Induction of Decision Trees via Evolutionary Programming

Robert Kirk DeLisle*,† and Steven L. Dixon‡

Department of Molecular Modeling, Pharmacopeia, P.O. Box 5350, Princeton, New Jersey 08543-5350, and Schrödinger, 120 West 45th Street, 32nd Floor, New York, New York 10036

Received August 27, 2003

Decision trees have been used extensively in cheminformatics for modeling various biochemical endpoints including receptor−ligand binding, ADME properties, environmental impact, and toxicity. The traditional approach to inducing decision trees based upon a given training set of data involves recursive partitioning which selects partitioning variables and their values in a greedy manner to optimize a given measure of purity. This methodology has numerous benefits including classifier interpretability and the capability of modeling nonlinear relationships. The greedy nature of induction, however, may fail to elucidate underlying relationships between the data and endpoints. Using evolutionary programming, decision trees are induced which are significantly more accurate than trees induced by recursive partitioning. Furthermore, when assessed on previously unseen data in a 10-fold cross-validated manner, evolutionary programming induced trees exhibit a significantly higher accuracy on previously unseen data. This methodology is compared to single-tree and multiple-tree recursive partitioning in two domains (aerobic biodegradability and hepatotoxicity) and shown to produce less complex classifiers with average increases in predictive accuracy of 5−10% over the traditional method.

INTRODUCTION

Decision trees have been used extensively in numerous fields for exploratory data mining and pattern recognition. Their appeal for data analysis and as classifier systems stems primarily from three inherent properties: their ability to model nonlinear relationships, their ease of interpretability, and their nonmetric nature. Decision trees have the capacity to model complex data spaces, and, unlike traditional methods such as linear discriminant analysis (LDA), decision trees are capable of capturing nonlinear relationships within representative data. The XOR problem is a fundamental example of a nonlinear classification problem which is unresolvable via linear methodologies but can be modeled by decision trees. [The XOR problem is a prototypical nonlinearly separable classification problem. Consider four data points in two-dimensional space with coordinates (0,0), (0,1), (1,0), and (1,1), with (0,0) and (1,1) representing one class and (0,1) and (1,0) representing another. The result is a classification problem that cannot be solved by linear methodologies; however, nonlinear methods (e.g., neural networks, decision trees, etc.) are capable of a solution.] Given the expected level of complexity of typical cheminformatic and bioinformatic problems, an assumption of linear separability is rarely valid; thus, nonlinear classifiers may be required for robust data analysis. Furthermore, decision trees have been shown to yield classifiers which are competitive in performance with other nonlinear methods such as neural networks (NNs) and *k*-nearest neighbor (*k*-NN) classifiers.[1]

While capable of modeling nonlinear relationships, decision trees retain a high level of interpretability. The typical structure of a decision tree consists of a root node linked to two or more child nodes which may or may not link to further child nodes. Each nonterminal node within the tree represents a point of decision or data splitting based upon the training data; e.g., $x_1 < y_1$. (For convention in this publication, an answer of "yes" branches to the left child while a "no" answer branches to the right child.) Decisions are made in sequence until a terminal node, or leaf, is reached within which a classification is assigned. Effectively, the path of decisions can be interpreted as successive antecedents of IF...THEN clauses with the classification obtained from the resulting leaf node representing the consequent. Interpretation of the predicted classification can be obtained by direct analysis of the decision nodes in the path leading to this result.

Finally, decision trees are nonmetric and robust. Many pattern recognition methods are dependent upon the concept of distance within the data space, e.g., *k*-NN and LDA. This distance metric is dependent upon the existence of real-valued feature vectors and often is complicated by the presence of nominal data. Decision trees, however, are fully capable of modeling within nominal data space, and further, a mixture of real-valued and nominal data can be easily facilitated. Furthermore, the presence of a limited number of outliers is not necessarily detrimental to decision tree induction, providing a robust classifier system.[2]

In light of the benefits of their use, induction of decision trees is an area of active research and not a trivial task. The fundamental problem of selecting splitting rules has been studied extensively, and methods of selection can be categorized as derived from information theory, distance measure dependence, statistical hypothesis testing, and others.

* Corresponding author phone: (609) 919-6150; fax: (732) 422-0156; e-mail: kdelisle@pharmacop.com.
† Pharmacopeia.
‡ Schrödinger.

10.1021/ci034188s CCC: $27.50
Published on Web 03/20/2004

All of these represent different algorithmic approaches to recursive partitioning (RP)[3] and will be collectively referred to as RP methods. (It is beyond the scope of this paper to review these methods extensively; however, the interested reader is referred to ref 4 for a review.) The popular applications C4.5 and C5.0 utilize the minimum description length (MDL) principle, which is based upon information gain.[5] The statistical package S-Plus contains induction algorithms based upon deviance[6,7] which is an effective distance measure. Likewise, the Gini index and Twoing rule popularized by CART[2] are diversity -based distance measures. Hawkins and Kass[8] use statistical hypothesis testing for split point selection during tree induction.

While an abundance of feature selection methods is available far beyond those presented above, it has been shown that no particular method is superior to any other.[4] Moreover, the very nature of RP algorithms is to induce the construction of decision trees in a greedy fashion. Specifically, at any particular node the selection of a splitting rule is based solely upon the effect on the immediate children of that node. This greedy methodology is analogous to a steepest descent minimization scheme and will result in utilization of the feature most correlated with the endpoint through the selection criteria used regardless of the downstream impact of that choice. The most obvious solution to this problem is to impose a look-ahead methodology in which the impact is investigated at levels within the tree beyond the immediate children. One level look-ahead has been shown, however, to degrade decision tree quality,[4] and deeper levels become extremely computationally demanding. Moreover, the question of how far ahead to assess the effect of selection remains, a phenomenon known as the Horizon effect.[1]

Improvements in decision trees have been shown using simulated annealing in which multiple variables are used simultaneously within a single decision node.[9] This induction method remains effectively greedy, however, and the presence of multiple variables within a single decision node complicates the interpretability. Conceptually, improvements in decision trees could also be accomplished using genetic algorithms (GAs) in which the GA is used as a wrapper method to select a subset of the available features presented to a standard RP algorithm. This does not, however, address the issue of greedy induction. Alternatively, utilization of genetic programming (GP) for decision tree induction has also been shown beneficial and does indeed address the problem of greedy feature selection.[10] GP is a powerful technique within evolutionary computation; however, it suffers from the problem of bloat in which the resulting classifier structures tend to become exceedingly large and complex and thus difficult to interpret.[11] No explanation of how classifier bloat was controlled in these experiments or of the structure or interpretability of the resulting classifiers is given. An alternative strategy for producing regression trees in which the leaf nodes provide a real-valued prediction rather than a categorical prediction has involved Langdon ants.[12] While using the same structure as classification trees, regression trees are applied to continuous rather than discrete endpoints, and typically an averaging of the endpoints occurring within particular leaf nodes provides values for predictions.[2] Tree induction via Langdon ants would presumably be applicable to categorical data analysis; however, this has yet to be seen within the cheminformatics literature.

