Contemporary QSAR Classifiers Compared

Craig L. Bruce,[†] James L. Melville,[†] Stephen D. Pickett,[‡] and Jonathan D. Hirst*,[†]

School of Chemistry, University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom, and GlaxoSmithKline, Gunnels Wood Road, Stevenage, Hertfordshire SG1 2NY, United Kingdom

Received August 1, 2006

We present a comparative assessment of several state-of-the-art machine learning tools for mining drug data, including support vector machines (SVMs) and the ensemble decision tree methods boosting, bagging, and random forest, using eight data sets and two sets of descriptors. We demonstrate, by rigorous multiple comparison statistical tests, that these techniques can provide consistent improvements in predictive performance over single decision trees. However, within these methods, there is no clearly best-performing algorithm. This motivates a more in-depth investigation into the properties of random forests. We identify a set of parameters for the random forest that provide optimal performance across all the studied data sets. Additionally, the tree ensemble structure of the forest may provide an interpretable model, a considerable advantage over SVMs. We test this possibility and compare it with standard decision tree models.

INTRODUCTION

The pharmaceutical industry needs to address the increasing cost and time for drug development,[1,2] and in silico lead discovery and lead optimization are becoming increasingly important means to achieve this. Lead optimization often involves quantitative structure−activity relationships (QSARs),[3,4] which focus on predicting the biological activity of a compound from a vectorial representation of molecular structure. In the past few years, the computer science community has developed new machine learning algorithms[5] suitable for QSAR development. One success story is the support vector machine (SVM),[6] which has featured regularly in the bioinformatics[7−9] and cheminformatics literature.[10−13] Some studies have suggested that SVMs show improvement over neural networks for classification and QSAR.[14−16] The advantages offered by SVMs are robust predictions even with sparse and noisy data, because the formulation of the SVM solution ensures that there is only one minimum, thus avoiding the problems of premature convergence found with neural networks.

Another popular machine learning method is the decision tree, also known as recursive partitioning.[17,18] By partitioning the data into disjoint groups, decision trees produce nonlinear models that are interpretable, a valuable property of any statistical machine learning method when applied to QSAR studies.[19,20] A further improvement in the accuracy of decision tree predictions was achieved with the introduction of ensemble methods, where multiple trees are constructed and the outputs combined to produce a final prediction.[21,22] The most popular of the ensemble techniques are boosting[23] and bagging;[24] other variants are known as decision forests[25,26] and random forests.[27] We note that ensembles are not restricted to consist only of decision trees; other algorithms used for base learners include linear discriminant analysis,[28] neural networks,[29,30] and partial least-squares regression.[31,32]

Given the increasingly widespread adoption of these advanced machine learning techniques in chemometric[33−35] and cheminformatics fields,[29,36,37] it is timely to carry out a rigorous assessment of these algorithms. Recently, Plewczynski et al. presented a comparison of some machine learning methods for virtual screening.[38] The focus of our study differs in several aspects. First, we seek to apply rigorous statistical tests to our results. It is still rare when evaluating classifiers in cheminformatics to make use of tests of statistical significance. However, there is a further caveat: in wide-scale comparisons of learning algorithms, it is not sufficient to use pair-wise statistical comparison tests, such as the paired *t*-test. Therefore, we make use of nonparametric statistical tests that are suitable for multiple comparisons. Second, apart from predictive ability, there are other requirements that make such tools valuable for mining chemical data. First, it is desirable to avoid having to manipulate multiple parameters to find the optimal performance for an algorithm. The requirement to tweak such parameters can lead to overfitting,[39] giving a false picture of predictivity, as well as being potentially time-consuming. Therefore, techniques with few parameters, or for which a widely applicable set of parameters can be obtained, are desirable. Second, an interpretable model is extremely valuable in extracting structure−activity relationships and relating the predictions of algorithms to physicochemical principles. SVMs are difficult to interpret; ensemble methods using decision trees may therefore have an advantage in this area. The translation of individual tree-based methods into classification rules is already widely documented,[40−44] but there have been only limited attempts to extend the interpretation to ensembles.[37,45] We investigate and highlight some challenges in this endeavor, compared to a single decision tree.

* Corresponding author phone: +44 115 951 3478; fax: +44 115 951 3562; e-mail: jonathan.hirst@nottingham.ac.uk.
[†] University of Nottingham.
[‡] GlaxoSmithKline.

10.1021/ci600332j CCC: $37.00
Published on Web 01/03/2007

Table 1. Summary of QSAR Data Sets

data set	compound type	no. compounds	no. descriptors	
			2.5D	fragments
ACE	angiotensin converting enzyme	114	56	1024
AchE	acetyl-cholinesterase inhibitors	111	63	774
BZR	benzodiazepine receptor	163	75	832
COX2	cyclooxygenase−2 inhibitors	322	74	660
DHFR	dihydrofolate reductase inhibitors	397	70	952
GPB	glycogen phosphorylase b	66	70	692
THER	thermolysin inhibitors	76	64	575
THR	thrombin inhibitors	88	66	527

METHODS

Learning Algorithms. For generating classifiers, we used the Java machine learning workbench Weka version 3.4.7.[5] Details of the algorithms we use in this study have been described in detail previously.[5] However, we provide a brief introduction. SVMs are considered to provide state-of-the-art classification performance, and we therefore use them as a benchmark to compare the performance of the ensemble methods, which make use of decision trees. SVMs create a separating hyperplane in the descriptor space of the training data, and molecules are classified on the basis of what side of the hyperplane they are located.[6] The advantage of the SVM is that, by use of the so-called "kernel trick", the distance between a molecule and the hyperplane can be calculated in a transformed (nonlinear) feature space, but without requiring the explicit transformation of the original descriptors. A variety of kernels have been suggested, such as the polynomial and radial basis function (RBF). A polynomial kernel with an exponent of one reduces to the case of the linear SVM. Finding the optimal separating hyperplane requires the use of quadratic programming. As this can be time-consuming, we make use of the sequential minimal optimization[46] (SMO) variant of SVMs, which provides an approximation to the quadratic programming step. SVMs come with a range of parameters: those that affect the overall SVM and those specific to the kernel. Even with the speedup associated with SMO, a full search of the entire parameter space would be prohibitively time-consuming, and it is necessary to focus on the parameters that are most crucial to the performance of the algorithm, such as the choice of kernel.[47] We tune one kernel-independent parameter, two different kernels, and one kernel-specific parameter. The two kernels we investigate are the polynomial and RBF. In addition, the complexity constant is varied from the default 1 to 0.05 and 50. For each kernel, one kernel-specific parameter is altered, the exponent for the polynomial (default 1, altered to 2 and 3) and γ, the RBF width (default 0.01, altered to 0.001 and 0.1). The complexity constant controls the tolerance of misclassified molecules: the higher the value, the greater the importance of reducing misclassifications in the training model.

