Performance of Machine Learning Methods for Ligand-Based Virtual Screening

Dariusz Plewczynski[1], Stéphane A.H. Spieser[2] and Uwe Koch[*,2]

[1]*Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Pawinskiego 5a Street, 02-106 Warsaw, Poland*

[2]*Istituto di Ricerche di Biologia Molecolare P. Angeletti, Merck Research Laboratories, Via Pontina km 30600, Pomezia 00040, Italy*

Abstract: Computational screening of compound databases has become increasingly popular in pharmaceutical research. This review focuses on the evaluation of ligand-based virtual screening using active compounds as templates in the context of drug discovery. Ligand-based screening techniques are based on comparative molecular similarity analysis of compounds with known and unknown activity. We provide an overview of publications that have evaluated different machine learning methods, such as support vector machines, decision trees, ensemble methods such as boosting, bagging and random forests, clustering methods, neuronal networks, naïve Bayesian, data fusion methods and others.

Keywords: QSAR, machine learning, virtual screening, drug discovery.

INTRODUCTION

There are two major applications of virtual screening methods in drug discovery. One application is to eliminate compounds with undesirable properties and the other is to select compounds with an increased probability of activity on a specific target. Early recognition of compounds with undesirable features is driven by the fact that consecutive phases of the drug discovery pipeline become successively more expensive. Accordingly, it is advantageous to eliminate problematic compounds as early as possible. Since most pharmaceutical companies maintain compound acquisition programs or create large libraries in order to increase their compound collections, such filters can be applied before synthesis.

The second major application is to select for accurate biological testing a small subset of compounds from a much larger collection. This approach is particularly useful if there are resource constraints on accurate biological testing, in which case having accurate activity data for a smaller number of compounds enriched with actives can have advantages over less accurate results for a larger number of compounds. This type of focused screening requires the knowledge of some active reference compounds that are used to train the machine learning model. Compound selection and testing can be performed in an iterative fashion using the experimental results to refine the search model for the successive step.

Docking is another method for virtual screening that uses the target structure, usually a protein, to identify compounds that can fit into a binding pocket. Unlike ligand-based methods, docking does not require any *a priori* knowledge of active ligands. Both approaches have been compared, and the results confirm that ligand-based virtual screening is a valid approach that is often superior to docking in terms of performance [1].

*Address correspondence to this author at the IRBM, Via Pontina km 30,600, Pomezia 00040, Italy; Tel: +39 06 910 93644;
E-mail: uwe_koch@merck.com

Ligand-based approaches to design or identify novel active compounds exploit molecular similarity. Structurally similar compounds often have similar biological activity as stated by the similarity-property principle [2]. Datasets in which a set of compounds is tested against a constant panel of targets provide a complete structure-targets IC_{50} matrix confirming that structurally similar compounds often also exhibit a similar activity profile [3-5]. However, medicinal chemists also know that minor structural modifications to an active molecule can dramatically affect its activity. According to an analysis by Maggiora, structure-activity relationships (SAR) can be either continuous or discontinuous [6]. If the SARs are discontinuous similarity methods are likely to fail. While it is worthwhile bearing in mind that SAR can be discontinuous, this seems to be the exception rather than the rule and in general a similarity based virtual screen performs much better than a random selection.

The focus in this review is on evaluating the performance of machine learning methods for ligand-based virtual screening. Due to the limited space available, only a subset of representative publications comparing the performance of different methods will be discussed. We will give also a short discussion of each method with a focus on ways to fine-tune their performance.

DECISION TREE

Decision trees (DT) or recursive partitioning is a well established machine learning method with the advantage of being easily understandable and transparent [7,8]. DTs extract interpretable classification rules as the path through the tree from the root to the leaf [9-13]. DTs are employed to divide a large dataset into smaller and more homogenous sets. The method identifies recursively the feature that created the most diverse subsets according to the criterion chosen. The *t*-test is used for continuous response variables for feature selection to compare the groups with and without a given feature. The growing of the trees stops when the Bonferroni adjusted *p*-value for the test lacks significance.

1386-2073/09 $55.00+.00

Simple DTs usually perform better than more complex ones and are easier to interpret, which is why several strategies have been developed for pruning DTs. A common strategy is postpruning which overgrows the initial tree, eventually overfitting the training data and then prunes it back using cost complexity criteria. The pruning phase seeks a subtree that balances tree size and number of misclassifications or the mean square error on the training set. Other approaches for pruning include defining a minimum node size, a maximum tree depth or stopping when partition significance falls below a threshold.

The strength of DTs is in its capacity to handle very large and structurally diverse compound collections, to use large and heterogeneous descriptor sets, to ignore irrelevant descriptors and to generate a decision path for understanding the prediction of a test compound. A weakness of DTs is its relatively low prediction accuracy compared to other machine learning methods, and a number of extensions have been introduced to improve the prediction quality.

Decision trees have been widely applied in chemoinformatics in particular when the machine learning method had to cope with a very large number of descriptors. Applications include *in silico* ADME predictions [14,15], prediction of drug-likeness [16,17], analysis of large biological data sets [18], design of focused combinatorial libraries [19] and analysis of high throughput screening results [20].

ENSEMBLES OF CLASSIFIERS

An ensemble of classifiers combines several diverse machine learning algorithms (classifiers) to reduce the total error of classification. Diversity can be constructed by training machine learning algorithms on diverse training data sets, and this approach generates diverse decision rules for each classifier. Since errors are not the same for each classifier combination reduces the overall error due to cancellations of error. Several methods are used for combination of multiple classifiers - dynamic classifiers selection [21], fusion of classifiers [22], experts mixture [23], committees of neural networks [24], generalization and composite classification [25] schemes, to cite a few. A number of representative reviews cover this topic of machine learning [26,27].

Ensemble approaches differ both in the way by which individual classifiers are combined, or generated. In general there are two ways to combine classifiers: first by selection, and secondly by fusing them into a single predictor. In the case of classifiers selection, each algorithm is trained as an expert in the selected, local area in the entire feature space. The incoming sample is classified by those classifiers, which were trained on the most similar training examples to this query one. The final prediction is done by combining the weaker individual classifiers to obtain stronger single expert decisions. Such combinations can be done in various ways, for example by analysis of classification labels or rank ordering. The used combination schemes include ranking, voting, sum, products, posterior probabilities, integrals, fuzzy features, or decision templates. This combination of classifiers in order to improve the classifiers performance using a single training set is able to boost the overall classification accuracy significantly [27-29].

ENSEMBLE METHODS AND DECISION TREES

In this section the combination of ensemble methods and DTs will be shortly discussed since this combination is particularly popular in the context of chemoinformatics. Growing a DT is a hierarchical process in which the effect of an error at the top split is propagated down to all following splits, and as a consequence DTs are unstable predictors. Small modifications of the training data can cause large changes in the model structure and output. One method for avoiding this problem is to use several DTs or an ensemble. These ensemble techniques are also applicable to other learners. The predictive ability of an individual DT is improved by combining the predictions of multiple trees by training each tree on a different subset of the data.

Bagging and boosting are two ensemble techniques. Bagging was originally proposed by Leo Breiman [30] and is a type of the model averaging approach. It is often used with DTs but can, in principle, be used with any classifier. The bagging ensemble is formed by repeatedly selecting bootstrap samples of the dataset and training the trees on these data. Bootstrapping refers to the sampling of a dataset for training with replacement [31]. Boosting is another meta algorithm for supervised learning. AdaBoost is one of the earliest and most popular algorithm for boosting. It was formulated by Freund and Schapire [28,29,32]. Boosting typically adds incrementally weak learners to a final strong learner, and forms a "committee" by combining the outputs of many weak classifiers. In boosting, as iterations proceed, the focus turns to compounds that are difficult to classify. The accuracy of the previous tree determines the sample for the following tree concentrating on less well predicted compounds [33,34]. A weighing scheme is applied to each vote by taking into account the accuracy of each tree, and each following classifier focuses on those observations not yet classified in previous steps.