In this study, the results of inducing decision trees using evolutionary programming (EPTree), another powerful technique within evolutionary computation, are presented. This method avoids the problem of greedy induction as it functions and is assessed on the complete, intact classifier rather than a portion of the classifier as done by standard RP methods. Using two relevant data sets (specifically, aerobic biodegradability and hepatotoxicity), we show a significant improvement in prediction accuracy on external sets of observations when compared to trees produced by single-tree and multiple-tree RP methods. Furthermore, we are able to demonstrate consistent improvement in overall accuracy of the resulting decision trees on hold out samples by 10-fold cross-validation.

METHODS

Data Sets. Two real-world data sets were chosen for the purpose of evaluating the performance of inducing decision trees via evolutionary programming. The first data set, used extensively during initial development and refinement, consisted of 300 organic, small-molecule compounds for which aerobic biodegradability had been assessed by the Japanese Ministry of International Trade and Industry.[13] This data set was derived from a larger set of 852 compounds by random selection of 150 from each endpoint class—readily biodegradable and not readily biodegradable under aerobic conditions. This selection was done in order to provide a balance of each modeled class and to reduce the overall size of the data set, which was helpful during early methodological development. The compounds represent a diverse set of chemical classes including multiple-substituted benzene, single-substituted benzene, and aliphatic, alicyclic, and heterocyclic structures. Descriptors were generated using Cerius2 (Accelrys, San Diego), and the default descriptor set consisting of 37 descriptors was used. All descriptors were either physiochemical or topological in nature.

The second data set consisted of 436 structures with known hepatotoxicity endpoints and is identical to that used by Cheng and Dixon.[14] Data were obtained from various books,[15−18] public compilations,[19−23] and journal articles.[24−34] Briefly, descriptors consisted of an extensive set of Cerius2-generated descriptors (E-states, physiochemical properties, topological descriptors) and the novel one-dimensional pairwise similarity descriptor matrix developed by Dixon and Merz.[35] The complete set of descriptors was reduced to 25 using a simulated annealing selection algorithm which selected the combination of descriptors providing the best linear regression against the modeled endpoint, specifically, hepatotoxic or nonhepatotoxic.

Both initial data sets were partitioned into 10-fold cross-validation sets.[10,36] In this procedure, the data sets were randomly divided into 10 disjoint subsets, each of which maintained the original proportion of endpoint representatives, and then triplets were composed as follows. Eight subsets were recombined to construct the training set which was used during the evolution phase of evolutionary programming. One subset was used as a cross-validation set, and classifier performance was monitored on this set during training; however, the data within the subset had no impact upon the training phase. The final subset was retained as an external set and used only for final evaluation of classifier

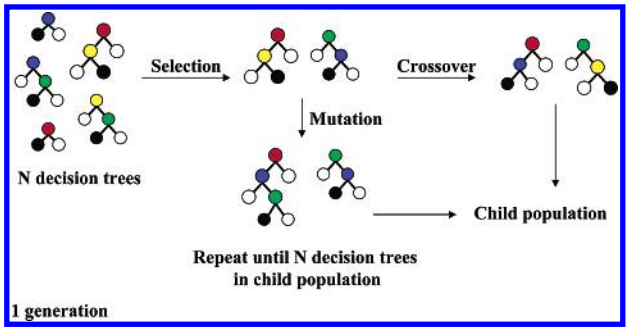

Figure 1. Evolutionary programming paradigm. Decision trees are shown with nodes color-coded according to the descriptor selected at that node as the splitting rule. Leaf nodes are coded black and white to represent distinct endpoints. The trees selected are subject to either crossover or mutation with probabilities of 0.3 and 0.7, respectively. Crossover consists of swapping random subtrees within each parent tree resulting in two child trees. Mutation (see Table 1) is performed on each tree independently. The left tree in this example has undergone regrowth (RegrowSubtree) of the subtree rooted at the yellow node. The right node could have undergone a shift in the value used for decision (ShiftRight or ShiftLeft) that did not alter the identity of the leaves.

accuracy on previously unseen data. (Further details of the usage of the three independent data sets are provided under *Evolutionary Programming*.) As 10 triplets consisting of a training, a cross-validation, and an external set were composed, and the 10 external sets were mutually exclusive, this methodology represents a "Leave-10%-Out" approach to model validation.

Evolutionary Programming. Evolutionary programming is a subdiscipline within evolutionary computation which differs significantly from the well-known genetic algorithm but still uses the fundamental principles of natural selection for population improvement.[16] Genetic algorithms utilize vector representations of the solution space typically taking the form of bit strings but are clearly not limited to this representation. This vector representation, typically referred to as a chromosome, is manipulated by crossover and mutation operators and requires an interpretation layer which constructs a solution from a given vector encoded representative. In contrast, evolutionary programming does not require an interpretation layer as the representatives are the actual data structures which, in this case, compose the decision tree classifier. All genetic operations are carried out directly upon the data structures themselves. Other aspects of evolutionary programming such as fitness, selection, and replacement, etc., are treated equivalently to other evolutionary computation methods.

The general paradigm, illustrated in Figure 1, consists of generation of an initial population of decision trees constructed at random based upon the training set and given mild constraints on the minimum number of observations allowed in terminal or leaf nodes. These constraints help to ensure the construction of a logically valid decision tree. Each tree within the population is assigned a fitness score derived from the accuracy of the tree for the training data as well as the overall complexity of the tree. A fitness-dependent selection method is used to select two members of the population, which are copied, returned to the original population, and the copies acted upon by either crossover or mutation. In the case of crossover, a subtree is randomly selected within each tree (the root node being exempt from selection) and subsequently swapped. Alternatively, a mutation operator (Table 1) is selected at random for each tree independently. Mutation operators have an equal probability of selection. Regardless of the chosen operator, two new decision trees result. These children are kept in a separate population until the number of children generated equals the number of individuals in the original population. At this point, the entire original population is deleted and replaced with the new population, an event considered as one generation. This evolution phase is then repeated for the desired number of generations.

Decision tree generation starts with all training data associated with a single (root) node and selection of a random descriptor and value for that descriptor directly from the training data. Observations are partitioned into the left-hand child if their values for the chosen descriptor are less than the selected value and into the right-hand child otherwise. This process continues for each child node and is limited only by the constraint that no leaf node can contain less than a preset number of observations. Specifically, 5 or 10% of the total number of data set observations were used to evaluate performance. This constraint prevented the creation of empty leaf nodes and also imposed a limitation on the overall complexity of the resulting tree. Once a leaf node has been reached, it is assigned a class based upon the observations present within. A simple majority rules voting method was used such that the endpoint class in the largest number determined the classification of the leaf.