Decision trees recursively partition molecules into a series of disjoint sets or nodes, starting from a pool of all training data (called the root node). The node into which a molecule is placed is dependent on a threshold value of a particular descriptor (a branching rule). When a node is reached for which no branching rule is defined (a terminal node, or "leaf"), the molecule is classified, on the basis of the properties of the molecules with which it shares the node. This is normally achieved via majority vote. Each path through the tree from the root to a leaf can be extracted and represented as a classification rule, enabling interpretations to be made, which accounts for the popularity of this technique.[40−44] The tree-building algorithm used in this study is J48, a Java implementation of the C4.5 algorithm due to Quinlan.[18] A feature of the J48 algorithm is that it "prunes" leaves that do not contribute greatly to the predictive accuracy of the tree. This creates smaller trees, which may be more resistant to overfitting.

While decision trees have the advantage of being interpretable, their predictive abilities are normally inferior to that of more advanced techniques. Ensemble techniques attempt to compensate for the weakness of an individual tree by combining the predictions of multiple trees. The key to ensemble methods is therefore producing diverse selections of trees. This is normally achieved by training each tree on a different subset of the data. This can involve subsampling both molecules and descriptors. A commonly used technique is the bootstrap,[48] which samples the molecules with replacement. The resulting bootstrap sample is likely to contain duplicate molecules, while some of the original training data may not appear at all. The simplest ensemble method we study is bagging.[24] Here, the ensemble is formed by simply repeatedly training trees on bootstrap samples of the original data. Classification of new molecules is by majority voting across all trees. Boosting[49] also uses bootstrap samples. However, the accuracy of the previous tree is used to bias the selection of molecules for the next sample, with poorly predicted molecules being given a greater chance of selection (or a larger weight) so that the next tree will focus on these more difficult cases. Again, voting is used to classify new molecules, but the vote is weighted to give more influence to trees with greater accuracy. Boosting is widely considered to be more accurate than bagging.[50] In our experiments we use the AdaBoostM1 algorithm.[23] A random forest[27] is similar to bagging, except that, as well as sampling training molecules randomly, only small subsets of descriptors are used to build each tree. Like bagging, classification is by majority vote. For bagging and boosting, we use the J48 algorithm to build the base decision tree classifiers. For random forest, we use random trees as a base classifier rather than trees built with J48. The main difference between J48 trees and random trees is that no pruning is carried out for random trees.

Data Sets. Eight data sets (Table 1) have been taken from the study of Sutherland et el.:[51] (1) a set of angiotensin-converting enzyme (ACE) inhibitors originally used for comparative molecular field analysis (CoMFA) modeling;[52] (2) a set of acetyl-cholinesterase (AchE) inhibitors, a subset of which was used in CoMFA studies;[53] (3) a set of ligands

for the benzodiazepine receptor (BZR) also used for validating several QSAR methods;[54] (4) a set of cyclooxygenase-2 (COX2) inhibitors—a subset was used in CoMFA studies;[55] (5) a set of dihydrofolate reductase (DHFR) inhibitors, which were also used for comparative molecular similarity indices and analysis modeling;[56] (6) a set of glycogen phosphorylase b (GPB) inhibitors;[57] (7) a set of thermolysin (THER) inhibitors;[58] and (8) a set of thrombin (THR) inhibitors.[59]

Our study focuses on classification of activity, either active or inactive. The original data sets provide continuous numerical values for activity (pIC_{50} for the first five data sets and pK_i for the last three); each data set shows a uniform distribution of activity values. Therefore, the median activity value was used as a threshold between active or inactive compounds to create a 50/50 split of active/inactive observations. Balancing the data set in this way simplifies the validation of the classifiers and allows the use of percentage classification accuracy as a measure of classifier performance.

To create QSARs, descriptors of the molecules are required. We use the 2.5D descriptors generated by Sutherland et al.[51] using Cerius^2.[60] In addition, linear fragment descriptors for these data sets were computed. For each data set, descriptors containing data on the atomic number, bond types, connectivity, chirality, and number of hydrogen atoms associated with each atom were generated for all nonbranched molecular fragments of four to seven atoms in size. The number of occurrences of each fragment in each molecule is recorded, producing an integer string as a descriptor of each molecule. The fragment data sets are of much larger dimensionality than the 2.5D descriptors. The original eight data sets are henceforth referred to as 2.5D data sets, while the new data sets are referred to as the linear fragment data sets.

To validate the performance of each classifier, we have used the percentage of correctly classified molecules from 10-fold cross-validation as the measure for the model, averaged over 10 entire repeats of the cross-validation, using different random seeds. In cross-validation, the data set is split into *n* folds; one fold is used for testing, the rest for training. This is repeated *n* times, so all the data have been used as test data once. The *n* errors are averaged, giving the error rate for the data set. Ten is a commonly used value for *n*.[61] We stress that the reported cross-validated results are for genuine predictions and distinct from the internal validation used by some resampling techniques, which make use of the fact that approximately one-third of the original training data is unused after bootstrap sampling. Predictions on this unused data are known as out-of-bag (OOB) data. Subsequent to completing the work reported here, it came to our attention that the OOB accuracy reported by Weka version 3.4.7 is overly pessimistic. This has been corrected in version 3.4.8. This alteration does not affect the results reported, as the OOB error is not used to measure the accuracy of any of the algorithms in our study, and rerunning the affected algorithms resulted in identical cross-validated performance. In addition to the classification accuracy, we assessed the robustness of the models using the standard deviation of the accuracy across the 10 repeats of the entire 10-fold cross-validation results, averaged for all data sets.

Table 2. Percentage of Correctly Classified Molecules for Different Classifiers on 2.5D Descriptor Data Sets[a]

data set	tree	bagged tree	boosted tree	random forest	SVM	tuned forest[b]	tuned SVM[c]
ACE	86.9	86.5	86.6	85.4	**90.3**	89.3	89.9
AchE	70.6	71.6	72.7	72.6	72.0	**79.5**	74.3
BZR	71.7	75.5	75.4	74.0	77.4	79.5	**81.6**
COX2	75.6	75.7	**76.1**	73.4	75.4	75.7	75.2
DHFR	78.8	83.2	83.4	83.1	79.6	**84.9**	82.2
GPB	70.6	74.5	76.2	74.1	73.9	**76.7**	75.3
THER	67.2	69.2	67.8	69.7	69.5	**74.6**	**74.6**
THR	66.5	69.1	68.0	69.1	67.2	**72.5**	69.0

[a] Values in bold denote the highest accuracy for that data set. [b] 100 trees. [c] Polynomial kernel, exponent = 2, complexity constant = 0.05.

