavi R (1996) Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Simoudis E, Han J, Fayyad UM (eds) Proc of the 2nd international conference on knowledge discovery and data mining (KDD-96). AAAI Press, pp 202–207 CE2
83. Kononenko I, Kukar M (2007) Machine learning and data mining: introduction to principles and algorithms. Horwood, Westergate, p 454
84. Konovalov DA, Coomans D, Deconinck E, Vander Heyden Y (2007) Benchmarking of QSAR models for blood-brain barrier permeation. J Chem Inf Model 47:1648–1656
85. Kumar R, Kulkarni A, Jayaraman VK, Kulkarni BD (2004) Structure-activity relationships using locally linear embedding assisted by support vector and lazy learning regressors. Internet Electron J Mol Des 3:118–133
86. le Cessie S, van Houwelingen JC (1992) Ridge estimators in logistic regression. Appl Statist 41:191–201
87. Leong MK (2007) A novel approach using pharmacophore ensemble/support vector machine (PhE/SVM) for prediction of hERG liability. Chem Res Toxicol 20:217–226
88. Lepp Z, Kinoshita T, Chuman H (2006) Screening for new antidepressant leads of multiple activities by support vector machines. J Chem Inf Model 46:158–167
89. Li H, Yap CW, Ung CY, Xue Y, Cao ZW, Chen YZ (2005) Effect of selection of molecular descriptors on the prediction of blood-brain barrier penetrating and nonpenetrating agents by statistical learning methods. J Chem Inf Model 45:1376–1384
90. Li S, Fedorowicz A, Singh H, Soderholm SC (2005) Application of the random forest method in studies of local lymph node assay based skin sensitization data. J Chem Inf Model 45:952–964
91. Li W-X, Li L, Eksterowicz J, Ling XB, Cardozo M (2007) Significance analysis and multiple pharmacophore models for differentiating *P*-glycoprotein substrates. J Chem Inf Model 47:2429–2438
92. Liao Q, Yao J, Yuan S (2007) Prediction of mutagenic toxicity by combination of recursive partitioning and support vector machines. Mol Divers 11:59–72
93. Mangasarian OL, Musicant DR (2000) Robust linear and support vector regression. IEEE Trans Pattern Anal Mach Intell 22:950–955
94. Mangasarian OL, Musicant DR (2002) Large scale kernel regression via linear programming. Mach Learn 46:255–269
95. Merkwirth C, Mauser HA, Schulz-Gasch T, Roche O, Stahl M, Lengauer T (2004) Ensemble methods for classification in cheminformatics. J Chem Inf Comput Sci 44:1971–1978
96. Mitchell TM (1997) Machine learning. McGraw-Hill, Maidenhead, p 432
97. Müller K-R, Rätsch G, Sonnenburg S, Mika S, Grimm M, Heinrich N (2005) Classifying 'drug-likeness' with kernel-based learning methods. J Chem Inf Model 45:249–253
98. Neugebauer A, Hartmann RW, Klein CD (2007) Prediction of protein-protein interaction inhibitors by chemoinformatics and machine learning methods. J Med Chem 50:4665–4668
99. Neumann D, Kohlbacher O, Merkwirth C, Lengauer T (2006) A fully computational model for predicting percutaneous drug absorption. J Chem Inf Model 46:424–429
100. Nidhi CE3, Glick M, Davies JW, Jenkins JL (2006) Prediction of biological targets for compounds using multiple-category Bayesian models trained on chemogenomics databases. J Chem Inf Model 46:1124–1133
101. Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JBO (2006) Melting point prediction employing *k*-nearest neighbor algorithms and genetic parameter optimization. J Chem Inf Model 46:2412–2422
102. Oloff S, Muegge I (2007) kScore: a novel machine learning approach that is not dependent on the data structure of the training set. J Comput-Aided Mol Des 21:87–95
103. Oloff S, Zhang S, Sukumar N, Breneman C, Tropsha A (2006) Chemometric analysis of ligand receptor complementarity: Identifying complementary ligands based on receptor information (CoLiBRI). J Chem Inf Model 46:844–851
104. Palmer DS, O'Boyle NM, Glen RC, Mitchell JBO (2007) Random forest models to predict aqueous solubility. J Chem Inf Model 47:150–158
105. Pelletier DJ, Gehlhaar D, Tilloy-Ellul A, Johnson TO, Greene N (2007) Evaluation of a published in silico model and construction of a novel Bayesian model for predicting phospholipidosis inducing potential. J Chem Inf Model 47:1196–1205
106. Platt J (1999) Fast training of support vector machines using sequential minimal optimization. In: Schölkopf B, Burges CJC, Smola AJ (eds) Advances in kernel methods – support vector learning. MIT Press, Cambridge, pp 185–208