Gradient boosting creates a series of trees. The first tree is fitted to the data and the residuals or error values are then used for the construction of the second tree, using least-squares fitting to the residuals to reduce the error. This process is repeated many times and the final predicted value is obtained as the sum of the weighed contributions of each tree. Friedman introduced stochastic gradient boosting showing that the accuracy and execution speed is improved by introducing randomization into the procedure [35]. A randomly selected subset of the data is used at each iteration step to train the base learner.

Whereas the boosting methods described above create a series of trees in a sequential fashion Random Forest (RF) builds the trees in parallel and lets them vote on the prediction. The algorithm was first developed by Breiman and Cutler [36] combining Breiman's bagging and Ho's random subspace method [37]. For the construction of each tree RF selects the sample set of molecules with replacement from the original dataset for training. At each node a constant, small subset of descriptors (m_{try}) is chosen at random and used to define the best split on this node [11, 38]. Thus training starts with the selection of a bootstrap samples. For each sample a DT is constructed with a number of randomly selected descriptors until a predefined number of trees

achieved. Each tree is grown to the maximum size and no pruning is performed. Breiman shows that the error rate of the RF depends on the correlation between any two trees in the forest and on the error rate of each individual tree [38]. Reducing the number of descriptors selected (m_{try}) reduces correlation and increases the error rate of each individual tree. The optimal range of values is usually quite wide.

Tong *et al*. describe a different approach, decision forest [39], in which each model should make a unique contribution requiring each DT to be as different as possible from the others. In this case the descriptors used to grow a tree are removed from the descriptor set before growing the next tree. Thus each tree is built from a unique set of descriptors. Each tree assigns a probability of activity to a test compound and the ensemble prediction is the average probability over all trees.

Hawkins *et al*. introduced the Formal Inference-Based Recursive Modeling (FIRM) which is based on an ensemble of trees built on the same training data [40]. At each split point, a variable is randomly selected according to probabilities related to the variables significance from a statistical test defining the split. Random FIRM is mainly used for model interpretation and has been used for the analysis of chemical libraries [41].

DECISION TREES – HYBRID METHODS

Hybrid methods combine DTs with optimization procedures such as simulated annealing or other machine learning methods to select the optimal combination of descriptors for building the DT. DTs have been combined with *k*-nearest neighbors [42], artificial ant colony systems [43], simulated annealing [44] and evolutionary programming [45].

SUPPORT VECTOR MACHINE CLASSIFICATION

The basic concept of the support vector machine (SVM) algorithm is based on the statistical learning theory developed by Vapnik [46-48]. The application and theory of SVM is described in a number of excellent monographs [49-52] and reviews [53-58]. The training of a SVM is based on a set of learning data belonging to two different classes. For these data SVM constructs the maximal separating hyperplane separating the training objects into two classes. In many cases the data are not linearly separable. The SVM algorithm projects the input data vectors to a higher dimensional feature space using kernel functions [48,58], and a maximal separating hyperplane is then constructed in the feature space. Various kernel functions have been developed for this purpose and those relevant for chemical applications have been recently reviewed by Ivanciuc [58]. He enumerates also a set of guidelines for the selection of the kernel function, emphasizing the importance of comparing predictions from different kernels and combinations of parameters using the results from the linear kernel as a reference. If classification is possible with a linear kernel the use of non-linear kernels should be avoided. The radial basis function (RBF) kernel is not necessarily the best kernel for data with a non-linear relation between input data and class attribution. Overfitting of the training data can occur with SVM and is more likely to occur with more complex kernels. Ivanciuc demonstrates for a number of examples the possibility of overfitting and recommends comparing SVM models based on non-linear kernel functions with SVM models obtained with a linear kernel since the separating hypersurface may be almost linear [58].

Model selection, choosing the kernel and its parameters, is usually performed *via* cross-validation experiments that can be quite expensive. Interpretability of the model produced by SVM has not been a focus of research although there are some recent examples in this direction [59]. For a given set of parameters there is only one possible SVM classification system, whereas stochastic methods, such as artificial neural networks, produce different models under the same set of parameters, because they use a random number seed to generate the model. SVM has become one of the most popular machine learning methods in drug design, virtual screening and combinatorial chemistry [60-65].

ARTIFICIAL NEURAL NETWORKS

Another machine-learning algorithm used in chemoinformatics is the artificial neural network (ANN). The structure of ANNs is reminiscent of biological neural networks, namely many non-linear computational elements or nodes operate in parallel and are connected *via* weights that are modified during learning. ANNs are non-linear statistical data modeling tools that are often employed to identify patterns in data or to model complex relationships between input and output data. The theory and principles of ANN are reviewed in a number of books [66-70]. Many examples of applications in chemistry and medicine have been described [71-78].

The back propagation neural network using supervised learning is a particularly popular type of neural network often used for pattern and trend analysis. It is a feed-forward network in which the results of each layer are fed to the successive layer of nodes. It can use three or more layers passing the results of the first layer to the following layers. The difference between the output and the training set is used to iteratively adjust the connection weights [79], and the process is repeated until the difference falls within a predefined tolerance. The method is well established in drug discovery and chemistry [80-84].

A popular algorithm for unsupervised learning is the self-organizing map (SOM) developed by Kohonen [70,75], and consists of a single-layer feedforward network which is, unlike the feed-forward networks, organized as a grid. Since the number of input dimensions is usually higher than the number of output dimensions SOMs are often used for the reduction of the number of dimensions. SOMs are now used in different fields [85], such as drug design [86,87], toxicity prediction [88,89] and virtual screening [90-92].

NEAREST NEIGHBOR CLUSTERING

Clustering represents one of the most important unsupervised learning problems and has been reviewed in various books [93,94]. Different clustering techniques have been developed, and they can be separated into hierarchical clustering methods, like Ward's Clustering [95], and nonhierarchical clustering techniques such as Jarvis-Patrick clustering [96], *k*-means clustering [97], or Baye's unsupervised clustering [98].

The *k*-nearest neighbors (*k*NN) approach simultaneously evaluates a combination of descriptors without a defined

function; however, it assumes that each descriptor used in the model equally impacts the training activity. The activity of each compound is predicted as the average activity of *k* most chemically similar compounds from the data set [99]. Similarity is quantified using an appropriate distance metric. In order to avoid overtraining one needs to reduce the dimensionality of the feature space using for example only the most discriminatory descriptors. One approach is to combine the *k*NN with a genetic algorithm (GA). The algorithm optimizes the feature weights and the number of neighbors *k* in an evolution-like procedure. The evolutionary process is monitored by the implemented fitness-function that takes into account the overall prediction accuracy and the balance between classes. A modified version of the *k*-means clustering algorithm was developed to analyze large compound libraries to obtain an overview of the data distribution and inherent cluster structure [100]. Brown *et al.* have compared various clustering algorithms applied to chemical structures [101].

