Evaluation of Mutual Information and Genetic Programming for Feature Selection in QSAR

Vishwesh Venkatraman,[†] Andrew Rowland Dalby,[†] and Zheng Rong Yang*,[‡]

School of Biological Sciences, University of Exeter, Exeter EX4 4QF, Great Britain and School of Engineering and Computer Science, University of Exeter, Exeter EX4 4QF, Great Britain

Received February 17, 2004

Feature selection is a key step in Quantitative Structure Activity Relationship (QSAR) analysis. Chance correlations and multicollinearity are two major problems often encountered when attempting to find generalized QSAR models for use in drug design. Optimal QSAR models require an objective variable relevance analysis step for producing robust classifiers with low complexity and good predictive accuracy. Genetic algorithms coupled with information theoretic approaches such as mutual information have been used to find near-optimal solutions to such multicriteria optimization problems. In this paper, we describe a novel approach for analyzing QSAR data based on these methods. Our experiments with the Thrombin dataset, previously studied as part of the KDD (Knowledge Discovery and Data Mining) Cup 2001 demonstrate the feasibility of this approach. It has been found that it is important to take into account the data distribution, the rule "interestingness", and the need to look at more invariant and monotonic measures of feature selection.

INTRODUCTION

A QSAR model is a mathematical relationship between a set of physicochemical descriptors (structural, geometric, etc.) and a property (biological activity, solubility, etc.) of the system being studied. Based on this assumption of structure−activity correlation, new active compounds can be devised that not only bind, but also have all the other properties required for a drug, by combining models of drug action and the needs of bioavailability. Existing compound libraries provide a huge set of chemical descriptors to model the activity of interest. Given the multivariate nature of such data, the process of feature selection and modeling can be quite challenging. The task requires a solution of several complicated problems:

(1) Redundancy exhibited by the multitude of features tends to exert an undue influence in the analysis, giving rise to misleading associations between the variables.[1]

(2) Existence of an inherent nonlinearity between most descriptors and observed activity.[2]

(3) Presence of a relatively small set of molecules with well-established bioactivities of interest.[3]

(4) Skewed nature (generally) of QSAR data. Standard algorithms often fail to produce robust models as they assume that training examples are evenly distributed among different classes.

(5) Multitude of features would provide more discriminating power but may also be accompanied by a drop in generalization performance owing to the redundant and irrelevant information in the input.

(6) Combinatorial nature of the feature selection problem (2^n possible combinations of n available features) makes the process computationally intensive.

Large feature sets can be handled through a relevance analysis step yielding a finite subset of the given features that contain enough information to define a system with satisfactory performance. Most of these techniques, however, tend to choose features incrementally and may not focus on combinations whose components are not individually predictive. Numerous artificial-intelligence based approaches[4−9] have therefore focused on identifying information-rich combinations of features yielding models delivering desired levels of accuracy, robustness, and interpretability.

In building a classification model, the information about the class that is inherent in the features is of utmost importance. Thus, an evaluation of the information content of each individual feature with regard to the output class can be used as a measure for the stochastic dependency of discrete random variables. Information theory[10−12] provides intuitive tools to quantify how much information is required to describe random quantities or how much information they share. Well-studied applications include computer-aided diagnosis,[13] microarray gene selection,[14] and protein sequence structure compatibility.[15]

Choices of the level of complexity and the best features to combine remain the key issues of a successful combination. Following Occam's principle,[16] one tries to find a suitable combination of features that optimizes complexity and classification performance. Optimization strategies based on principles of evolution have been used with considerable success in dealing with such combinatorial problems.[17,18] Efforts have also been directed toward achieving greater accuracy and reliability based on the idea of Pareto optimality.[19] As both optimization and regression need to be intertwined,[20] combining evolutionary algorithms with information theory provides a suitable framework that is well suited to tackle such multiobjective problems. The main advantage lies in their being able to effectively sample large

* Corresponding author e-mail: Z.R.Yang@exeter.ac.uk.
[†] School of Biological Sciences, University of Exeter.
[‡] School of Engineering and Computer Science, University of Exeter.

10.1021/ci049933v CCC: $27.50 © 2004 American Chemical Society
Published on Web 08/11/2004

search spaces while investigating multiple solutions simultaneously,[21−23] leading to a near-optimal combination of features that predict the response.

In this study, we investigate the use of information-theoretic approaches based on the concept of mutual information gain and genetic programming to determine the optimal set of features. Mutual information is a good indicator of the relevance between variables. Being a nonlinear statistical criterion, it is able to measure the interdependence of random variables without making any assumptions of their underlying relationships.[24] Mutual information gain filtering produces a finite subset of features that excludes features that have a weak correlation with the target variable. After the most relevant features have been selected, the next stage is to build models (mapping) for biological activity prediction. In this paper, we use genetic programming to produce models using symbolic regression. This approach offers a multimodal search mechanism exposing possible relationships while taking advantage of correlations only available in combination. However even with all the rules (models) having evolved with respect to data-driven constraints, the solutions may be ineffective**.** Padmanabhan et al.[25] observe that it is possible to produce a large number of rules that are interesting objectively, but of little interest to the user. In this paper we look at the concept of interestingness of a rule incorporating both objective and domain-specific measures. Our experiments with the thrombin dataset, previously studied as part of the KDD (Knowledge Discovery and Data Mining) Cup 2001[26] demonstrate the feasibility of this approach to feature selection for knowledge discovery.