RESULTS AND DISCUSSION

Classifier Accuracy. We first consider the results of the 2.5D data set. The percentage of correctly classified molecules is given in the first five columns of Table 2. In general, all classifiers perform reasonably, with classifications between 67 and 90%. The strongest trends are clearly data-set-dependent, with classification being most successful on the ACE and DHFR data sets, and least successful on the THER and THR data sets. As anticipated, the decision tree classifier is the least effective of all the classifiers studied, being the least accurate of the five methods studied in six of the eight data sets. However, it still performs creditably, as the difference between the best and worst and classifier differs by no more than 6% on any data set. Additionally, it should be borne in mind that the average standard deviation of the cross-validation accuracies was 2.5%, with the smaller data sets (GPB, THER, and THR) showing larger standard deviations in the results than the larger data sets. Contrary to our expectations, SVM was the best classifier on only two data sets. However, the Weka default is to use a linear SVM; we assess the performance of a nonlinear SVM below.

The ensemble classifiers (bagging, boosting, and random forest) improved classification accuracy over the decision tree for all of the data sets except ACE. Within the ensemble techniques, there was little difference in performance. Boosting had a slight edge over bagging and random forest, outperforming both on five data sets. Having established the default behavior of the algorithms as implemented in Weka, an obvious avenue of exploration for improving the performance of the ensemble classifiers is to increase the number of trees in the ensembles. Additionally, random forests have an extra parameter, the number of descriptors available to the tree-building algorithm when creating branching rules, which we investigated separately. First, we increased the number of trees in the aggregates from 10 to 200 in steps of 10. Similar results were observed for bagging, boosting, and random forest. We concentrate on random forest in the following discussion, because it improved the most upon adding more trees, but the comments below apply equally to boosting and bagging. Accuracy increased upon adding more trees, reaching a maximum for all data sets between 30 and 50 trees. Little improvement in accuracy was observed beyond this point. The robustness of the accuracies was also improved with an increasing number of trees, although the improvement is not monotonic, as can be observed in Figure 1.

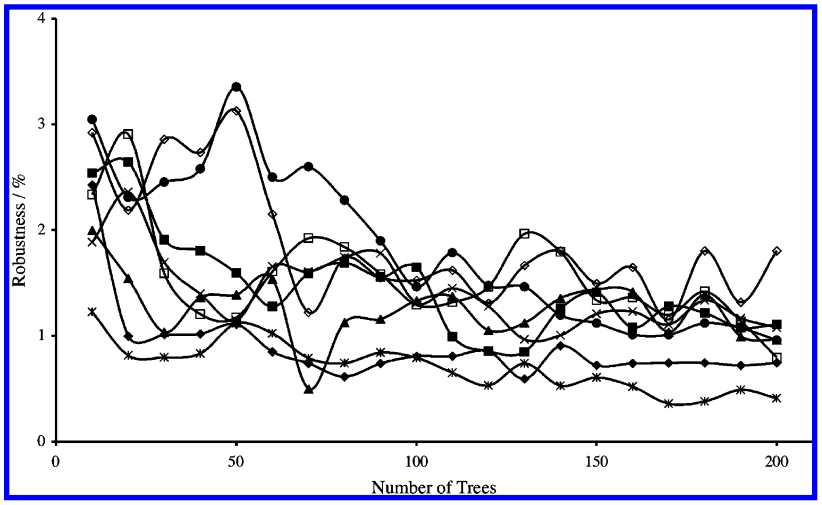

Figure 1. Robustness with increasing number of trees (on 2.5D descriptors). Legend: ACE (♦); AchE (■); BZR (▲); COX2 (×); DHFR (*); GPB (●); THER (□); THR (◊).

Nonetheless, at 100 trees, the robustness of all trees is increased over that observed for 10 trees, with all standard deviations below 2%. No obvious improvement is observed upon increasing the number of trees to 200. Hence, we consider the 100-tree forest as "converged". The accuracies for random forest classifiers built with 100 trees are shown in Table 2, in the random forest column marked "tuned". The improvement in accuracy over the 10-tree forest ranges from 2 to 7%. To establish whether the difference in performance was statistically significant, a two-tailed paired *t*-test[62] was carried out on 10 versus 100 trees, using 100 repeats (10-fold cross-validation repeated 100 times with different random seeds). For all data sets, the improvement in performance is statistically significant at $p = 0.001$ for 100 trees. Second, we looked at the number of features available during tree construction, a parameter available only for random forests. By default, only $\log_2 M + 1$ descriptors, selected randomly, are available for selection to construct each branching rule in a tree, where M is the total number of descriptors in the data set. For the 2.5D descriptors, this corresponds to six or seven descriptors per branch. We therefore looked at increasing the descriptor choice to 10, 20, 30, and 40 descriptors. To ensure our results were not dependent on the choice of random seed, we repeated the procedure 10 times, for two different forest sizes with 10 and 30 trees. The results showed that increasing the descriptor choice did not provide a consistent improvement, and in most cases, the default value was optimal. Therefore, we retained the default setting.

While increasing the number of trees in ensemble algorithms provides a clear means for optimizing performance, a principled approach to improving the SVM is more difficult to achieve, simply because it has a much larger number of parameters to modify. Our initial experiments suggested that two parameters had the greatest effect on SVM: the complexity constant and the type of kernel (polynomial or RBF). Associated with both kernels was a single parameter: the exponent for the polynomial kernel and the γ value (RBF width) for the RBF kernel. We therefore chose an optimal set of these parameters based on the mean accuracy across all eight data sets. Further detail on the results obtained for the various settings during the optimization process is provided in the Supporting Information. Only the accuracies for the best SVM results are given in Table 2, under the SVM column "tuned". The results show that a nonlinear SVM can improve performance for six out of eight data sets over the default linear SVM implementation in Weka. However, the performance of the 100-tree random forest is still superior to the tuned SVM for five of the eight data sets.

Table 3. Percentage of Correctly Classified Molecules of the Different Classifiers on Linear Fragment Descriptor Data Sets[a]

data set	tree	bagged tree	boosted tree	random forest	SVM	tuned forest[b]	tuned SVM[c]
ACE	80.4	82.0	81.0	80.5	78.9	80.0	**82.2**
AchE	64.1	68.0	68.8	70.5	69.4	70.5	**77.1**
BZR	74.0	75.0	69.8	67.3	74.0	68.7	**75.8**
COX2	71.1	71.5	71.0	68.1	**72.6**	68.7	71.1
DHFR	84.4	85.4	83.1	84.9	83.5	85.5	**86.5**
GPB	73.8	75.6	76.2	74.5	**77.4**	75.2	76.7
THER	72.2	75.8	75.5	75.4	75.3	**76.7**	73.4
THR	**71.5**	69.2	68.8	66.7	71.1	68.4	69.8

[a] Values in bold denote the highest accuracy for that data set. [b] 100 trees. [c] RBF kernel, $\gamma = 0.1$, complexity constant = 1.