NAÏVE BAYESIAN

A Bayesian classifier is a classification method based on the Bayes rule [102] for conditional probability, *i.e.* the class (C) posterior probabilities given a feature vector X is equal to the class-conditional feature probability distribution times the prior class probability divided by the prior feature probability:

$$P(C = i \mid X = x) = P(X = x \mid C = i)P(C = i)/P(X = x)$$

Assuming that prior feature probabilities are identical for all classes, $P(X = x)$ can be neglected, and that features are independent given each class, yield the naïve Bayesian (NB) classifier:

$$P(C = i \mid X = x) = \Pi[P(X = xj \mid C = i)]P(C = i)$$

Basically NB relies only upon simple probability calculations, making it efficient for large datasets as execution time is fast and scaling is linear with respect to the number of molecules. NB weights features by assigning greater significance to features that appear to distinguish good samples from baseline samples. Finally NB does not require tuning which makes it robust and easy to use, though non-discrete features need to be discretized. Also, if discrete features have a large number of possible values, NB can easily run into trouble. NB suffers from its intrinsic simplicity in that it cannot analyze the combined effect of multiple descriptors [103,104].

TREND VECTOR

The trend vector method was first described by Carhart *et al.* [105] as one vector linking the center of gravity of inactive compounds toward the center of gravity of active compounds computed in an *N*-dimensional descriptor space. The authors compared trend vector calculations with the dipole moment calculation where activities replace charges and descriptor vectors replace atom coordinates. Thus the trend vector (TV) is the first moment of activity in descriptor space. Sheridan *et al.* [106] enhanced the trend vector calculation by expanding the set of descriptors (cross-terms) and introducing higher Partial Least Squares components (the trend vector being proportional to the first PLS component). A major advantage of the TV lies in its computational simplicity. There are no tunable parameters and the model building and application are extremely fast.

BINARY KERNEL DISCRIMINATION

Non-parametric kernel regression and binary kernel discrimination methods have recently been shown to compete favorably with other methods for virtual screening [107-109]. Compound classification in binary kernel discrimination is based on a binary description of its molecular structure. Probability distributions calculated from kernel density estimators are used to score novel compounds and classify them. The probability distributions are calculated for the training set containing a set of active and inactive compounds. These distributions are compared with those of a new set of compounds yielding an estimate of how likely the new compound is to be active. Usually binary kernel distribution is not used to extract information about the features that render a compound active.

DATA FUSION

Data fusion in general refers to the integration of information from various sources with the aim to improve the quality of the data [110,111]. In the case of ligand-based virtual screening data fusion usually refers to merging the results of similarity searches of a lead structure against a database of chemical compounds [112-114]. The individual hit lists can have been generated using different similarity measures, similarity fusion, or the same similarity measures but different machine learning methods, consensus scoring. Two different forms of similarity fusion can be distinguished. In one the similarity coefficient is the same but the representation different in the other a different similarity coefficient is used but with the same representation. Various types of similarity measures are used, starting from the Tanimoto coefficients, Cosine, Kulczynski, Baroni-Urbani, Pearson, Squared Euclidean, Russel-Rao, Simpson or Yule similarity coefficients [115]. Those similarity measures are then fused into single, consensus prediction of activity. The rational for similarity fusion using different molecular representations or descriptors is that different descriptors can capture different properties of the molecule. It has been shown that active compounds appearing top on the list of hits selected with one similarity measure can be far down on another [116]. In the group fusion method, similarity lists obtained from similarity searches using multiple reference structures of the same activity class and the same descriptors are combined to give a prediction [117]. Various rules exist for data fusion. The MIN rule takes the minimum rank a specific compound obtained in a series of ranked lists and the MAX rule takes the corresponding maximum ranking. Thus both methods are sensitive to extreme rankings. The SUM rule assigns to each compound the sum of the rankings it obtains in the various lists.

COMPARISON OF METHODS

In the following section we will discuss the performance of different machine learning methods. Various metrics have been used for evaluating the performance of prediction methods. Amongst the most common are accuracy, enrichment factor (EF), recall, precision and the area under the receiver operating characteristic curve (ROC). Of particular interest in the context of virtual screening are those com-

Table 1. Percentage of Correctly Predicted Molecules Using a 2.5D Descriptor and a Linear Fragment Descriptor (LFD) Dataset

	DT	Bagged DT	Boosted DT	RF	SVM	Tuned RF	Tuned SVM
2.5D	73.5	75.7	75.8	75.2	75.7	79.1	77.8
LFD	73.9	75.3	74.3	73.5	75.3	74.2	76.6

et al

pounds ranked early in an ordered list. The problems some of the performance metrics have in terms of measuring this "early recognition problem" have been noted and addressed by introduction of a new metric [118]. We concentrate on the performance related to the prediction quality and note other aspects of performance such as computational cost only occasionally.

SVM, DT, RF, BAGGING AND BOOSTING

Bruce *et al*. assessed different machine learning techniques applying rigorous statistical tests [119], with DT results based ensemble methods are compared with SVM results used as a benchmark. For SVM the authors evaluate two kernels, the polynomial and radial basis function (RBF), and try different values for the complexity constant, which controls the tolerance for misclassified molecules. Bruce *et al*. used eight data sets and two different types of descriptors, 2.5D descriptors and linear fragment descriptors to compare the following commonly used tree based ensemble techniques bootstrap, bagging, boosting and RF [119]. The performance of each method was measured as the percentage of correctly classified compounds based on a 10-fold cross-validation. All methods correctly classify between 67% and 90% of the molecules with significant differences between individual datasets (Table **1**), and with only one exception, the ensemble methods improve the performance compared to a single DT. Bagging, boosting and RF show comparable performances. Increasing the number of trees improves accuracy and the robustness of the prediction with convergence occurring in this case at around 100 trees.

SVM was used with a linear and non-linear kernel (tuned in Table **1**) and the complexity constant was modified. Switching from the linear to the non-linear kernel improved results for six out of eight datasets. In five of the eight datasets the 100-tree RF (tuned RF in Table **1**) gave better results than the non-linear SVM. The results however suggest that the difference in prediction quality between SVM and the tree-based ensemble methods is small. The authors conclude that, due to the large number of adjustable parameters, tuning of SVM can become challenging and there does not seem to be a single best set of parameters applicable to all test sets. Bruce *et al*. studied also the influence of increasing the number of descriptors available to the tree building algorithm when creating branching rules, and found that increasing the number of descriptors from the default value of six to seven descriptors per branch did not significantly improve prediction quality.

SVM, DT, RF, PARTIAL LEAST SQUARES (PLS), DECISION FOREST AND ANN

Svetnik *et al*. [11] compared the performance of non-optimized standard implementations of RF, DT and Partial Least Squares (PLS) for six chemoinformatics data sets (Table **2**). In terms of prediction performance RF ranks amongst the best algorithms improving results uniformly compared to DT. PLS comes close to RF except for two datasets on which it performs less well. For one dataset RF, SVM and ANN were compared showing superior performance for RF and SVM compared to ANN. On another dataset RF and Decision Forest showed similar performance. The authors analyzed also the influence of tuning $m_{try,}$, the subset of descriptors choosen to define the best split at each node, on the performance of RF. Their results show up to 5% improvement by reducing m_{try} from the bagging case in which m_{try} equals the number of descriptors. For $m_{try} = 1$ the error rate increased to 30%. They conclude that the performance of RF varies little over a wide range of values except near the extremes with the default values, $m_{try} = p^{1/2}$ for classification and $m_{try} = p/3$ for regression usually giving good results. The authors did not find an improved performance using descriptor selection in agreement with descriptor selection being intrinsic to the tree growing process.

Table 2. Average Accuracy and Standard Deviation (in Brackets) from 50 5-Fold Cross-Validations for Five Classification Data Sets Calculated from the Values in the Original Publication [11]

	RF	DT	PLS
Average	0.81 (0.02)	0.74 (0.03)	0.77 (0.05)

Decision forests were also tested on a dataset from a competitive estrogen receptor binding assay [39]. Compared with a standard DT the decision forest improved concordance (the number of compounds correctly classified divided by total compounds) by 5% to 97.8%.