107. Plewczynski D, Spieser SAH, Koch U (2006) Assessing different classification methods for virtual screening. J Chem Inf Model 46:1098–1106
108. Quinlan R (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Mateo
109. Ren S (2002) Classifying class I and class II compounds by hydrophobicity and hydrogen bonding descriptors. Environ Toxicol 17:415–423
110. Ripley BD (2008) Pattern recognition and neural networks. Cambridge University Press, Cambridge, p 416
111. Rodgers S, Glen RC, Bender A (2006) Characterizing bitterness: identification of key structural features and development of a classification model. J Chem Inf Model 46:569–576
112. Rusinko A, Farmen MW, Lambert CG, Brown PL, Young SS (1999) Analysis of a large structure/biological activity data set using recursive partitioning. J Chem Inf Comput Sci 39:1017–1026
113. Sakiyama Y, Yuki H, Moriya T, Hattori K, Suzuki M, Shimada K, Honma T (2008) Predicting human liver microsomal stability with machine learning techniques. J Mol Graph Modell 26:907–915
114. Schneider N, Jäckels C, Andres C, Hutter MC (2008) Gradual in silico filtering for druglike substances. J Chem Inf Model 48:613–628
115. Schölkopf B, Smola AJ (2002) Learning with kernels. MIT Press, Cambridge
116. Schölkopf B, Sung KK, Burges CJC, Girosi F, Niyogi P, Poggio T, Vapnik V (1997) Comparing support vector machines with gaussian kernels to radial basis function classifiers. IEEE Trans Signal Process 45:2758–2765
117. Schölkopf B, Burges CJC, Smola AJ (1999) Advances in kernel methods: support vector learning. MIT Press, Cambridge
118. Schroeter TS, Schwaighofer A, Mika S, ter Laak A, Suelzle D, Ganzer U, Heinrich N, Müller K-R (2007) Estimating the domain of applicability for machine learning QSAR models: a study on aqueous solubility of drug discovery molecules. J Comput-Aided Mol Des 21:485–498

CE3 Please provide author initials.

Editor's or typesetter's annotations (will be removed before the final TEX run)

119. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
120. Shen M, LeTiran A, Xiao Y, Golbraikh A, Kohn H, Tropsha A (2002) Quantitative structure-activity relationship analysis of functionalized amino acid anticonvulsant agents using *k*-nearest neighbor and simulated annealing PLS methods. J Med Chem 45:2811–2823
121. Shen M, Xiao Y, Golbraikh A, Gombar VK, Tropsha A (2003) Development and validation of *k*-nearest-neighbor QSPR models of metabolic stability of drug candidates. J Med Chem 46:3013–3020
122. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
123. Sommer S, Kramer S (2007) Three data mining techniques to improve lazy structure-activity relationships for noncongeneric compounds. J Chem Inf Model 47:2035–2043
124. Sorich MJ, McKinnon RA, Miners JO, Smith PA (2006) The importance of local chemical structure for chemical metabolism by human uridine 5′-diphosphate-glucuronosyltransferase. J Chem Inf Model 46:2692–2697
125. Sun H (2005) A naive Bayes classifier for prediction of multidrug resistance reversal activity on the basis of atom typing. J Med Chem 48:4031–4039
126. Suykens JAK (2001) Support vector machines: a nonlinear modelling and control perspective. Eur J Control 7:311–327
127. Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore
128. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP (2003) Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 43:1947–1958
129. Svetnik V, Wang T, Tong C, A Liaw, Sheridan RP, Song Q (2005) Boosting: an ensemble learning tool for compound classification and QSAR modeling. J Chem Inf Model 45:786–799
130. Swamidass SJ, Chen J, Phung P, Ralaivola L, Baldi P (2005) Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. Bioinformatics 21[S1]:i359–i368
131. Terfloth L, Bienfait B, Gasteiger J (2007) Ligand-based models for the isoform specificity of cytochrome P450 3A4, 2D6, and 2C9 substrates. J Chem Inf Model 47:1688–1701
132. Tobita M, Nishikawa T, Nagashima R (2005) A discriminant model constructed by the support vector machine method for HERG potassium channel inhibitors. Bioorg Med Chem Lett 15:2886–2890
133. Todeschini R, Consonni V (2003) Descriptors from molecular geometry. In: Gasteiger J (ed) Handbook of chemoinformatics, vol 3. Wiley, Weinheim, pp 1004–1033