The choice of an appropriate fitness function is critical to the success of any evolutionary computation approach. For the development of classifier systems, it is obvious that the accuracy of the classifier must play a role in determining the fitness of any proposed solution. Use of accuracy alone is not likely adequate, however, as it is possible to build an arbitrarily complex classifier by fitting the noise within the data set. Such a classifier would have high accuracy on the training set but would likely perform poorly on previously unseen data. This was indeed seen to occur in experiments where accuracy alone was used as the sole fitness parameter (data not shown) and can be seen in traditional RP approaches when the minimum number of observations allowed in leaves is set to an excessively small value.[2] To avoid this situation, the minimum description length (MDL) was used as the fitness function (eq 1) in order to score induced decision trees based on their complexity as well as their accuracy.[5,37]

Table 1. Mutation Operators

ShiftLeft	The value used in the decision node is shifted to the next lowest value for the current descriptor.
ShiftRight	The value used in the decision node is shifted to the next highest value for the current descriptor.
ShiftRandom	The value used in the decision node is shifted to a random value for the current descriptor.
ChangeDescriptor	A random descriptor and value is chosen from the data set. Child nodes retain their descriptors and decision values.
RegrowSubtree	The selected subtree is regrown randomly.

$$\text{MDL} = (\text{error coding length}) + (\text{tree coding length}) \quad (1)$$

$$\text{error coding length} = \sum_{x \in \text{leaves}} L(n_x,k_x) \quad (2)$$

$$L(n_x,k_x) = \log_2\left(\binom{n}{k}\right) \quad (3)$$

where

n = number of observations present within the leaf and
k = number of observations misclassified within the leaf

$$\text{tree coding length} = (n_i + n_t) + n_i \log_2 T_i + n_t \log_2 T_s \quad (4)$$

where n_i = number of internal nodes,
n_t = number of terminal nodes (leaves),
T_i = total possible number of splits, and
T_s = total number of classes.

The error coding length is based upon the binomial distribution and represents the number of combinations possible given the total number of observations (n) and the number of incorrectly predicted observations (k), or "n choose k." This relates to the likelihood of a particular (n,k) combination arising by random chance. Minimization of this value is desired. The tree coding length is dependent upon the overall size of the decision tree and also requires minimization. During development, it was determined that the only parameters necessary for the tree coding length were the number of nodes and the number of leaves. The remaining parameters merely contribute an additional constant amount to the fitness function given a particular training set, and their removal resulted in no alterations in performance apart from a reduced computational cost.

Selection of individuals from the population for the purpose of crossover or mutation can use any of a number of possible methods. Various selection methods were examined, and it was determined that two methods in particular were useful, specifically, fitness- based roulette wheel selection and tournament selection. In roulette wheel selection, the probability of selecting any particular member of the population is related to its fitness score with those having better fitness scores having a higher probability of selection. Tournament selection consists of choosing a specified number of population members at random with equal probability and determining which of those chosen has the better fitness score. While both methods are stochastic with a degree of fitness imposed bias, tournament selection with small numbers of individuals imposes a lower selection pressure and thus encourages exploration of the search space by allowing weaker members of the population a greater chance for survival. Roulette wheel selection imposes a higher level of selection and encourages exploitation of good members of the population and refinement of their characteristics. In later analyses it was found that tournament of four selection (in which four population members were chosen at random and the fittest member among them retained) was modestly superior to roulette selection as well as being computationally less demanding, and this method was used for the results presented herein.

Simulations were run for 1000 generations during which time the accuracy of the best decision tree of the current generation was monitored for the training and the cross-validation sets (Figure 2). It is well-known within the neural networks literature that during the process of network training, the error on the training set consistently decreases, whereas the error on an additional (cross-validation) data set will initially decrease but begin to increase at some point.[38,39] It is at this point that further training consists of fitting the idiosyncrasies of the training data and not the true relationship between the features and the endpoint, i.e., overtraining. A technique of early stopping in which training is terminated at the epoch prior to an increase in cross-validation-set error results in the most generalizable networks. Considering that the evolutionary programming process described above is attempting to reduce the error and complexity of classifiers based upon the training set, it must be realized that overtraining is a significant possibility, and this can clearly be seen in Figure 2. The cross-validation set thus served as a sentry to indicate the point at which evolution should be ceased, and the best decision tree (as assessed by the MDL score) of the early stopping generation used for prediction on the external test set, which up to this point was unseen by the classifier. The application for inducing decision trees via evolutionary programming as described above has been named EPTree.

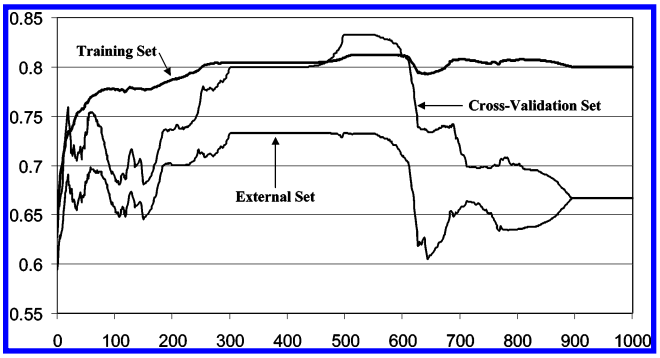

Figure 2. Early stopping. The average accuracy (Y-axis) for a population of decision trees modeling aerobic biodegradability is depicted per generation number with accuracy on the training set, cross-validation set, and external set illustrated. Generations 1−300 illustrate the improvement in accuracy of the population across all data sets, followed by a stable period. Generation 500 represents a viable early stopping point followed by a short, stable period after which overtraining is obvious from the stabilization of training set accuracy and significant decreases in cross-validation and external set accuracies.

Recursive Partitioning. For the purposes of comparison, decision trees were also induced by a standard, single-tree recursive partitioning algorithm as implemented in S-Plus 6.0. For these decision trees, the minimum leaf size was set to 5 or 10% of the total data set size, as was done in EPTree. As it is well-known that better results can be obtained by allowing particularly small numbers of observations in leaf nodes of decision trees followed by pruning of leaf nodes based on a secondary data set,[2] an additional series of decision trees were induced with S-Plus in which the minimum leaf size was set to five observations and the cross-validation set served as the pruning set. Furthermore, the cost-complexity pruning factor was adjusted manually in order to identify the degree of pruning which produced the best performance on the cross-validation and training sets. The external set was used as in EPTree to assess the generalizability of the decision tree induction.