SVM, DT, RF, BAGGING, BOOSTING, kNN, PLS AND NB

Svetnik *et al*. analyzed the performance of another ensemble technique, boosting [34], comparing stochastic gradient boosting (SGB) with a single DT, RF, kNN, partial least squares, NB and SVMs with both linear and radial kernels. The authors show that the performance of SGB is similar to RF and competitive or superior to other QSAR methods. When applied to six different data sets in five cases RF gives the highest accuracy. Also the accuracy averaged over all datasets is highest for RF, followed by SGB and SVM with a radial basis function (Table **3**). The authors conclude that both RF and SGB ensemble methods are superior to a single DT. SGB appears to perform somewhat better on large regression tasks whereas RF excels on classification, but the differences are small. More importantly, SGB requires more tuning including the optimization of the number of trees. On the other hand, RF training on large data sets can become

Table 3. Average Accuracy from 50 5-Fold Cross-Validations for Six Classification Data Sets Calculated from the Values in the Original Publication [11]

	RF	SGB	rbf_SVM	lin_SVM	PLS	kNN	DT	NB
Average	0.80	0.78	0.78	0.76	0.75	0.74	0.73	0.73
StDev	0.03	0.04	0.03	0.04	0.07	0.04	0.03	0.02

computationally very expensive, even prohibitive in some cases.

Banfield *et al.* compared eight different tree ensemble creation methods by applying them to 57 publicly available datasets [33]. Boosting, random subspaces, RFs and randomized C4.5 were compared with bagging. For 37 out of 57 tests none of the ensemble methods gave a statistically significant improvement over bagging. The best ensemble building approaches appear to be boosting with 1000 trees and a RF algorithm, RFs-1g. Bagging did not outperform these two methods in any dataset. Boosting with only 50 trees performs also well although performance is significantly reduced compared to boosting by resampling 1000 classifiers.

SVM, DT, RF, kNN, ANN, NB AND TV

Plewczynski *et al.* compared seven different QSAR algorithms by applying these to a selection of known ligands of five different target proteins [120]. The compounds were taken from the MDDR, encoded by atom pair descriptors and each method trained on a small subset and tested on a much larger, unrelated test set. Two datasets were used, one with a very small number of actives corresponding to validated actives in the MDDR (0.1 to 0.4% of all compounds) and another, more balanced dataset. Although Recall, Precision and Enrichment Factors were reduced for the data set containing less actives, in particular RF and SVM still showed a good performance (Table **4**). This observation is important since in most practical applications only a very small number of active compounds is known. In terms of Recall, avoiding false negatives, RF outperforms SVM only by a narrow margin. RF and SVM perform better than DTs and the other methods evaluated. Particularly high values for Precision are obtained with TV and RF.

In drug discovery the capability of a method to identify structurally new actives, called scaffold hopping, is a particularly desirable feature of any virtual screening method. Plewczynski *et al.* selected all compounds patented for each target and used either one third or two thirds of compounds patented first for training RF and SVM, and the remaining compounds, which were patented later, were used as the test set for the evaluation of the performance of the two methods [120]. The underlying assumption is that the compounds in a patent granted after an earlier patent of a competitor need to be structurally different from the earlier set. Atom pair descriptors were used in this study. The results in Table **5** show that SVM outperforms RF in terms of Recall whereas a reverse situation is found for Precision. Importantly, in each case SVM retrieves more than 50% of the actives of the test set of compounds, and the performance improves when the number of actives in the training set increases. Zhang *et al.* performed an extensive study on scaffold hopping comparing 2D and 3D descriptors as well as voting, ranking and consensus scoring [121], and found that a combination of atom pair descriptors and the maximum ranking method has in average the best performance.

SVM, DT AND NB

Glick *et al.* performed a retrospective analysis of four HTS data sets using DT, Laplacian modified NB and SVM [122]. SVM outperformed the other two methods in capturing actives in the top 1% of the ranked lists. In terms of the area under the ROC curve for one data set the Laplacian modified NB showed the best results whereas for the other data sets the performance of all methods was similar. The authors also analyzed the influence of noise on the performance of the three methods. Noise was added by randomly changing the annotation of inactive compounds to active and that of active compounds to inactive. All three methods were fairly tolerant to the addition of noise. No significant decrease in the accuracy was observed for a ratio of 1:1 active to misclassified compounds. SVM appeared to be most sensitive to addition of false negatives with the percentage of compounds captured at the top 1% decreasing from 43.8% to 30.5%.

SVM, kNN, PLS, DT AND ANN

An early direct application of SVM in drug design for

Table 4. Recall and Precision Averaged Over Datasets for Five Targets [120]

	SVM	RF	kNN	DT	ANN	TV	NB
Dataset I: Observed Actives 0.7 – 4.2%							
% Recall	92(4)	88 (6)	91 (4)	83 (7)	78 (14)	65 (21)	58 (30)
% Precision	66 (13)	87 (6)	49 (15)	41 (9)	60 (10)	87 (7)	32 (23)
Dataset II: Observed Actives 0.1 – 0.4%							
% Recall	69 (18)	45 (23)	61 (30)	45 (30)	15 (15)	38 (25)	25 (40)
% Precision	39 (14)	65 (22)	31 (10)	23 (12)	12 (15)	55 (28)	12 (4)

Standard deviations in brackets.

pharmaceutical data analysis proved that this classification method is very powerful in analyzing structure-activity relationships [123]. In a benchmark test (dihydrofolate reductase inhibition by pyrimidines obtained from the UCI repository) three artificial neural networks, a radial basis function network, and a DT were compared with SVM using the test errors averaged over five cross-validation folds. SVM and the neural network with manual capacity control performed best with test errors of 0.1269 and 0.1381 whereas the error rate of the DT was 0.l87. Holden and coworkers emphasize the importance of model selection in particular for the four different neural network architectures they have used. Each requires computationally intensive heuristic methods for model selection. The authors give the computer time for model selection as 1800 s for SVM and 20,715 s for the neural network with manual capacity control.

Table 5. Recall and Precision Averaged Over Datasets for Four Targets [120]

	Recall		Precision	
	1/3	2/3	1/3	2/3
SVM	48.2 (12.9)	59.4 (27.2)	52.3 (16.2)	36.2 (14.1)
RF	15.4 (10.0)	39.3 (14.9)	81.9 (24.5)	79.8 (23.5)

The training set contained either the one third (1/3) or the two thirds (2/3) of compounds patented first. Standard deviations in brackets.

Schneider and co-workers compared SVM and ANN in the context of drug/non-drug classification [124], as an example of binary decision problems in early-phase virtual compound filtering and screening. The results obtained by SVM seem to be more robust with a smaller standard error compared to neural network training. The SVM has higher accuracy of correct predictions irrespective of the type of descriptors used for molecule encoding (Ghose-Crippen fragment descriptors, different properties and physicochemical descriptors from the Molecular Operating Environment MOE package, topological pharmacophore CATS descriptors) or the size of the training data sets. In particular, the performance of SVM increases compared to ANN with an increasing number of features or descriptors. Thus SVM predicted 82% of the compounds correctly using all descriptors compared to 80% for ANN. For both methods classification accuracy increased with the number of training samples reaching a plateau after 2000-3000 samples. Yet, both methods were shown to complement each other, because the predicted sets of actives and negatives were not identical [124]. The average number of compounds correctly predicted by both methods as positives was 72% and the number of mutual false negatives was 11%, whereas 10% of the compounds correctly classified by SVM were not predicted correctly by ANN and for 6% the opposite was true.