Having established some useful parameters and observed a pattern of behavior across the eight data sets with the 2.5D descriptors, we investigated whether similar results could be obtained using the fragment descriptors. Classification accuracies are given in Table 3. Results are comparable to those of the 2.5D descriptors. The difference between the most and least accurate classifier was slightly increased, ranging between 2 and 8%, depending on the data set. Again, the ACE and DHFR data sets are the easiest to classify. However, a clear difference emerges when comparing performances based on the number of molecules in the data set. For the "small" data sets (<100 molecules), accuracies improve upon moving to a higher dimensionality of descriptors. Conversely, for the "large" data sets (>100 molecules), the majority of classifiers record a decrease in accuracy. As the larger data sets are described by a larger pool of descriptors, this may suggest that the increase in the dimensionality of the descriptor space outstrips the increase in information provided by the larger data sets; that is, the "curse of dimensionality"[63] is occurring. The decision tree classifier was not invariably the worse classifier tested, and for the THR data set, it was the best. However, for all other

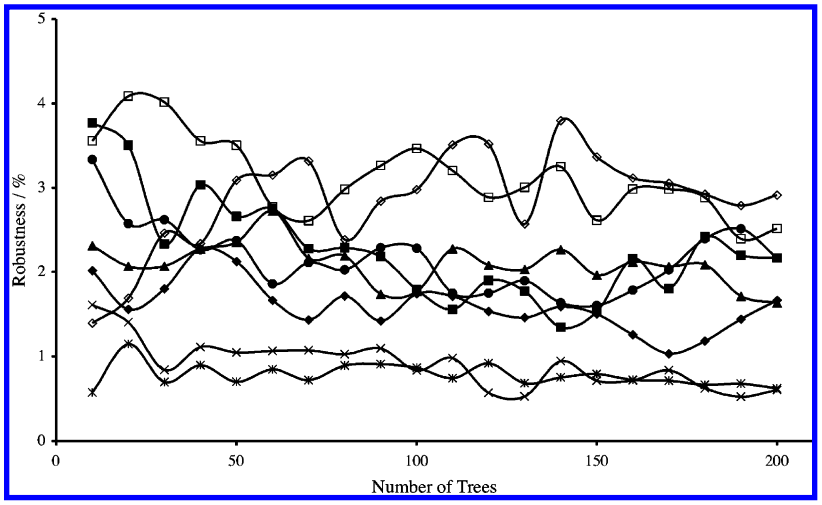

Figure 2. Robustness with increasing number of trees (on linear fragment descriptors). Legend: ACE (♦); AchE (■); BZR (▲); COX2 (×); DHFR (*); GPB (●); THER (□); THR (◊).

data sets, at least one of the ensemble algorithms was able to improve upon the decision tree performance. Bagging was the best at this, while boosting and random forest could improve over the standard decision tree in four and five data sets, respectively. As with the 2.5D descriptors, the SVM was the best classifier on two data sets. Increasing the number of trees to 100 in the random forest improved the accuracy for six out of eight data sets. However, there was less improvement in robustness in going to 100 trees, compared to the 2.5D descriptors, as shown in Figure 2.

A two-tailed *t*-test showed no significant difference for $p = 0.001$ for the AchE, COX2, and GPB data sets. It is conceivable that the larger dimensionality of the fragment data set requires a larger number of trees in the ensembles; however, increasing the number of trees to 1000 did not result in any major increase in accuracy or robustness. Again, there was no clear improvement in the performance of the random forest when the number of descriptors available to the tree-building algorithm was increased. For the fragment descriptors, a RBF SVM was found to be optimal and gave the best performance of any algorithm on the fragment descriptors for four data sets. However, boosting, bagging, and random forests gave comparable accuracies on all of the data sets, except AchE and COX2, where one or more of the algorithms struggled.

In order to put these observations on a more quantitative footing, we carried out a multiple-comparison statistical analysis, using the Friedman statistic, with a correction by Iman and Davenport, to detect the existence of a statistically significant difference between the classifiers. This test does not, however, identify *which* classifiers are different, only that a difference exists. To identify significantly different classifiers, we perform a post-hoc test using the Nemenyi test. Details of the procedure to carry out these tests are given in the appendix. For these tests, we pooled the results for both sets of descriptors, giving $N = 16$ data sets. Both the default and tuned versions of the SVM and random forest classifiers were considered separately, making $k = 7$ classifiers. Carrying out the Friedman test with the Iman and Davenport correction indicated that statistically significant differences between the classifiers existed at the $p = 0.01$ significance level. We therefore carried out the Nemenyi post-hoc analysis to determine which classifiers were significantly different from each other. It is a known weakness of the Nemenyi test that it has a smaller power than the Friedman test. However, it was still possible to detect statistically significant differences between classifiers at $p = 0.05$. Thus, we can deduce that the performance of the tuned SVM (with the best average rank over all 16 datasets), and tuned random forest are significantly different from the performance of a single decision tree (with the lowest average rank over all datasets). However, it is not possible to detect whether SVM is statistically different from the ensemble algorithms, or if, in turn, the ensemble algorithms are significantly different from the use of a single decision tree.

Interpretation of Tree Models. Beyond assessing the predictive capabilities of the ensemble methods, we also sought to characterize the interpretability of the resulting models. Decision trees are widely used for their interpretability; therefore, an examination of the ensemble might provide similar insights. There are two main issues to account for in an interpretation of an ensemble: the size and shape of each individual tree and dealing with the large number of trees generated. Figure 3 shows two trees, one generated by the J48 algorithm (a) and one that is part of a random forest ensemble (b). For clarity, we have not displayed the value of the descriptor threshold applied at each branch. These trees were used to predict activity for the ACE data set using the 2.5D descriptors but have been chosen to represent the typical structure observed across all eight data sets. It is apparent from Figure 3 that decision trees generated by J48 are less "bushy" and less balanced than those used in random forest. This is a consequence of the pruning that takes place in the J48 algorithm, which is not applied to the trees grown for use in random forest. Terminal leaf populations are therefore on average smaller in trees in the random forest ensemble, and this makes interpretation less reliable for these nodes. Note also that the random tree contains the IC descriptor (a descriptor related to information theoretic concepts of entropy) twice. An analysis of the descriptors used in the ensembles and the J48 trees shows a reasonable degree of overlap for most data sets. One exception to this was with the ACE data set, where J48 decision trees often contain a descriptor indicating the presence of a nitrogen atom type; these were rarely chosen in the ensembles.