134. Tong W, Hong H, Fang H, Xie Q, Perkins R (2003) Decision forest: Combining the predictions of multiple independent decision tree models. J Chem Inf Comput Sci 43:525–531
135. Tong W, Xie Q, Hong H, Shi L, Fang H, Perkins R (2004) Assessment of prediction confidence and domain extrapolation of two structure-activity relationship models for predicting estrogen receptor binding activity. Env Health Perspect 112:1249–1254
136. Trinajstić N (1992) Chemical graph theory. CRC Press, Boca Raton
137. Urrestarazu Ramos E, Vaes WHJ, Verhaar HJM, Hermens JLM (1998) Quantitative structure-activity relationships for the aquatic toxicity of polar and nonpolar narcotic pollutants. J Chem Inf Comput Sci 38:845–852
138. Vapnik VN (1979) Estimation of dependencies based on empirical data. Nauka, Moscow
139. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
140. Vapnik VN (1998) Statistical learning theory. Wiley, New York
141. Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10:988–999
142. Vapnik V, Chapelle O (2000) Bounds on error expectation for support vector machines. Neural Comput 12:2013–2036
143. Vapnik VN, Chervonenkis AY (1974) Theory of pattern recognition. Nauka, Moscow
144. Vapnik V, Lerner A (1963) Pattern recognition using generalized portrait method. Automat Remote Control 24:774–780
145. Varnek A, Kireeva N, Tetko IV, Baskin II, Solov'ev VP (2007) Exhaustive QSPR studies of a large diverse set of ionic liquids: how accurately can we predict melting points? J Chem Inf Model 47:1111–1122
146. Vogt M, Bajorath J (2008) Bayesian similarity searching in high-dimensional descriptor spaces combined with Kullback–Leibler descriptor divergence analysis. J Chem Inf Model 48:247–255
147. von Korff M, Sander T (2006) Toxicity-indicating structural patterns. J Chem Inf Model 46:536–544
148. Votano JR, Parham M, Hall LM, Hall LH, Kier LB, Oloff S, Tropsha A (2006) QSAR modeling of human serum protein binding with several modeling techniques utilizing structure-information representation. J Med Chem 49:7169–7181
149. Wang J, Du H, Yao X, Hu Z (2007) Using classification structure pharmacokinetic relationship (SCPR) method to predict drug bioavailability based on grid-search support vector machine. Anal Chim Acta 601:156–163
150. Watson P (2008) Naïve Bayes classification using 2D pharmacophore feature triplet vectors. J Chem Inf Model 48:166–178
151. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco, p 525
152. Xiao Z, Xiao Y-D, Feng J, Golbraikh A, Tropsha A, Lee K-H (2002) Antitumor agents. 213. Modeling of epipodophyllotoxin derivatives using variable selection *k*-nearest neighbor QSAR method. J Med Chem 45:2294–2309
153. Xue Y, Li ZR, Yap CW, Sun LZ, Chen X, Chen YZ (2004) Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents. J Chem Inf Comput Sci 44:1630–1638
154. Yamashita F, Hara H, Ito T, Hashida M (2008) Novel hierarchical classification and visualization method for multiobjective optimization of drug properties: application to structure-activity relationship analysis of cytochrome P450 metabolism. J Chem Inf Model 48:364–369
155. Yap CW, Chen YZ (2005) Prediction of cytochrome P450 3A4, 2D6, and 2C9 inhibitors and substrates by using support vector machines. J Chem Inf Model 45:982–992
156. Yap CW, Cai CZ, Xue Y, Chen YZ (2004) Prediction of torsade-causing potential of drugs by support vector machine approach. Toxicol Sci 79:170–177
157. Yu G-X, Park B-H, Chandramohan P, Munavalli R, Geist A, Samatova NF (2005) In silico discovery of enzyme-substrate specificity-determining residue clusters. J Mol Biol 352:1105–1117

158. Yue P, Li Z, Moult J (2005) Loss of protein structure stability as a major causative factor in monogenic disease. J Mol Biol 353:459–473
159. Zhang S, Golbraikh A, Oloff S, Kohn H, Tropsha A (2006) A novel automated lazy learning QSAR (ALL-QSAR) approach: method development, applications, and virtual screening of chemical databases using validated ALL-QSAR models. J Chem Inf Model 46:1984–1995
160. Zhang S, Golbraikh A, Tropsha A (2006) Development of quantitative structure-binding affinity relationship models based on novel geometrical chemical descriptors of the protein-ligand interfaces. J Med Chem 49:2713–2724
161. Zheng WF, Tropsha A (2000) Novel variable selection quantitative structure-property relationship approach based on the *k*-nearest-neighbor principle. J Chem Inf Comput Sci 40:185–194