The SVM algorithm was used in drug-likeness, agrochemical-likeness, and enzyme inhibition predictions [125]. In an experiment similar to the one described before with a different data set of drugs and non-drugs and using different descriptors in SVM with the RBF kernel achieves the highest accuracy (75.2%) followed by the ANN method with a multilayer perceptron and one hidden layer (72.5%) and SVM with a linear kernel (68.7%). Compared with neural networks, SVM with the RBF kernel outperforms them in the context of drug-likeness and agrochemical-likeness when using the same set of descriptors. The SVMs are also successfully applied to the problem of assessing the 'drug-likeness' of a molecule from a given set of descriptors with the error rate of about 7% on unseen compounds [126]. The various adverse drug reactions (ADRs), such as torsade de pointes [127], are important issues in the approval of drugs for certain diseases. The available experimental data provide an opportunity to the use of SVM for TdP prediction. TdP involves multiple mechanisms, therefore a set of linear solvation energy relationship (LSER) descriptors was used [128]. The accuracies for the SVM prediction of TdP-causing agents and non-TdP-causing agents are 97.4 and 84.6% respectively.

ANN and SVM were used to predict the blood-brain barrier permeability of different classes of molecules, and to develop a method to predict the ability of drug compounds to penetrate the CNS. The training data set consisted of 179 CNS active molecules and 145 CNS inactive molecules, and training parameters included molecular weight, lipophilicity, hydrogen bonding, and other characteristics that can affect the diffusion through a membrane. The SVM outperforms the neural network, predicting up to 96% of the molecules correctly, whereas the ANN average performance is 75.7% [129].

*k*NN was compared with SVM regarding the prediction of vitro cytogenetic results for a diverse set of 383 organic compounds using calculated molecular structure descriptors [130]. Each compound is represented by calculated molecular structure descriptors encoding the topological, electronic, geometrical, or polar surface area. The genetic algorithm was used for each machine learning classification to select the significant subsets of informative descriptors. The overall classification success rate for a *k*NN classifier built using six topological descriptors was 81.2% for the training set and 86.5% for test set. The overall classification success rate for a SVM model based on three descriptors was 99.7% for the training set, and 83.8% for an testing set [130].

Comparison of linear and nonlinear classification algorithms for the prediction of drug and chemical metabolism was performed on twelve isoforms of human UDP-glucuronosyltransferase (UGT) [131]. The authors compared partial least squares discriminant analysis (PLSDA), Bayesian regularized artificial neural network (BRANN), and support vector machine (SVM) methodologies by their ability to classify substrates and non-substrates using two-dimensional chemical descriptors. The SVM methodology was able to produce models with the best predictive performance, followed by BRANN and then PLSDA. SVM classification models showed extraordinary predictive accuracy reaching 60% of correct predictions over all test datasets. In the case of five test sets it reached 80% prediction accuracy. However, for some data sets PLSDA performed similar or even better, i.e. when linear relationships in the training data sets are observed [131]. In the similar approach multiple linear regression (MLR), radical basis function neural network (RBFNN) and support vector machine (SVM) were compared on natural, synthetic and environmental endocrine disrupting compounds for binding to the androgen receptor. Five chemical descriptors (hydrogen-

bonding interaction, distribution of atomic charges and molecular branching degree) were selected to build predictive QSAR models. The SVM method exhibited the best overall performances [132].

LINEAR DISCRIMINANT ANALYSIS, DT AND CLUSTERING

In an early study Dixon *et al*. analyzed the performance of linear discriminant analysis, DT, and hierarchical agglomerative clustering using topological descriptors on data sets for four different targets. Discriminant analysis yielded the smallest number of false negatives whereas DT performed best in terms of avoiding false positives. The authors also showed that replacing a single training set by an ensemble of smaller balanced training sets improved sensitivity towards active compounds, presumably because pharmaceutical databases tend to be skewed towards inactive compounds and models built from unbalanced datasets show low sensitivity for active compounds [45].

kNN AND DT

Application of *k*NN model to a set of compounds with anti-HIV activity using MDL MACCS keys as descriptors lead to results outperforming DTs in terms of enrichment [42], and a *k*NN combination with the artificial ant colony system [43] improved the prediction quality. Decision trees were combined with simulated annealing to select the best combination of descriptors at each step and then applied to identify structurally homogeneous classes of highly potent anticancer agents [44]. DeLisle used genetic algorithms to refine the selection of descriptors [133]. The method was tested on a hepatotoxicity dataset and improved accuracy by 5-10% compared to standard DT.

SOM, NB, *k*NN AND SVM

Self-organizing maps (SOM) were used for toxicity prediction based on substructure fragments of known toxicity from RTECS, IDDB databases and the World Drug Index [89]. The proper selection of toxicity indicating structural patterns allows for rapid toxicity risk assessment, even for previously untested molecules. The toxicity classification performance was compared to naïve Bayesian clustering, *k*NN, and SVMs. The authors found that a SVM performed best at classifying compounds of defined toxicity into appropriate toxicity classes, whereas SOM performed well in separating general toxic from nontoxic substances [89].

BKD, *k*NN, NB AND SVM

Harper *et al*. have used HTS data to show that binary kernel discrimination outperforms a merged similarity search as well as a neuronal net when applied to moderately noisy HTS data (ca 40% false positives) [109]. Chen *et al*. have shown that the optimal value of the smoothing parameter and thus the predictive power depend on the number of false positives in the training set [134]. The smoothing parameter determines how many nearest neighbors are used to predict the activity of a test molecule. The authors show that binary kernel discrimination performs well for high quality data for training with a value of $\lambda = 0.6$. If the training set contains a larger share of false positives then a much lower value is recommended. A comparison of different distance metrics leads to a ranking of 20 different similarity coefficients showing that Jaccard/Tanimoto performs best. In another publication Chen *et al*. compare the performance of continuous kernel discrimination based on the Parzen window method to SVM with a radial basis function [135]. Training and test sets were taken from eleven activity classes from the MDDR. Using physicochemical pipeline pilot descriptors CKD retrieves in average 56.2% of test set actives in the top 1% of the ranked test set and SVM 52.4% of the same set. The same authors also evaluate the performance of a naïve Bayesian classifier on data sets containing only a small number of actives. The results are compared to those obtained from group fusion applied to conventional similarity searches. Using the number of actives retrieved in the top 1% of the ranked data set as a performance measure depends on the diversity of the data set, with NB performing well on data sets containing low diversity activity classes and with similarity searches performing well on data sets with more diverse actives. Overall, NB performs well in reproducing the features of diverse inactives [122]. The authors conclude that NB can be expected to perform well on data sets with an active class of low diversity and inactive class of high diversity. Citing results from Bender [136] and Hert *et al*. [137] NB retrieves on average 64.9% of actives in the top 5% of ranked lists of 11 targets whereas similarity based group fusion retrieves 67.4%, thus showing that on data sets with few actives NB is not superior to group fusion.

Wilton *et al*. [138] compared trend vector versus binary kernel discrimination (BKD), similarity searching (*k*-nearest neighbor classifier), substructural analysis, and SVM, in the context of virtual screening. Two data sets from the NCI AIDS and the Syngenta corporate database were employed with structures encoded by two types of fragment bit-string and by sets of high-level molecular features. Trend vector showed more than acceptable results, being however consistently outperformed by other methods like binary kernel discrimination.

DATA FUSION

Willett and co-workers have studied the effect of combining thirteen different similarity measures in screening for active compounds belonging to seven different bioactivity classes. With one exception the searches with fused similarity coefficients results outperformed searches using a single similarity coefficient. However, the authors show that there is no single combination outperforming all others in each case [139].