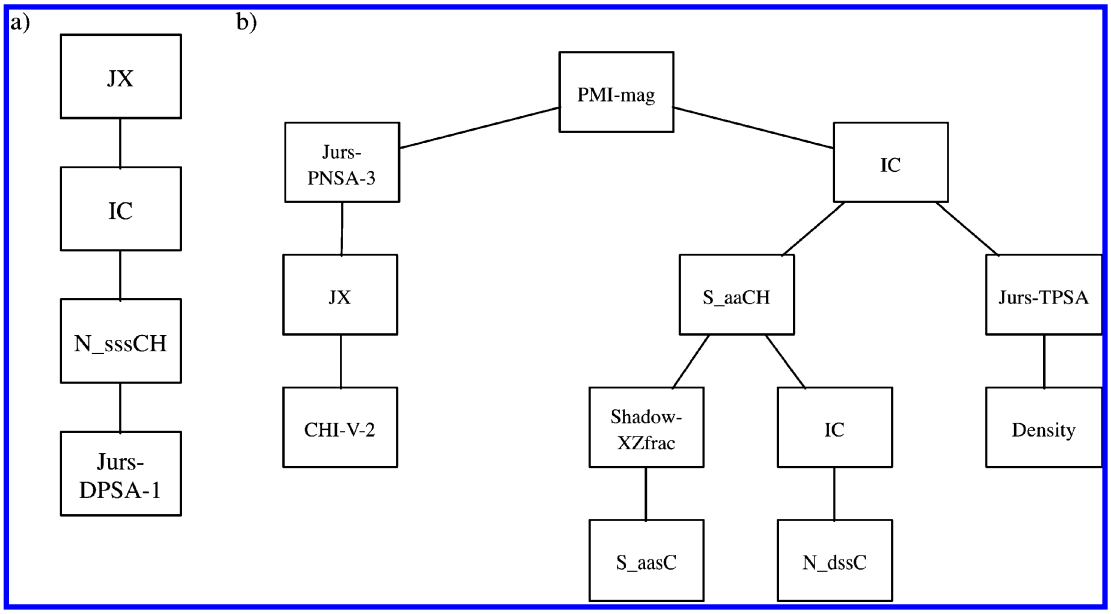

Figure 3. Decision trees classifying activity of the ACE data set generated by (a) the J48 algorithm with pruning and (b) the random tree algorithm without pruning.

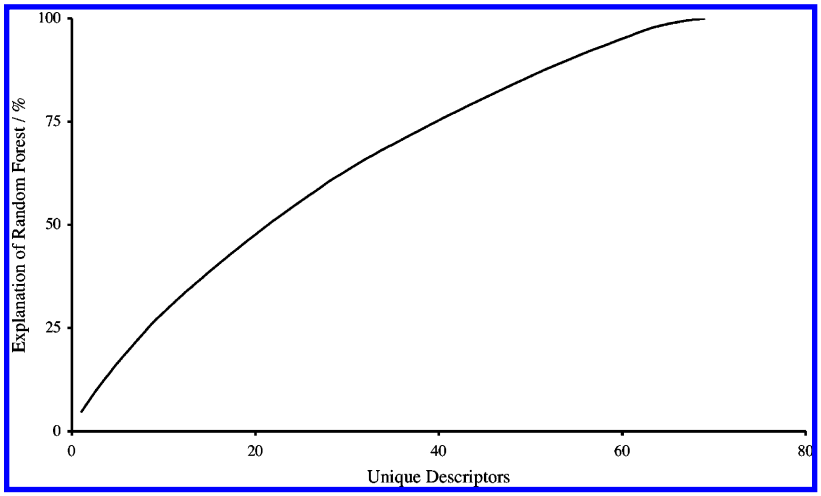

Figure 4. Percentage of model explained by unique descriptors for the DHFR data set.

We next consider the descriptor analysis of ensembles as a whole in more detail. We focus here on the frequency with which descriptors are chosen to appear in the trees. One obvious aid to interpretability would be if a subset of descriptors was chosen with a frequency much higher than that of others. To see whether this is the case, we recorded which descriptor was selected for each branching rule in a 100-tree random forest, for all eight data sets. We then plotted the number of descriptors as a percentage of the total number used in the forest against the number of unique descriptors selected. An example for the DHFR data set is shown in Figure 4.

Results for all the other data sets showed the same shape. The diagram can be interpreted similarly to a receiver operating characteristic curve. If only a few descriptors were selected by the tree-building algorithm, we would expect to see an initially steep vertical line at $x = 0$, indicating that most trees consisted of a few descriptors. This pattern is not observed to a large extent, although it can be seen that 50% of the branching points in the DHFR ensemble consist of 20 descriptors, which represents only 25% of the total number of descriptors chosen. Therefore, it seems that no "super descriptors" emerge from the random forest ensemble. This is perhaps to be expected, given that only approximately seven descriptors, chosen randomly from all descriptors, are available for selection at any given branch point during tree construction. A related issue is whether the same descriptors are chosen most frequently in ensembles of different sizes. We considered the top 10 most frequently chosen descriptors for this purpose and examined the effect of increasing the number of trees in the forest from 100 to 1000 in steps of 100 trees. Larger data sets show greater consistency: the same top 10 descriptors are present in all forests for the DHFR, the largest data set.

As the data sets grow smaller, there is less overlap between the ensembles, with only seven descriptors consistently found in all forests for ACE, AchE, and BZR. For the small data sets (GPB, THER, and THR), at most three descriptors are in common across all forests. Despite this variability for some data sets, some commonly occurring descriptors can be identified. Atom type descriptors are commonly found, which describe hybridization and bonding information associated with an atomic center; these are conceptually similar to the fragment descriptors. Additionally, some frequently occurring

whole-molecule descriptors could be identified for some data sets. For the ACE data set, we identified IC and JX as falling into this category; they correspond to information entropy and the Balaban index,[64] respectively. The latter topological descriptor accounts for both cycles and conjugation in a structure, and active compounds in this data set are well-characterized by the presence of aromatic rings. Conversely, for the COX2 data set, electronic and partial surface area descriptors (Apol and Jurs-RPCG) are frequently selected. This would suggest that the overall electronegativity of the compounds is important for activity. For the DHFR data set, topological and partial surface area descriptors appeared in all forests (Kappa-3, Jurs-FPSA-3, and Jurs-FNSA-3). This shape and surface information can help distinguish between inactive compounds which may have longer, positively charged chains, for example, nitrogen cations, which are less favorable than short, negatively charged chains.