In addition to the single-tree RP method described above, a multiple-tree RP method was also used for purposes of

Table 2. Average Accuracy Values for Aerobic Biodegradability

induction method[a]	training (%)	cross-validation (%)	external (%)	tree depth	decision nodes	emergent generation
EPTree30	82	80	79	4.2	6.2	120.3
EPTree15	85	80	79	5.7	9.5	185.1
RP30	77	75	71	3.8	5.9	
RP30+P	77	76	71	3.3	5.1	
RP15	83	73	74	6.0	11.4	
RP15+P	80	78	74	3.6	5.9	
RP5	92	70	72	9.1	23.6	
RP5+P	85	78	76	5.1	9.5	
RP30 (boot)	85	84	76	3.6	5.3	
RP15 (boot)	88	90	72	5.2	11.8	

[a] Methods are abbreviated as follows: EPTree, evolutionary programming induction; RP, recursive partitioning; numerals represent the minimum number of observations per leaf constraints; +P, cost-complexity pruning performed using the cross-validation set as the guiding data set; (boot), the training set was bootstrapped 1000 times and the best of 1000 trees selected based upon the cross-validation set.

comparison. Decision trees were generated on the basis of a bootstrapped training set derived from the original training set. The bootstrapped set was constructed by randomly sampling from the original training set with replacement until the number of observations in the bootstrapped set was equivalent to the number in the original training set. Decision trees were induced using the deviance method as described above, and this procedure was repeated 1000 times, resulting in 1000 trees. The best tree was determined from this collection of 1000 trees based upon the cross-validation set and used for comparisons.

Implementation. All programming was done using Microsoft Visual C++ 6.0 SP5, with ANSI standard C++ for coding of the classes, and Microsoft Foundation Classes for the user interface. Code was also transferred to a UNIX server running IRIX 6.5 where a command line interface was used for programmatic access, and code was compiled using the CC compiler with global optimization (−O2) and ISO/ANSI standards. The final code was able to generate 1000 random trees in less that 1 min. Simulatons of 1000 generations generally took less than 30 min. Simulations were run either on Intel PIII 1 GHz processors running Microsoft Windows 2000 Professional with 512 MB RAM or on a single R10000 processor of a UNIX server with 4 GB RAM.

RESULTS AND DISCUSSION

General Method. Decision trees were generated for all 10-fold cross-validated triplets across 25 independent, sequential runs for each triplet. Leaf size constraints were set to either 5 or 10% of the complete data set size resulting in 500 complete evolutions in total for each data set (25 runs for each constraint across 10-fold cross-validation). Likewise, each 10-fold triplet was analyzed using single-tree and multiple-tree RP (via S-Plus 6.0) with the same leaf size constraints and the additional constraint of five observations for the single-tree method. Pruning was performed in all RP analyses using the cross-validation subset and manually adjusting the cost-complexity parameter until accuracy on the cross-validation set was maximized with minimal impact on the training set accuracy, thus producing 60 total trees via RP for each data set (pruned and unpruned trees at three constraints across the 10-fold cross-validation). All trees produced were assessed for predictive accuracy on the corresponding training, cross-validation, and external prediction subsets and the results averaged for comparison.

Aerobic Biodegradability. A comparison of EPTree and RP performances on the aerobic biodegradability data set (Table 2) shows some significant and interesting trends. Most notably it is apparent that the average accuracy of EPTree exceeds that of single-tree RP on all three subsets (training, cross-validation, and external prediction), with the single exception being RP's accuracy on the training set when the minimum node size was set to a very small value (five observations) in which overtraining has likely occurred for the RP-generated trees. Further inspection of the results shows regardless of the constraint placed upon leaf size, EPTree-generated decision trees that were approximately 80% accurate for all three subsets. Compared to RP-generated trees with the constraint of 30 observations per leaf minimum, prediction accuracy was improved from 77% for the training set, 76% for the cross-validation set, and 71% for the external prediction set. EPTree evolved trees also compared very favorably to RP-generated trees with minimum leaf sizes of 15 and 5 observations regardless of the minimum number of observations per leaf constraints. Furthermore, while the use of pruning after RP induction did indeed enhance performance on the external data set while reducing classifier complexity, EPTree resulted in improved tree accuracies without introducing additional complexity and thus did not require pruning. The use of a multiple-tree RP methodology in which the training set was bootstrapped improved the RP results, particularly for the higher leaf size constraint of 30 compounds minimum per leaf node. This result is not surprising; however, it is interesting to note that EPTree-generated trees were still more accurate on the hold out structures with a comparable level of complexity. Classification accuracy on the cross-validation set was largely improved by the multiple-tree method; however, accuracy on this set is somewhat biased as it was used for selection of the best tree from those produced by bootstrapping. The average accuracy for the 1000 trees produced by multiple-tree RP was approximately 60−65% for all three data sets. The best decision trees obtained from EPTree at each constraint level are illustrated in Figure 3. While the best RP-induced tree (Figure 3C) is comparable in accuracy to the EPTree-induced tree on training, cross-validation, and external subsets, the complexity is reduced substantially by EPTree. Furthermore, when evaluated on the remaining 552 compounds of the original data set, an increase of 3% and 10% in predictive accuracy results for EPTree with the highest and lowest leaf size constraints, respectively.

When results are evaluated on each of the 10-fold cross-validation triplets individually, the mean accuracy of EPTree-