Hert *et al*. used group fusion with ten active reference compounds on eleven different data sets based on the MAX fusion rule. Group fusion was compared to similarity searches using a single reference by considering each active compound from each activity class as a reference structure for a similarity search and then calculating the mean and maximum Recall for each search. The Recall value averaged over all eleven data sets is 53% for group fusion and 31% for the mean of all similarity searches using a single reference structure. Even more convincing is the comparison with the results obtained with the best possible similarity search for each activity class using a single reference structure. The average Recall obtained with the best possible similarity search is also 53% and thus comparable to the group fusion result. Whittle *et al*. [117] as well as Hert *et al*. [140] studied

the influence of diversity in reference structures on the performance of group and similarity fusion. They demonstrated that the advantage of group fusion is largest when diversity of the active reference compounds is large whereas similarity fusion performs well with actives showing a high degree of self-similarity.

The effect of combining classification lists obtained by using different machine learning methods on the same dataset and using the same descriptors was studied by Plewczynski *et al*. [120]. Compounds from five different bioactivity classes were classified using seven different methods. The classification lists were combined by calculating Recall and Precision on the basis of the compounds found to be active by at least one method, two methods, three methods and up to seven different methods. As can be seen in Table **6** combining the hit lists has significant consequences for Recall and Precision showing that even if the same dataset is used, different machine learning methods generate diverse lists. As a general rule and in agreement with intuition, accepting compounds identified by an increasing number of methods as actives will reduce the number of false positives thus improving Precision, whereas accepting all compounds identified by at least one method as active will reduce the number of false negatives at the cost of an increasing number of false positives.

Table 6. Recall and Precision Obtained from the Consensus Approach

Number Agreeing Methods	1	2	3	4	5	6	7
Recall	96%	93%	89%	86%	79%	64%	44%
Precision	23%	54%	75%	84%	89%	94%	99%

Each value is calculated on the basis of compounds predicted to be active by at least one, two, three etc. methods and averaged over five bioactivity classes.

CONCLUSIONS AND OUTLOOK

The examples discussed above show that in the context of chemoinformatics the application of machine learning offers significant advantages compared to a random selection. In practical terms this means that electronic filters and virtual screening can make important contributions to the drug discovery process, in particular in its early phase. The application of machine learning is particularly beneficial when the objective is to reduce a large dataset to a smaller chemical library. Particularly important in this context is the quality of the dataset that easily becomes the limiting factor for any type of machine learning.

There does not appear to be a single best method for each problem. Each method models different aspects of the structure activity relationship in the dataset particularly well. Depending on the dataset, most methods can be optimized by a variation of parameters or by the choice of the appropriate kernel. For this reason fusion methods which combine the results from different approaches usually improve the performance of the selection [120] and may become a convenient alternative to the adjustment of a single method to a given dataset. The applications discussed in this review show that ensemble methods on the basis of DTs, such as boosting and RF, outperform methods employing a single classifier. In most comparisons SVM performs very well and yields results comparable to RF or boosting. The prediction quality of SVM can often be improved by adjusting its parameters to the particular problem.

Since machine learning is a very active field of research, new methods and protocols will be developed outperforming established techniques. We expect data fusion approaches allowing incorporation of new methods to find more wide spread application in chemoinformatics. In general terms the increasing quantity of experimental data affected by noise will increase the acceptance of machine learning methods in order to extract useful information.

ACKNOWLEDGEMENTS

This work was supported in part by EC BioSapiens (LHSGCT-2003-503265) 6FP project as well as the Polish Ministry of Education and Science (N N301 159735).

REFERENCES

[1] McGaughey, G. B.; Sheridan, R. P.; Bayly, C. I.; Culberson, J. C.; Kreatsoulas, C.; Lindsley, S.; Maiorov, V.; Truchon, J. F.; Cornell, W. D. *J. Chem. Inf. Model.*, **2007**, *47*, 1504-1519.
[2] Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. *J. Med. Chem.*, **2002**, *45*, 4350-4358.
[3] Cleves, A. E.; Jain, A. N. *J. Med. Chem.*, **2006**, *49*, 2921-2938.
[4] Fliri, A. F.; Loging, W. T.; Thadeio, P. F.; Volkmann, R. A. *Proc. Natl. Acad. Sci. USA*, **2005**, *102*, 261-266.
[5] Krejsa, C. M.; Horvath, D.; Rogalski, S. L.; Penzotti, J. E.; Mao, B.; Barbosa, F.; Migeon, J. C. *Curr. Opin. Drug Discov. Dev.*, **2003**, *6*, 470-480.
[6] Maggiora, G. M. *J. Chem. Inf. Model.*, **2006**, *46*, 1535.
[7] Friedman, J. H. *IEEE Trans. Comput.*, **1977**, *26*, 404-408.
[8] Quinlan, J. R. *Mach. Learn.*, **1986**, *1*, 81-106.
[9] Amimoto, R.; Prasad, M. A.; Gifford, E. M. *J. Biomol. Screen.*, **2005**, *10*, 197-205.
[10] Burton, J.; Ijjaali, I.; Barberan, O.; Petitet, F.; Vercauteren, D. P.; Michel, A. *J. Med. Chem.*, **2006**, *49*, 6231-6240.
[11] Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 1947-1958.
[12] van Rhee, A. M. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 941-948.
[13] Zhang, Q. Y.; Aires-de-Sousa, J. *J. Chem. Inf. Model.*, **2007**, *47*, 1-8.
[14] van de Waterbeemd, H.; Gifford, E. *Nature Rev. Drug Discov.*, **2003**, *2*, 192-204.
[15] Xu, J.; Hagler, A. *Molecules*, **2002**, *7*, 566-600.
[16] Muegge, I. *Med. Res. Rev.*, **2003**, *23*, 302-321.
[17] Wagener, M.; van Geerestein, V. J. *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 280-292.
[18] Young, S. S. *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 1017-1026.
[19] Rusinko Iii, A.; Young, S. S.; Drewry, D. H.; Gerritz, S. W. *Comb. Chem. High Throughput Screen.*, **2002**, *5*, 125-133.
[20] Xue, L.; Bajorath, J. *Comb. Chem. High Throughput Screen.*, **2000**, *3*, 363-372.
[21] Woods, K.; Kegelmeyer, W. P.; Bowyer, K. *IEEE Trans. Pattern Anal. Machine Intell.*, **1997**, *19*, 405-410.
[22] Kuncheva, L. I. *IEEE Trans. Pattern Anal. Machine Intell.*, **2002**, *24*, 281-286.
[23] Jordan, M. I.; Jacobs, R. A. *Proceedings of 1993 International Joint Conference on Neural Networks, 1993.* IJCNN'93-Nagoya., **1993**, *2*.
[24] Krogh, A.; Vedelsby, J. *Adv. Neural Inf. Process. Syst.*, **1995**, *7*, 231-238.
[25] Gama, J. *AI Commun.*, **2000**, *13*, 135-136.
[26] Dietterich, T. G. *Lect. Notes Comput. Sci.*, **2000**, *1857*, 1-15.
[27] Opitz, D.; Maclin, R. *J. Artif. Intell. Res.*, **1999**, *11*, 12.
[28] Freund, Y. *Inform. Comput.*, **1995**, *121*, 256-285.
[29] Freund, Y.; Schapire, R. E. *Machine Learning: Proceedings of the Thirteenth International Conference*, **1996**, *148*, 156.
[30] Breiman, L. *Mach. Learn.*, **1996**, *24*, 123-140.