CONCLUSIONS

Over the eight data sets, the support vector machine was the best of the classifiers studied, whether using the 2.5D descriptors or the fragments. We confirmed this superiority over a single decision tree through a multiple comparison statistical test. However, statistical significance should not be confused with practical significance, and the performance of boosting, bagging, and random forest was in most cases comparable with that of SVM, and in some cases superior. Furthermore, achieving the optimal performance for SVM can be a formidable task, because of the large number of parameters associated with it. We did not succeed in finding a set of parameters that was universally applicable. For example, a quadratic polynomial kernel was optimal for the 2.5D descriptors, while a RBF kernel was best for the fragment descriptors. Conversely, boosting, bagging, and random forest have fewer parameters, while performing competitively with the best SVM results. All of the ensemble algorithms benefited from the use of an increasing number of trees—these had a generally positive effect on both classification accuracy and robustness. We were able to identify 100 trees as being an optimal setting for most cases, and we recommend that users of the Weka machine-learning workbench use this value for cheminformatics applications, rather than the smaller number available as a default. In general, all the algorithms we studied were marginally less effective for the more numerous linear fragment descriptors. While the dimensionality of these descriptors was large, even by current standards (~1000 descriptors per data set), data sets are likely to become larger in the future, rather than smaller. Although ensemble algorithms like random forest were designed with high dimensional data sets in mind, cheminformatics data sets tend to be at least an order of magnitude larger, in terms of the ratio of descriptors to observations, than those commonly found in machine learning. An additional advantage of decision-tree-based methodologies is their potential to provide interpretable models. We have found that there are challenges associated with extending this interpretation to ensembles. In particular, trees in the ensembles are "bushier" than standard decision trees. A descriptor-level frequency analysis can provide some insight, but for the data sets studied here, there was a wide distribution of descriptors in the ensemble, so attempting to select a small subset of descriptors from the ensemble is somewhat subjective. Therefore, choosing a decision-tree-based learning algorithm over the SVM may be less advantageous than hoped. Despite the success of random forest and SVM on our data sets, a single decision tree was surprisingly competitive, and its interpretability is not in doubt. Clearly, therefore, along with more sophisticated ensemble-building algorithms, there is substantial scope for concomitantly advanced procedures for extracting information from decision tree ensembles, and this will be the focus of future work from this laboratory.

ACKNOWLEDGMENT

We thank GlaxoSmithKline and EPSRC (GR/575765/01) for financial support and the EPSRC for an equipment grant (GR/62052/01) for computers. We thank Gavin Harper and Chris Luscombe from GlaxoSmithKline for helpful discussions. We are grateful for the use of the High Performance Computing facility at the University of Nottingham.

APPENDIX

Nonparametric Multiple-Comparison Statistical Tests. Traditional pairwise statistics (such as the *t*-test) are inappropriate when making multiple comparisons. Here, we present a two-stage, nonparametric approach. The first test that should be applied is to determine if any significant differences between the classifiers can be detected. For this, the Friedman test is used.[65,66] The null hypothesis for the Friedman test is that there is no difference between any of the classifiers. If the null hypothesis is rejected, it does not determine which groups are different from each other; for this, a separate test is required. However, the Friedman test should be applied first, to determine if further analysis is justified. In the following, we shall assume that there are k classifiers and N data sets, and that all classifiers have been applied to all data sets, in each case yielding a real-valued measure of the performance of the classifier. In the work discussed in this study, we have used percentage accuracy, but other measures, such as the area under the receiver operating characteristic curve, are also possible.[67]

The Friedman test proceeds as follows. For each data set, the performances of the classifiers are ranked, in ascending order; that is, the best performing classifier has rank k, the next best rank $k - 1$, and so on. Then, for each classifier j, the mean average rank across all data sets, $\bar{R}_j$, is calculated. The individual rank of a given classifier, j, and data set, i, is denoted by r_{ij}.

$$\bar{R}_j = \frac{\sum_i r_{ij}}{k} \tag{1}$$

The Friedman statistic, χ_F^2, is then calculated as

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j \bar{R}_j^2 - \frac{k(k+1)^2}{4}\right] \tag{2}$$

For a sufficiently large k and N (Demšar[68] suggests $k > 5$ and $N > 10$), the Friedman statistic is distributed according to the χ-squared statistic, χ^2, with $k - 1$ degrees of freedom.

Critical values of χ_F^2 for smaller values of k and N are provided in ref 62.

In the event of tied ranks (i.e., two or more classifiers giving identical performances on a data set), then a correction factor may be applied to χ_F^2. One such correction (used, for example, in the statistical package R[69]) where the sum of the ranks across all data sets, R_j, is

$$R_j = \sum_i r_{ij} \tag{3}$$

and the correction statistic is

$$\chi_C^2 = \frac{12\sum_j \left\{ \left[R_j - \frac{n(k+1)}{2} \right]^2 \right\}}{nk(k+1) - C} \tag{4}$$

where C, the correction factor, is

$$C = \frac{\sum_{i=1}^{m} (t_i^3 - t_i)}{m} \tag{5}$$

m is the number of groups of tied ranks for a classifier, and t_i is the number of ties in the ith tied group. Equation 4 without the correction factor, C, gives the same result as eq 2.

Iman and Davenport[70] demonstrated that the Friedman statistic was too conservative and suggested the following improvement:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \tag{6}$$

This is distributed according to the F distribution with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom.

If the null hypothesis is rejected, then statistically significant differences between the classifiers are present. The Nemenyi test[71] can then be applied to determine these differences in a pairwise fashion. To compare classifiers a and b, the difference in mean ranks, $\bar{R}_a - \bar{R}_b$, is calculated. This value is then compared to the critical difference, CD:

$$\mathrm{CD} = q'_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{7}$$

where the corrected q statistic, q'_α, is given by

$$q'_\alpha = \frac{q_\alpha}{\sqrt{2}} \tag{8}$$

q_α is the critical value of the "Studentized range" statistic at a given level of significance, α. Tables of the critical values of the q distribution can be found in statistical packages such as R and elsewhere.[62] If the difference between the average ranks is larger than CD, then the difference between the classifiers is significantly different at the specified value of α. Demšar has discussed multiple classifier comparisons in greater detail.[68]

Supporting Information Available: Linear fragment descriptors for all data sets are available in Weka format (ARFF). Full parameter tuning results of SVMs for both data sets is also provided. This information is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Clark, D. E.; Pickett, S. D. Computational Methods for the Prediction of 'Drug-Likeness'. *Drug Discovery Today* **2000**, *5*, 49−58.

(2) Claus, B. L.; Underwood, D. J. Discovery Informatics: Its Evolving Role in Drug Discovery. *Drug Discovery Today* **2002**, *7*, 957−966.

(3) Hansch, C.; Maloney, P. P.; Fujita, T. Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. *Nature* **1962**, *194*, 178−180.