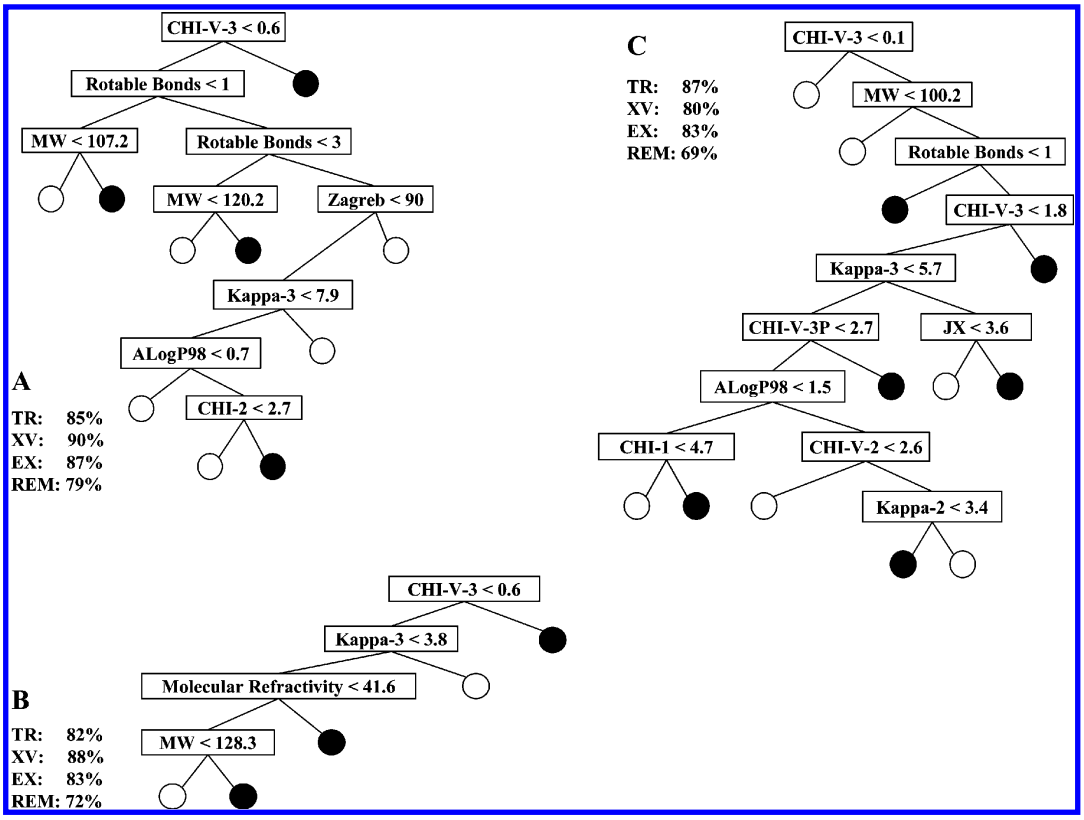

Figure 3. Decision trees predicting aerobic biodegradability. The left branch at each decision node corresponds to agreement with the decision condition at that node while the right branch represents disagreement. Leaves are colored black to represent nonbiodegradable and white for biodegradable under aerobic conditions. Classification accuracies are listed in the figure as follows: TR, training set; XV, cross-validation set; EX, external (hold out) set; REM, the remaining 552 compounds from the original data set. (A) The best decision tree identified under the minimum of 15 observations per leaf constraint was obtained at the 131st generation for the third cross-validation triplet. (B) The best decision tree identified under the minimum of 30 observations per leaf constraint was obtained at the 49th generation for the fourth cross-validation triplet. (C) The best tree produced by RP with a minimum leaf size of five followed by pruning using the cross-validation set as a guide for the ninth cross-validation triplet.

Table 3. Average Accuracy Values for Hepatotoxicity

induction method[a]	training (%)	cross-validation (%)	external (%)	tree depth	decision nodes	emergent generation
EPTree44	78	77	76	5.1	6.2	168.5
EPTree22	81	78	77	7.1	11.2	242.5
RP44	75	69	68	4.1	6.2	
RP44+P	75	70	68	4.0	6.1	
RP22	80	71	71	6.0	12.0	
RP22+P	79	72	72	5.0	9.2	
RP5	92	68	68	11.2	36.0	
RP5+P	87	75	71	8.4	20.3	
RP44 (boot)	81	83	71	4.1	5.1	
RP22 (boot)	84	85	69	6.2	11.2	

[a] Methods are abbreviated as follows: EPTree, evolutionary programming induction; RP, recursive partitioning; numerals represent the minimum number of observations per leaf constraints; +P, cost-complexity pruning performed using the cross-validation set as the guiding data set; (boot), the training set was bootstrapped 1000 times and the best of 1000 trees selected based upon the cross-validation set.

produced trees on the training, cross-validation, and external subsets is generally higher than the trees produced by RP. Furthermore, using a single-tailed *t*-test of significance, ρ values range from 0.05 to less than 0.0005. Occasionally, however, a decision tree induced by RP performs better or is equivalent on one of the subsets in a particular fold. This is most often seen for the training set and RP-induced trees with small minimum leaf size values prior to any pruning operations, particularly when compared to EPTree-induced trees with high minimum leaf sizes. This is not surprising, as these conditions would suggest a degree of overtraining is likely to exist within these RP classifiers. It can be seen, however, that the frequency of EPTree producing classifiers with statistically significant improved accuracy far outweighs the frequency of producing statistically worse or equivalent classifiers (Figure 4A). It is noteworthy to realize that those trees produced by EPTree with the highest minimum leaf size setting were statistically superior to trees produced by RP regardless of leaf size 52 out of 60 times with regard to predictive accuracy on the external set. Furthermore, all 52 improvements had ρ values of 0.005 or less.

Hepatotoxicity. As was seen with the aerobic biodegradability data set, an average improvement of 5−10% accuracy was seen in EPTree-generated trees relative to RP trees (Table 3). Notably, decision trees generated with a minimum leaf size of 44 exhibited at least 76% accuracy of prediction

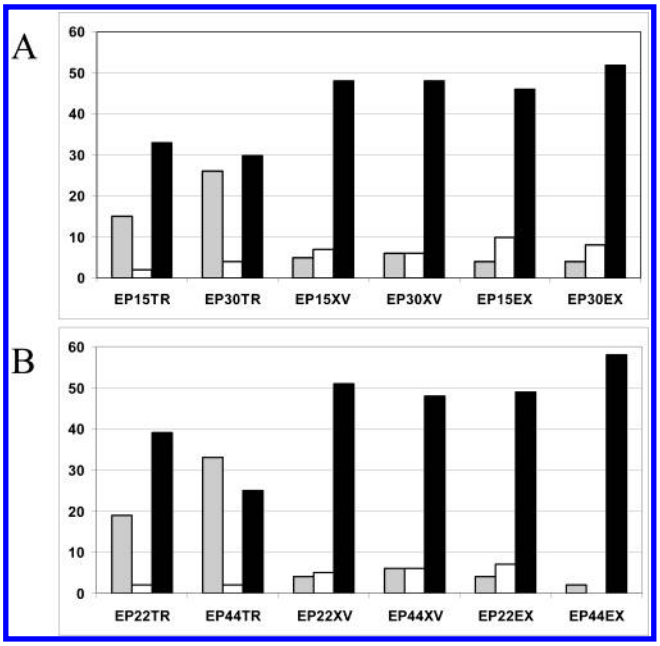

Figure 4. Frequency of evolving significantly better decision trees. Within each cross-validation fold, the distribution of accuracy values for evolved decision trees was compared to the performances of RP-induced trees using a single-tailed *t*-test. The distribution of EPTree accuracy values for each constraint (listed on the *X*-axis) was compared individually to all six RP methods (minimum leaf sizes of 5 and 10%, and 5 observations, with and without pruning) for training (TR), cross-validation (XV), and external (EX) data subsets. ρ values of 0.05 or less were considered significant. Gray, white, and black bars represent counts of RP-induced trees which were statistically better, equivalent, or worse than EPTree-induced trees with the illustrated leaf size constraints and on the illustrated data subset. (A) Results for the aerobic biodegradability data set. Clusters of bars are labeled with the number of observations defining the constraint (15 or 30) and the subset as listed above. (B) Results for the hepatotoxicity data set. Clusters of bars are labeled with the number of observations defining the constraint (22 or 44) and the data subset.