[31] Witten, I. H.; Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*; Elsevier: New York, **2005**.
[32] Schapire, R. E.; Freund, Y.; Bartlett, P.; Lee, W. S. *Ann. Statist.*, **1998**, *26*, 1651-1686.
[33] Banfield, R. E.; Hall, L. O.; Bowyer, K. W.; Kegelmeyer, W. P. *IEEE Trans. Pattern Anal. Machine Intell.*, **2007**, *29*, 173-180.
[34] Svetnik, V.; Wang, T.; Tong, C.; Liaw, A.; Sheridan, R. P.; Song, Q. *J. Chem. Inf. Model.*, **2005**, *45*, 786-799.
[35] Friedman, J. H. *Comput. Stat. Data Anal.* **2002**, *38*, 367-378.
[36] Breiman, L. *Mach. Learn.*, **2001**, *45*, 5-32.
[37] Ho, T. K. *IEEE Trans. Pattern Anal. Machine Intell.*, **1998**, *20*, 832-844.
[38] Breiman, L. *Mach. Learn.*, **2001**, *45*, 5-32.
[39] Tong, W.; Hong, H.; Fang, H.; Xie, Q.; Perkins, R. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 525-531.
[40] Hawkins, D. M. *Am. Stat.*, **1991**, *45*, 155-155.
[41] Young, S. S.; Hawkins, D. M. *J. Med. Chem.*, **1995**, *38*, 2784-2788.
[42] Miller, D. W. *J. Chem. Inf. Comput. Sci.*, **2001**, *41*, 168-175.
[43] Izrailev, S.; Agrafiotis, D. *J. Chem. Inf. Comput. Sci.*, **2001**, *41*, 176-180.
[44] Blower, P.; Fligner, M.; Verducci, J.; Bjoraker, J. *J. Chem. Inf. Comput. Sci.*, **2002**, *42*, 393-404.
[45] Dixon, S. L.; Villar, H. O. *J. Comput. Aided Mol. Des.*, **1999**, *13*, 533-545.
[46] Vapnik, V. *Estimation of Dependencies Based on Empirical Data*; Nauka: Moscow, **1979**.
[47] Vapnik, V. *The Nature of Statistical Learning Theory*, Springer: New York, **1995**.
[48] Vapnik, V. N. *Statistical learning theory. Adaptive and learning systems for signal processing, communications, and control*; Wiley: New York, **1998**.
[49] Christianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*; Cambridge University Press: Cambridge (UK), **2000**.
[50] Schölkopf, B.; Burges, C. J. C.; Smola, A. J. *Advances in kernel methods: support vector learning*; MIT Press Cambridge, MA, USA, **1999**.
[51] Scholkopf, B.; Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press Cambridge, MA: USA, **2001**.
[52] Smola, A. J.; Schölkopf, B. *Statistics and Computing*, **2004**, *14*, 199-222.
[53] Burges, C. J. C. *Data Mining Knowledge Discov.*, **1998**, *2*, 121-167.
[54] Byvatov, E.; Schneider, G. *Appl. Bioinformatics*, **2003**, *2*, 67-77.
[55] Cai, Y.D.; Liu, X.J.; Xu, X.; Zhou, G.P. *BMC Bioinformatics*, **2001**, *2*, 3.
[56] Cherkassky, V.; Ma, Y. *Neural Netw.*, **2004**, *17*, 113-126.
[57] Cristianini, N.; Schölkopf, B. *AI Magazine*, **2002**, *23*, 31-41.
[58] Ivanciuc, O. In *Reviews in Computational Chemistry*; T. R. Cundari K. B. Lipkowitz, Ed.; Wiley-VCH: Weinheim, **2007**; Vol. *23*, pp. 291-400.
[59] Byvatov, E.; Schneider, G. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 993-999.
[60] Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. *Comput. Chem.*, **2001**, *26*, 4-15.
[61] Doniger, S.; Hofmann, T.; Yeh, J. *J. Comput. Biol.*, **2002**, *9*, 849-864.
[62] Lengauer, T.; Lemmen, C.; Rarey, M.; Zimmermann, M. *Drug Discov. Today*, **2004**, *9*, 27–34.
[63] Trotter, M. W. B.; Buxton, B. F.; Holden, S. B. *Measure. Control*, **2001**, *34*, 235-239.
[64] Trotter, M. W. B.; Holden, S. B. *QSAR Comb. Sci.*, **2003**, *22*, 533-548.
[65] Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 667-673.
[66] Fausett, L. *Fundamentals of neural networks: architectures, algorithms, and applications*; Prentice-Hall, Inc. Upper Saddle River, NJ: USA, **1994**.
[67] Hagan, M. T.; Demuth, H. B.; Beale, M. *Neural network design*, PWS Publishing Co. Boston, MA: USA, **1997**.
[68] Hassoun, M. H. *Fundamentals of Artificial Neural Networks*; MIT Press: Cambridge, MA: **1995**.
[69] Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR Upper Saddle River, NJ: USA, **1994**.
[70] Kohonen, T. *Self-organization and associative memory*; Springer-Verlag New York, Inc. New York, NY: USA, **1989**.
[71] Chem, A. *Anal. Chem.*, **2004**, *76*, 1726-1732.
[72] Cronin, M. T. D.; Livingstone, D. J. *J. Med. Chem.*, **2005**, *48*, 661-663.
[73] Delaney, J. S. *Drug Discov. Today*, **2005**, *10*, 289-295.
[74] Dudek, A. Z.; Arodz, T.; Galvez, J. *Comb. Chem. High Throughput Screen.*, **2006**, *9*, 213-228.
[75] Gasteiger, J.; Zupan, J. *Angew. Chem. Intern. Ed.*, **1993**, *32*, 503-527.
[76] Niculescu, S. P. *J. Mol. Struct. (Theochem)*, **2003**, *622*, 71-87.
[77] Rost, B.; Sander, C. *Proteins*, **1994**, *19*, 55-72.
[78] Schneider, G.; Wrede, P. *Progr. Biophys. Mol. Biol.*, **1998**, *70*, 175-222.
[79] Tsoi, A. C.; Back, A. D. *IEEE Trans. Neural Netw.*, **1994**, *5*, 229-239.
[80] Ivanciuc, O.; Rabine, J. P.; Cabrol-Bass, D.; Panaye, A.; Doucet, J. P. *J. Chem. Inf. Comput. Sci*, **1996**, *36*, 644-653.
[81] Lim, C. W.; Fujiwara, S.; Yamashita, F.; Hashida, M. *Biol. Pharm. Bull.*, **2002**, *25*, 361-366.
[82] Ozdemir, M.; Embrechts, M. J.; Arciniegas, F.; Breneman, C. M.; Lockwood, L.; Bennett, K. P. *Soft Computing in Industrial Applications, 2001. SMCia/01. Proceedings of the 2001 IEEE Mountain Workshop on*, **2001**, 53-57.
[83] Sardari, S.; Sardari, D. *Curr. Pharm. Des.*, **2002**, *8*, 659-670.
[84] Taskinen, J.; Yliruusi, J. *Adv. Drug Deliv. Rev.*, **2003**, *55*, 1163-1183.
[85] Oja, M.; Kaski, S.; Kohonen, T. *Neural Comput. Surveys*, **2003**, *3*, 1-156.
[86] Gasteiger, J.; Teckentrup, A.; Teroth, L.; Spycher, S. *J. Phys. Org. Chem.*, **2003**, *16*, 232-245.
[87] Polanski, J. *Adv. Drug Deliv. Rev.*, **2003**, *55*, 1149-1162.
[88] Ochoa, C.; Chana, A.; Stud, M. *Curr. Med. Chem. Central Nervous System Agents*, **2001**, *1*, 247-256.
[89] von Korff, M.; Sander, T. *J. Chem. Inf. Model.*, **2006**, *46*, 536-544.
[90] Bayram, E.; Santago, P.; Harris, R.; Xiao, Y. D.; Clauset, A. J.; Schmitt, J. D. *J. Comp.-Aided Mol. Des.*, **2004**, *18*, 483-493.
[91] Schneider, G. *Curr. Med. Chem.*, **2002**, *9*, 2095-2101.
[92] Stahura, F. L.; Bajorath, J. *Comb. Chem. High Throughput Screen.*, **2004**, *7*, 259-269.
[93] Hartigan, J. A. *Clustering Algorithms*; John Wiley & Sons, Inc. New York, NY: USA, **1975**.
[94] Jain, A. K.; Dubes, R. C. *Algorithms for clustering data*; Prentice-Hall, Inc. Upper Saddle River, NJ: USA, **1988**.
[95] Johnson, S. C. *Psychometrika*, **1967**, *32*, 241-254.
[96] Jarvis, R. A.; Patrick, E. A. *Trans. Comput.*, **1973**, *100*, 1025-1034.
[97] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y.; Center, A. R.; San Jose, C. A. *IEEE Trans. Pattern Anal. Machine Intell.*, **2002**, *24*, 881-892.
[98] Binder, D. A. *Biometrika*, **1978**, *65*, 31.
[99] Zheng, W.; Tropsha, A. *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 185-194.
[100] Bocker, A.; Derksen, S.; Schmidt, E.; Teckentrup, A.; Schneider, G. *J. Chem. Inf. Model.*, **2005**, *45*, 807-815.
[101] Brown, R. D.; Martin, Y. C. *J. Chem. Inf. Comput. Sci*, **1996**, *36*, 572-584.
[102] Bayes, T. *Philos. Trans. R. Soc. London*, **1763**, *53*, 370-418.
[103] Kononenko, I. *Artificial Intelligen. Med.*, **2001**, *23*, 89-109.
[104] Cannon, E. O.; Amini, A.; Bender, A.; Sternberg, M. J. E.; Muggleton, S. H.; Glen, R. C.; Mitchell, J. B. O. *J. Comput.-Aided Mol. Des.*, **2007**, *21*, 269-280.
[105] Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. *J. Chem. Inf. Comput. Sci.*, **1985**, *25*, 64-73.
[106] Sheridan, R. P.; Nachbar, R. B.; Bush, B. L. *J. Comput.-Aided Mol. Des.*, **1994**, *8*, 323-340.
[107] Willett, P.; Wilton, D.; Hartzoulakis, B.; Tang, R.; Ford, J.; Madge, D. *J. Chem. Inf. Model.*, **2007**, *47*, 1961 -1966.
[108] Wilton, D. J.; Harrison, R. F.; Willett, P.; Delaney, J.; Lawson, K.; Mullier, G. *J. Chem. Inf. Model.*, **2006**, *46*, 471-477.
[109] Harper, G.; Bradshaw, J.; Gittins, J. C.; Green, D. V.; Leach, A. R. *J. Chem. Inf. Comput. Sci.*, **2001**, *41*, 1295-1300.
[110] Abidi, M.A.; Gonzalez, R.C. *Data Fusion in Robotics and Machine Intelligence*; Academic Press, New York, NY: USA, **1992**.
[111] Hall, D.L.; Llinas, J. *Handbook of multisensor data fusion*; Boca Raton, FL: CRC Press, **2001**.