(4) Hansch, C.; Fujita, T. $\rho-\sigma-\pi$ Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *J. Am. Chem. Soc.* **1964**, *86*, 1616−1626.

(5) Witten, I. H.; Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed.; Morgan Kaufmann: San Francisco, CA, 2005.

(6) Burges, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowledge Discovery* **1998**, *2*, 121−167.

(7) Zhang, Q.; Yoon, S.; Welsh, W. J. Improved Method for Predicting β-Turn Using Support Vector Machine. *Bioinformatics* **2005**, *21*, 2370−2374.

(8) Furey, T. S.; Cristianini, N.; Duffy, N.; Bednarski, D. W.; Schummer, M.; Haussler, D. Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. *Bioinformatics* **2000**, *16*, 906−914.

(9) Zhang, S.-W.; Pan, Q.; Zhang, H.-C.; Zhang, Y.-L.; Wang, H.-Y. Classification of Protein Quaternary Structure with Support Vector Machine. *Bioinformatics* **2003**, *19*, 2390−2396.

(10) Czerminski, R.; Yasri, A.; Hartsough, D. Use of Support Vector Machine in Pattern Classification: Application to QSAR Studies. *Quant. Struct.-Act. Relat.* **2001**, *20*, 227−240.

(11) Lepp, Z.; Kinoshita, T.; Chuman, H. Screening for New Antidepressant Leads of Multiple Activities by Support Vector Machines. *J. Chem. Inf. Model.* **2006**, *46*, 158−167.

(12) Luan, F.; Ma, W. P.; Zhang, X. Y.; Zhang, H. X.; Liu, M. C.; Hu, Z. D.; Fan, B. T. QSAR Study of Polychlorinated Dibenzodioxins, Dibenzofurans, and Biphenyls Using the Heuristic Method and Support Vector Machine. *QSAR Comb. Sci.* **2006**, *25*, 46−55.

(13) Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active Learning with Support Vector Machines in the Drug Discovery Process. *J. Chem. Inf. Model.* **2003**, *43*, 667−673.

(14) Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. *J. Chem. Inf. Model.* **2003**, *43*, 1882−1889.

(15) Tetko, I. V.; Solov'ev, V. P.; Antonov, A. V.; Yao, X.; Doucet, J. P.; Fan, B.; Hoonakker, F.; Fourches, D.; Jost, P.; Lachiche, N.; Varnek, A. Benchmarking of Linear and Nonlinear Approaches for Quantitative Structure−Property Relationship Studies of Metal Complexation with Ionophores. *J. Chem. Inf. Model.* **2006**, *46*, 808−819.

(16) Zernov, V. V.; Balakin, K. V.; Ivaschenko, A. A.; Savchuk, N. P.; Pletnev, I. V. Drug Discovery Using Support Vector Machines. The Case Studies of Drug-Likeness, Agrochemical-Likeness, and Enzyme Inhibition Predictions. *J. Chem. Inf. Model.* **2003**, *43*, 2048−2056.

(17) Breiman, L.; Friedman, J.; Stone, C. J.; Olshen, R. A. *Classification and Regression Trees*; Chapman & Hall: Boca Raton, FL, 1984; p 368.

(18) Quinlan, R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, 1993; p 302.

(19) Rusinko, A.; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning. *J. Chem. Inf. Model.* **1999**, *39*, 1017−1026.

(20) Nilakantan, R.; Nunn, D. S.; Greenblatt, L.; Walker, G.; Haraki, K.; Mobilio, D. A Family of Ring System-Based Structural Fragments for Use in Structure−Activity Studies: Database Mining and Recursive Partitioning. *J. Chem. Inf. Model.* **2006**, *46*, 1069−1077.

(21) Sollich, P.; Krogh, A. Learning with Ensembles: How Overfitting Can Be Useful. In *Advances in Neural Information Processing Systems 8*, Denver, Colorado, 1996; Touretzky, D. S., Mozer, M. C., Hasselmo, M. E., Eds.; MIT Press: Denver, Colorado, 1996; pp 190−196.

(22) Bauer, E.; Kohavi, R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Mach. Learn.* **1999**, *36*, 105−139.

(23) Freund, Y.; Schapire, R. R. Experiments with a New Boosting Algorithm. In *Thirteenth International Conference on Machine Learning*, Bari, Italy, 1996; Morgan Kaufmann: Bari, Italy, 1996; pp 148−156.