across all three data sets and had on average seven leaves and six decision nodes. Those produced under the constraint of having no less than 22 compounds in leaves were at least 77% accurate on all three data sets and were more complex, averaging 11 leaves. The improvement of prediction accuracy by as much as 10% over RP is highly significant; however, it becomes even more compelling when the overall complexity of the resulting decision trees is considered in conjunction with accuracy. Decision trees produced by RP with a minimum leaf size of 44 compounds were similar in complexity to EPTree-produced trees under the same constraint; however, an accuracy of only 68% on the external subset was seen. The maximum accuracy trees generated by RP were still on the order of 72% accurate while expanding to as many as 19 decision nodes and 20 leaves. Clearly in this situation interpretation of the decision trees' details for the purpose of practical application of the discovered relationship is severely complicated by excessive complexity. Moreover, the more highly accurate decision trees with dramatically reduced complexity generated by EPTree represent a presumably more fundamental relationship that was not identified using the standard RP approach. The multiple-tree RP methodology boosted performance on the external set predictions; however, as seen with aerobic biodegradability, EPTree retained higher accuracy on the external set with a comparable level of tree complexity. Performance on the cross-validation set was also similar to that seen with aerobic biodegradability where accuracy was somewhat improved. As mentioned previously, this is not too surprising considering this set was used for selection of the best tree from the 1000 produced by the multiple-tree method.

An assessment of the performance on the 10-fold subsets individually held details similar to that done for the aerobic biodegradability data set (Figure 4B). Generally, when making comparisons regardless of minimum leaf size constraints, EPTree-induced trees were not as predictive on the training set but much more highly predictive for the cross-validation and external sets. With regard to the training set, the identification of less accurate classifiers actually outnumbered the number of statistically better RP-induced trees. It should be noted, however, that when compared to only those RP-induced trees in which the minimum leaf size criteria was equal to the EPTree-induced trees, EPTree produced significantly better classifiers in every case with $\rho < 0.001$. When a smaller minimum leaf size is used in RP as compared to the EPTree constraint, a higher training set accuracy is seen. Furthermore, as shown in Figure 4B, EPTree-induced decision trees with the highest minimum node size constraint (and thus the least complex trees) were statistically superior for predictions on the external set in 58 of 60 comparisons with RP-induced trees regardless of the leaf size constraints at a $\rho < 0.005$. A simpler analysis of these results is that RP-induced trees tended to be more accurate for the training set and less accurate for the cross-validation and external data sets, suggesting a lack of ability to predict outside the training set.

As mentioned previously, the hepatotoxicity data set is identical to that used by Cheng and Dixon[14] in which ensembles of decision trees were produced in order to improve the overall accuracy of predictions.[40] For the purpose of the studies presented thus far, the training and test sets developed by Cheng and Dixon were pooled and subsequently divided into 10-fold cross-validated sets. To make a direct and more valid comparison between the two techniques,[39] compounds were removed at random from the original training set to produce a cross-validation set of similar size to the 10-fold cross-validation subsets, and the same test set as described by Cheng and Dixon was used. Decision trees which were evolved by EPTree constrained to having no less than 35 compounds in leaves averaged 78% accurate on the external test set, which is significantly improved compared to similarly constrained RP-induced trees which were merely 56% accurate—a remarkable 22% improvement in predictive accuracy. The multiple-tree RP method improved external predictions as well to 69%; however, this value remains significantly lower than accuracy levels seen with EPTree. Moreover, individual trees produced by EPTree are competitive with ensembles produced by Cheng and Dixon that consisted of 151 individual trees resulting in 81% accuracy on the test set of compounds. While their results illustrate the power and utility of combining classifiers for improving prediction accuracies, improved accuracy comes at the cost of lost interpretability of the resulting ensemble. It is striking to note that the best evolution run (out of 25 independent, sequential runs) resulted in a single tree that was 76, 86, and 84% accurate on training, cross-validation, and external sets, respectively,

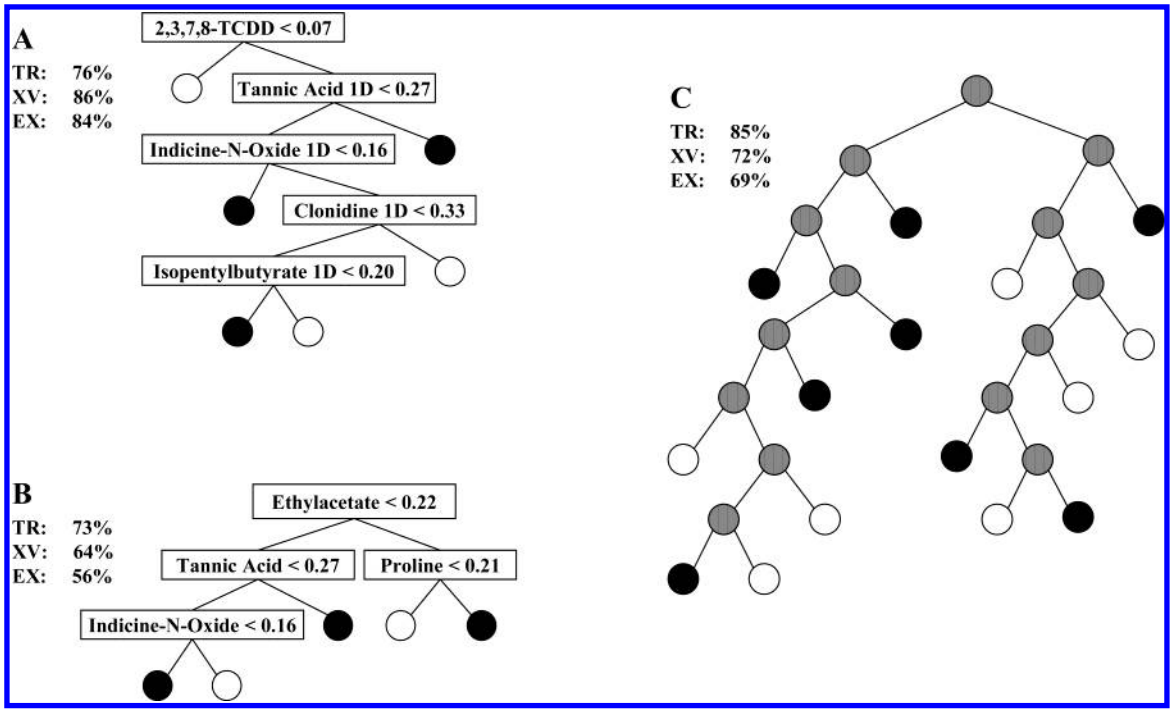

Figure 5. Decision trees predicting hepatotoxicity. The same conventional details are followed as in Figure 3. Descriptors shown in boxes are 1D similarity values.[35] Leaves are colored black to represent hepatotoxic and white for nonhepatotoxic classifications. (A) The best decision tree identified under the minimum of 35 observations per leaf constraint was obtained at the 56th generation. (B) The decision tree produced by RP which was obtained under the minimum of 35 observations per leaf constraint with pruning. (C) The decision tree produced by RP with minimum observations per leaf set to five with pruning performed to enhance generalizability. Labels are omitted for illustration clarity.