[112] Bajorath, J. *Chemoinformatics: Concepts, Methods, and Tools for Drug Discovery*; Humana Press: Totowa, **2004**.
[113] Buxton, B. F.; Langdon, W. B.; Barrett, S. J. *Measurement+ Control*, **2001**, *34*, 229-234.
[114] Hassan, M.; Brown, R. D.; Varma-O'Brien, S.; Rogers, D. *Mol. Div.*, **2006**, *10*, 283-299.
[115] Whittle, M.; Gillet, V. J.; Willett, P.; Alex, A.; Loesel, J. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 1840-1848.
[116] Willett, P. *Drug Discov Today*, **2006**, *11*, 1046-1053.
[117] Whittle, M.; Gillet, V. J.; Willett, P.; Alex, A.; Loesel, J. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 1840-1848.
[118] Truchon, J. F.; Bayly, C. I. *J. Chem. Inf. Model.*, **2007**, *47*, 488 - 508.
[119] Bruce, C. L.; Melville, J. L.; Pickett, S. D.; Hirst, J. D. *J. Chem. Inf. Model.*, **2007**, *47*, 219-227.
[120] Plewczynski, D.; Spieser, S. A.; Koch, U. *J. Chem. Inf. Model.*, **2006**, *46*, 1098-1106.
[121] Zhang, Q.; Muegge, I. *J. Med. Chem.*, **2006**, *49*, 1536-1548.
[122] Glick, M.; Jenkins, J. L.; Nettles, J. H.; Hitchings, H.; Davies, J. W. *J. Chem. Inf. Model.*, **2006**, *46*, 193-200.
[123] Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. *Comput. Chem.*, **2001**, *26*, 5-14.
[124] Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 1882-1889.
[125] Zernov, V. V.; Balakin, K. V.; Ivaschenko, A. A.; Savchuk, N. P.; Pletnev, I. V. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 2048-2056.
[126] Muller, K. R.; Ratsch, G.; Sonnenburg, S.; Mika, S.; Grimm, M.; Heinrich, N. *J. Chem. Inf. Model.*, **2005**, *45*, 249-253.
[127] Hii, J. T.; Wyse, D. G.; Gillis, A. M.; Duff, H. J.; Solylo, M. A.; Mitchell, L. B. *Circulation*, **1992**, *86*, 1376-1382.
[128] Yap, C. W.; Cai, C. Z.; Xue, Y.; Chen, Y. Z. *Toxicol. Sci*, **2004**, *79*, 170-177.
[129] Doniger, S.; Hofmann, T.; Yeh, J. *J. Comput. Biol.*, **2002**, *9*, 849-864.
[130] Serra, J. R.; Thompson, E. D.; Jurs, P. C. *Chem. Res. Toxicol.*, **2003**, *16*, 153-163.
[131] Sorich, M. J.; Miners, J. O.; McKinnon, R. A.; Winkler, D. A.; Burden, F. R.; Smith, P. A. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 2019-2024.
[132] Zhao, C. Y.; Zhang, R. S.; Zhang, H. X.; Xue, C. X.; Liu, H. X.; Liu, M. C.; Hu, Z. D.; Fan, B. T. *SAR QSAR Environ. Res.*, **2005**, *16*, 349-367.
[133] DeLisle, R. K.; Dixon, S. L. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 862-870.
[134] Chen, B.; Harrison, R. F.; Pasupa, K.; Willett, P.; Wilton, D. J.; Wood, D. J.; Lewell, X. Q. *J. Chem. Inf. Model.*, **2006**, *46*, 478-486.
[135] Chen, B.; Harrison, R. F.; Papadatos, G.; Willett, P.; Wood, D. J.; Lewell, X. Q.; Greenidge, P.; Stiefl, N. *J. Comput.-Aided Mol. Des.*, **2007**.
[136] Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 1708-1718.
[137] Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 1177-1185.
[138] Wilton, D.; Willett, P.; Lawson, K.; Mullier, G. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 469-474.
[139] Salim, N.; Holliday, J.; Willett, P. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 435-442.
[140] Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. *J. Chem. Inf. Model.*, **2006**, *46*, 462-470.

Received: May 6, 2008 **Revised: June 27, 2008** **Accepted: August 29, 2008**