(24) Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123−140.
(25) Ho, T. K. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832−844.
(26) Tong, W.; Hong, H.; Fang, H.; Xie, Q.; Perkins, R. Decision Forest: Combining the Predictions of Multiple Independent Decision Tree Models. *J. Chem. Inf. Model.* **2003**, *43*, 525−531.
(27) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(28) Arodz, T.; Yuen, D. A.; Dudek, A. Z. Ensemble of Linear Models for Predicting Drug Properties. *J. Chem. Inf. Model.* **2006**, *46*, 416−423.
(29) Agrafiotis, D. K.; Cedeno, W.; Lobanov, V. S. On the Use of Neural Network Ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 903−911.
(30) Fernandez, M.; Tudidor-Camba, A.; Caballero, J. Modeling of Cyclin-Dependent Kinase Inhibition by 1H-Pyrazolo[3,4-d]pyrimidine Derivatives Using Artificial Neural Network Ensembles. *J. Chem. Inf. Model.* **2005**, *45*, 1884−1895.
(31) Mevik, B. H.; Segtnan, V. H.; Naes, T. Ensemble Methods and Partial Least Squares Regression. *J. Chemom.* **2004**, *18*, 498−507.
(32) Zhang, M. H.; Xu, Q. S.; Massart, D. L. Boosting Partial Least Squares. *Anal. Chem.* **2005**, *77*, 1423−1431.
(33) He, P.; Fang, K. T.; Liang, Y. Z.; Li, B. Y. A Generalized Boosting Algorithm and Its Application to Two-Class Chemical Classification Problem. *Anal. Chim. Acta* **2005**, *543*, 181−191.
(34) Zhang, M. H.; Xu, Q. S.; Daeyaert, F.; Lewi, P. J.; Massart, D. L. Application of Boosting to Classification Problems in Chemometrics. *Anal. Chim. Acta* **2005**, *544*, 167−176.
(35) He, P.; Xu, C. J.; Liang, Y. Z.; Fang, K. T. Improving the Classification Accuracy in Chemistry via Boosting Technique. *Chemom. Intell. Lab. Syst.* **2004**, *70*, 39−46.
(36) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Model.* **2003**, *43*, 1947−1958.
(37) Svetnik, V.; Wang, T.; Tong, C.; Liaw, A.; Sheridan, R. P.; Song, Q. Boosting: An Ensemble Learning Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Model.* **2005**, *45*, 786−799.
(38) Plewczynski, D.; Spieser, S. A. H.; Koch, U. Assessing Different Classification Methods for Virtual Screening. *J. Chem. Inf. Model.* **2006**, *46*, 1098−1106.
(39) Hawkins, D. M. The Problem of Overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1−12.
(40) Zhang, H.; Yu, C.-Y.; Singer, B. Cell and Tumor Classification Using Gene Expression Data: Construction of Forests. *Proc. Natl. Acad. Sci. U.S.A.* **2003**, *100*, 4168−4172.
(41) Andres, C.; Hutter, M. C. CNS Permeability of Drugs Predicted by a Decision Tree. *QSAR Comb. Sci.* **2006**, *25*, 305−309.
(42) DeLisle, R. K.; Dixon, S. L. Induction of Decision Trees via Evolutionary Programming. *J. Chem. Inf. Model.* **2004**, *44*, 862−870.
(43) Deconinck, E.; Zhang, M. H.; Coomans, D.; VanderHeyden, Y. Classification Tree Models for the Prediction of Blood−Brain Barrier Passage of Drugs. *J. Chem. Inf. Model.* **2006**, *46*, 1410−1419.
(44) Leach, A. R.; Gillet, V. J. *An Introduction to Chemoinformatics*, 1st ed.; Springer: Dordrecht, Netherlands, 2003; p 259.
(45) Li, S. Q.; Fedorowicz, A.; Singh, H.; Soderholm, S. C. Application of the Random Forest Method in Studies of Local Lymph Node Assay Based Skin Sensitization Data. *J. Chem. Inf. Model.* **2005**, *45*, 952−964.
(46) Platt, J. C. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. In *Advances in Neural Information Processing Systems 11, 1999*; Kearns, M. S., Solla, S. A., Cohn, D. A., Eds.; MIT Press: Cambridge, MA, 1999.
(47) Chapelle, O.; Vapnik, V.; Bousquet, O.; Mukherjee, S. Choosing Multiple Parameters for Support Vector Machines. *Mach. Learn.* **2002**, *46*, 131−159.
(48) Efron, B.; Tibshirani, R. Statistical Data Analysis in the Computer Age. *Science* **1991**, *253*, 390−395.
(49) Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119−139.
(50) Meyer, D.; Leisch, F.; Hornik, K. The Support Vector Machine Under Test. *Neurocomputing* **2003**, *55*, 169−186.
(51) Sutherland, J. J.; O'Brien, L. A.; Weaver, D. F. A Comparison of Methods for Modeling Quantitative Structure−Activity Relationships. *J. Med. Chem.* **2004**, *47*, 5541−5554.
(52) DePriest, S. A.; Mayer, D.; Naylor, C. B.; Marshall, G. R. 3D-QSAR of Angiotensin-Converting Enzyme and Thermolysin Inhibitors: A Comparison of CoMFA Models Based on Deduced and Experimentally Determined Active Site Geometries. *J. Am. Chem. Soc.* **1993**, *13*, 5372−5384.
(53) Golbraikh, A.; Bernard, P.; Chretien, J. R. Validation of Protein-Based Alignment in 3D Quantitative Structure−Activity Relationships with CoMFA Models. *Eur. J. Med. Chem.* **2000**, *35*, 123−136.
(54) Maddalena, D. J.; Johnston, G. A. R. Prediction of Receptor Properties and Binding Affinity of Ligands to Benzodiazepine/GABAA Receptors Using Artificial Neural Networks. *J. Med. Chem.* **1995**, *38*, 715−724.
(55) Chavatte, P.; Yous, S.; Marot, C.; Baurin, N.; Lesieur, D. Three-Dimensional Quantitative Structure−Activity Relationships of Cyclooxygenase-2 (COX-2) Inhibitors: A Comparative Molecular Field Analysis. *J. Med. Chem.* **2001**, *44*, 3223−3230.
(56) Sutherland, J. J.; Weaver, D. F. Three-Dimensional Quantitative Structure−Activity and Structure−Selectivity Relationships of Dihydrofolate Reductase Inhibitors. *J. Comput.-Aided Mol. Des.* **2004**, *18*, 309−331.
(57) Gohlke, H.; Klebe, G. DrugScore Meets CoMFA: Adaptation of Fields for Molecular Comparison (AFMoC) or How to Tailor Knowledge-Based Pair-Potentials to a Particular Protein. *J. Med. Chem.* **2002**, *45*, 4153−4170.
(58) Klebe, G.; Abraham, U.; Mietzner, T. Molecular Similarity Indices in a Comparative Analysis (CoMSIA) of Drug Molecules to Correlate and Predict Their Biological Activity. *J. Med. Chem.* **1994**, *37*, 4130−4146.
(59) Bohm, M.; Sturzebecher, J.; Klebe, G. Three-Dimensional Quantitative Structure−Activity Relationship Analyses Using Comparative Molecular Field Analysis and Comparative Molecular Similarity Indices Analysis To Elucidate Selectivity Differences of Inhibitors Binding to Trypsin, Thrombin, and Factor Xa. *J. Med. Chem.* **1999**, *42*, 458−477.
(60) *Cerius*^{2}; Accelrys Inc.: San Diego, CA.
(61) Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence*, Montréal, Québec, Canada, August 20−25, 1995; Morgan Kaufmann: Montréal, Québec, Canada, 1995; pp 1137−1145.
(62) Zar, J. H. *Biostatistical Analysis*, 4th ed.; Prentice Hall: Upper Saddle River, NJ, 1999; p 633.
(63) Bellman, R. E. *Adaptive Control Processes*; Princeton University Press: Princeton, NJ, 1961.
(64) Balaban, A. T. Highly Discriminating Distance-Based Topological Index. *Chem. Phys. Lett.* **1982**, *89*, 399−404.
(65) Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675−701.
(66) Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of *m* Rankings. *Ann. Math. Stat.* **1940**, *11*, 86−92.
(67) Triballeau, N.; Acher, F.; Brabet, I.; Pin, J. P.; Bertrand, H. O. Virtual Screening Workflow Development Guided by the "Receiver Operating Characteristic" Curve Approach. Application to High-Throughput Docking on Metabotropic Glutamate Receptor Subtype 4. *J. Med. Chem.* **2005**, *48*, 2534−2547.
(68) Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1−30.
(69) *R: A Language and Environment for Statistical Computing*; R Development Core Team: Vienna, Austria, 2006.
(70) Iman, R.; Davenport, J. Approximations of the Critical Region of the Friedman Statistic. *Commun. Stat. - Theor. Meth.* **1980**, *9*, 571−595.
(71) Nemenyi, P. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, 1963.

CI600332J