with only five decision nodes and thus six leaves (Figure 5A). The best tree produced by RP under the same constraints (Figure 5B) was 73, 64, and 56% accurate on the training, cross-validation, and external sets, respectively, with little change in complexity. Furthermore, when using the currently accepted approach of overgrowing decision trees followed by pruning to enhance generalization, accuracies on the same subsets were 85, 72, and 69% with 16 decision nodes (Figure 5C). This is obviously a significant increase in complexity and a dramatic decrease in the interpretability of the classifier, while the accuracy remains noncompetitive with the EPTree-induced tree.

CONCLUSION

The utility of decision trees as predictive classifiers or for exploratory data analysis is demonstrated by their extensive use in various scientific domains, including cheminformatics and bioinformatics. By their very nature, decision trees are capable of modeling nonlinear relationships between features and endpoints and provide highly interpretable classifiers. Furthermore, their lack of parametric dependence allows their application to data spaces that do not necessarily have properties of normal distribution. Induction of decision trees, however, has classically been accomplished by greedy methods that may lack the ability to elucidate subtle relationships within the data that are dependent upon combinations of descriptors, the XOR problem being a good example.

The application of evolutionary programming to the task of inducing decision trees resulted in classifiers that averaged 5−10% more accurate on previously unseen data as compared to trees generated by single-tree or multiple-tree RP methods. Moreover, the overall complexity of the resulting trees was typically reduced. Trees modeling aerobic biodegradability, for example, generally averaged 80% accurate on all data sets regardless of the constraints placed on tree induction. It is interesting to consider the least complex of these, containing as few as 4−6 decision nodes, in comparison to RP-induced trees, which were comparable in accuracy but consisted of as many as 30 or more decision nodes. This dramatic reduction in classifier complexity suggests that the relationships discovered by EPTree are fundamentally more descriptive of the true relationship present between the representation space and the observational space of the data set.

Similar results were seen with the hepatotoxicity data set in which EPTree-induced decision tree accuracies averaged 76%, again an improvement of 5−10% with a reduction or no change in complexity. These improvements are similar to those seen by Cheng and Dixon in which ensembles of 151 decision trees were produced to improve classification accuracies. In stark contrast, however, EPTree was able to identify single trees with accuracies competing favorably with the 151 tree ensemble. More importantly, while both EPTree and the ensemble method produced classifiers that were 81% accurate on previously unseen data, single-tree RP methods result in a decision tree only 56% accurate. This highly significant degree of improvement emphasizes the utility of both the evolutionary programming and ensemble methodologies, with a further degree of favor placed with EPTree due to its ability to provide a single decision tree with a 25% improvement in accuracy on unseen data.

Further work is currently underway in which the use of EPTree to generate ensembles is being investigated.[41,42] Due

to the stochastic nature of the evolutionary programming methodology, independent runs of EPTree does result in distinctly different decision trees. Preliminary investigations in which the 25 trees produced for a particular cross-validation fold are combined into a single ensemble have shown even further enhancements in predictive accuracies (data not shown); however, duplicate trees exist within this ensemble. An underlying principle of creating ensembles of classifiers is that each member of the ensemble should exhibit failure on comparatively different observations within the data set. Methods to produce distinctly different classifiers with unique accuracies across observations while maintaining the improved global accuracies are being investigated. Additionally, the use of decision trees as feature selection devices for other modeling methods has been explored[43] and will be investigated for EPTree-induced trees.

ACKNOWLEDGMENT

The authors wish to thank Dr. David J. Diller and Dr. Susan M. Keenan for extensive discussions, assistance in coding, and thorough review of this manuscript.

REFERENCES AND NOTES

(1) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*, 2nd ed.; Wiley-Interscience: New York, 2000.
(2) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and Regression Trees*; Wadsworth International Group: Belmont, CA, 1984.
(3) Rusinko, A., 3rd; 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. Comput. Sci.* **1999**, *39*, 1017−1026.
(4) Murthy, S. K. Automatic Construction of Decision Trees from Data: A Multi-disciplinary Survey. *Data Min. Knowl. Discovery* **1998**, *2*, 345−389.
(5) Quinlan, J. R.; Rivest, R. L. Inferring Decision Trees Using the Minimum Description Length Principle. *Inf. Comput.* **1989**, *80*, 227−248.
(6) Dixon, S. L.; Villar, H. O. Investigation of Classification Methods for the Prediction of Activity in Diverse Chemical Libraries. *J. Comput.-Aided Mol. Des.* **1999**, *13*, 533−545.
(7) Crawley, M. J. *Statistical Computing: An Introduction to Data Analysis Using S-Plus*, 1st ed.; John Wiley & Sons: Ltd.: West Sussex, England, 2002.
(8) Hawkins, D. M.; Kass, G. V. In *Topics in Applied Multivariate Analysis*; Hawkins, D. M., Ed.; Cambridge University Press: Cambridge, U.K., 1982; pp 269−302.
(9) Blower, P.; Fligner, M.; Verducci, J.; Bjoraker, J. On Combining Recursive Partitioning and Simulated Annealing To Detect Groups of Biologically Active Compounds. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 393−404.
(10) Bains, W.; Gilbert, R.; Sviridenko, L.; Gascon, J. M.; Scoffin, R.; Birchall, K.; Harvey, I.; Caldwell, J. Evolutionary Computational Methods To Predict Oral Bioavailability. QSPRs. *Curr. Opin. Drug Discovery Dev.* **2002**, *5*, 44−51.
(11) Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA; London, England, 1992.
(12) Izrailev, S.; Agrafiotis, D. A Novel Method for Building Regression Tree Models for QSAR Based on Artificial Ant Colony Systems. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 176−180.
(13) Loonen, H.; Lindgren, F.; Hansen, B.; Karcher, W.; Niemela, J.; Hiromatsu, K.; Takatsuki, M.; Peijnenburg, W.; Rorije, E.; Struijs, J. Prediction of Biodegradability from Chemical Structure: Modeling of Ready Biodegradation Test Data. *Environ. Toxicol. Chem.* **1999**, *18*, 1763−1768.
(14) Cheng, A.; Dixon, S. L. In Silico Models for the Prediction of Dose-Dependent Human Hepatotoxicity. *J. Comput.-Aided Mol. Des.*, in press.
(15) Farrell, G. C. *Drug-Induced Liver Disease*, 1st ed.; Churchill Livingstone: New York, 1994.
(16) Zimmermann, H. J. *Hepatotoxicity: The Adverse Effects of Drugs and Other Chemicals on the Liver*, 2nd ed.; Lippincott Williams & Wilkins: Philadelphia, 1999.
(17) *Source Book of Flavors*, 2nd ed.; Chapman & Hall: New York, 1994.
(18) Stricker, B. H. C. *Drug-Induced Hepatic Injury*, 2nd ed.; Elsevier: Amsterdam, 1992.
(19) *Physicians' Desk Reference: Electronic Library*; Thomson Micromedex, Inc.: Greenwood Village, CO, 2001.
(20) *Registry of Toxic Effects of Chemical Substances (RTECS)*; The National Institute for Occupational Safety and Health: Cincinnati, OH.
(21) *Toxicology Data Network (TOXNET)*; U.S. National Library of Medicine: Bethesda, MD.
(22) *FDA Generally Recognized as Safe (GRAS) List*; U. S. Food and Drug Administration: Rockville, MD.
(23) *Fenaroli's Handbook of Flavor Ingredients*, 3rd ed.; CRC Press: Boca Raton, FL, 1995.
(24) Gold, E. J.; Mertelsmann, R. H.; Itri, L. M.; Gee, T.; Arlin, Z.; Kempin, S.; Clarkson, B.; Moore, M. A. Phase I Clinical Trial of 13-*cis*-Retinoic Acid in Myelodysplastic Syndromes. *Cancer Treat. Rep.* **1983**, *67*, 981−986.
(25) Vogel, C. L.; Gorowski, E.; Davila, E.; Eisenberger, M.; Kosinski, J.; Agarwal, R. P.; Savaraj, N. Phase I Clinical Trial and Pharmacokinetics of Weekly ICRF-187 (NSC 169780) Infusion in Patients with Solid Tumors. *Invest. New Drugs* **1987**, *5*, 187−198.
(26) von Mehren, M.; Giantonio, B. J.; McAleer, C.; Schilder, R.; McPhillips, J.; O'Dwyer, P. J. Phase I Trial of Ilmofosine as a 24 h Infusion Weekly. *Invest. New Drugs* **1995**, *13*, 205−210.
(27) Ryan, D. P.; Supko, J. G.; Eder, J. P.; Seiden, M. V.; Demetri, G.; Lynch, T. J.; Fischman, A. J.; Davis, J.; Jimeno, J.; Clark, J. W. Phase I and Pharmacokinetic Study of Ecteinascidin 743 Administered as a 72-hour Continuous Intravenous Infusion in Patients with Solid Malignancies. *Clin. Cancer Res.* **2001**, *7*, 231−242.
(28) Rowinsky, E. K.; Noe, D. A.; Ettinger, D. S.; Christian, M. C.; Lubejko, B. G.; Fishman, E. K.; Sartorius, S. E.; Boyd, M. R.; Donehower, R. C. Phase I and Pharmacological Study of the Pulmonary Cytotoxin 4-Ipomeanol on a Single Dose Schedule in Lung Cancer Patients: Hepatotoxicity Is Dose Limiting in Humans. *Cancer Res.* **1993**, *53*, 1794−1801.
(29) Raber, M. N.; Newman, R. A.; Newman, B. M.; Gaver, R. C.; Schacter, L. P. Phase I Trial and Clinical Pharmacology of Elsamitrucin. *Cancer Res.* **1992**, *52*, 1406−1410.
(30) O'Brien, J. T.; Eagger, S.; Levy, R. Effects of Tetrahydroaminoacridine on Liver Function in Patients with Alzheimer's Disease. *Age Aging* **1991**, *20*, 129−131.
(31) Lakhanpal, S.; Donehower, R. C.; Rowinsky, E. K. Phase II Study of 4-Ipomeanol, a Naturally Occurring Alkylating Furan, in Patients with Advanced Hepatocellular Carcinoma. *Invest. New Drugs* **2001**, *19*, 69−76.
(32) Lagadic-Gossmann, D.; Rissel, M.; Le Bot, M. A.; Guillouzo, A. Toxic Effects of Tacrine on Primary Hepatocytes and Liver Epithelial Cells in Culture. *Cell Biol. Toxicol.* **1998**, *14*, 361−373.
(33) Knip, M.; Douek, I. F.; Moore, W. P.; Gillmor, H. A.; McLean, A. E.; Bingley, P. J.; Gale, E. A. Safety of High-Dose Nicotinamide: A Review. *Diabetologia* **2000**, *43*, 1337−1345.
(34) Guzzo, C.; Benik, K.; Lazarus, G.; Johnson, J.; Weinstein, G. Treatment of Psoriasis with Piritrexim, a Lipid-Soluble Folate Antagonist. *Arch. Dermatol.* **1991**, *127*, 511−514.
(35) Dixon, S. L.; Merz, K. M., Jr. One-Dimensional Molecular Representations and Similarity Calculations: Methodology and Validation. *J. Med. Chem.* **2001**, *44*, 3795−3809.
(36) Jain, A. K.; Duin, R. P. W.; Mao, J. Statistical Pattern Recognition: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 4−37.
(37) Baeck, T.; Fogel, D. B.; Michalewicz, Z.; Eds. *Evolutionary Computation 1: Basic Algorithms and Operators*; Institute of Physics Publishing: Bristol, U.K., Philadelphia, PA, 2000; Vol. 1.
(38) Tetko, I. V.; Livingstone, D. J.; Luik, A. I. Neural Network Studies. 1. Comparison of Overfitting and Overtraining. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 826−833.
(39) Tetko, I. V.; Villa, A. E. P.; Livingstone, D. J. Neural Network Studies. 2. Variable Selection. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 794−803.
(40) Susnow, R. G.; Dixon, S. L. Use of Robust Classification Techniques for the Prediction of Human Cytochrome P450 2D6 Inhibition. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1308−1315.
(41) Tong, W.; Hong, H.; Fang, H.; Xie, Q.; Perkins, R. Decision Forest: Combining the Predictions of Multiple Independent Decision Tree Models. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 525−531.
(42) 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. Comput. Sci.* **2003**, *43*, 1947−1958.
(43) Ratanamahatana, C. A.; Gunopulos, D. Feature Selection for the Naive Bayesian Classifier Using Decision Trees. *Appl. Artif. Intell.* **2003**, *17*, 475−487.

CI034188S