METHOD

Mutual Information Feature Selection. Feature selection works on the principle that a strongly dependent variable would add little information about the output and therefore be removed. For any variable to be selected certain heuristic criteria need to be satisfied:

(1) Feature should be comparatively informative about the output.

(2) Feature should not be strongly dependent on other features selected.

The mutual information between two random variables A and B is a measure of the information about A contained in B and vice versa. If the random variables are independent of each other the mutual information is zero. The mutual information between A and B is defined as in eq 1.

$$I(A,B) = \sum_{a,b} P(a,b)\log\frac{P(a,b)}{P(a)P(b)} \qquad (1)$$

The marginal probabilities for the two features are represented by *P(a)* and *P(b)*, while *P(a,b)* gives the joint probability. Mutual Information measures the distance between the joint probability *P(a,b)* and the probability *P(a)P(b)*, which is the joint probability under the assumption of independence. For a multivariate problem, feature selection can be done by deriving *I(a,b)* for each feature with respect to the class label and selecting the top-ranking features in terms of the mutual information.

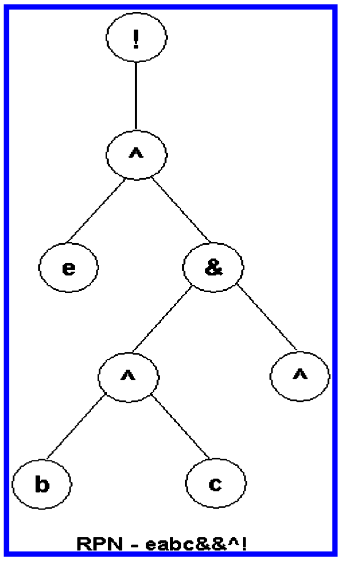


Figure 1. Rule represented in RPN and its corresponding tree implementation.

Genetic Programming Formulation. Genetic algorithms (GA) have shown to be very powerful optimization techniques by focusing on the application of selection, mutation and crossover (recombination) to a population of candidate solutions.[21] Koza[27] developed a hierarchical extension to GA by demonstrating that it is possible to automatically generate programs to perform specified tasks. Genetic programming (GP) consists of creating a population of randomly selected combinations of features where each combination can be considered as a possible solution to the feature-selection problem. The use of symbolic regression by GP enables determination of both model structure and complexity during the evolution process. Being stochastic approaches, they also offer advantages in terms of their relative insensitivity to noise and the ability to work without proper domain knowledge.[28,29]

In this article, we adopt a stack-based approach to create and evolve the population of individuals.[30] This approach offers advantages in terms of efficiency and simplicity of implementation. Since the functions receive the arguments from a numerical stack, the implementation automatically blends itself into a Reverse Polish Notation (RPN). Parse trees of arbitrary lengths can be formed that also eases the formation of legal executable programs, which is a symbolic regression function in this study. Postfix ordering is beneficial since the operand to each operator is evaluated before processing the operator. This saves time that would otherwise be spent in finding the operands.[31] Figure 1 shows a program in its postfix notation and the corresponding tree structure.

We have implemented an elimination-based *k*-tournament selection[32] that removes the weakest of the *k*-selected individuals. Each eliminated individual is immediately replaced by reproduction. To minimize the selection pressure the tournament size was limited to two.

Data Set. As an experimental dataset to test our model, we used the thrombin dataset provided by DuPont pharmaceuticals for the Knowledge and Drug Discovery (KDD) cup 2001. The objective of this cup was to learn a classifier that can effectively predict whether an organic molecule can bind well to a target site on thrombin, a key receptor in blood clotting, given the chemical structure of the compound. Each

Table 1. Dataset Characteristics

	compounds	features	positives	negatives	skew
training set	1909	139351	42	1867	1:44
testing set	634	139351	150	484	1:3

Table 2. Probability Estimates for Minority Class Examples

frequency-based estimate	Laplace estimate	corrected Laplace estimate
TP/*TP*+*FP*	*TP*+1/*TP*+*FP*+2	*TP*+*bm*/*TP*+*FP*+*m*

compound is represented by a feature vector consisting of 139 351 binary features (0: Inactive, 1: Active) that describe the three-dimensional properties of the compound. The test set had a higher proportion of positives as it was made based on the activity results for the training set. No prior biological knowledge was made available; since the data was highly skewed, the system assessment used a differential cost model so that the sum of the costs of the actives was equal to the sum of the costs of the inactives. This was expressed in terms of the unweighted average of the accuracies of the true actives and the accuracy on the true inactives. In addition, the training set has 593 of 1909 samples containing all zeros (none of the bits are 1) with 2 of them being active compounds. Table 1 gives the statistics of the two sets.

In the absence of a properly defined misclassification cost function and the difficulty associated with performing a k-fold crossvalidation, the task represents a difficult challenge for any learning algorithm.

Tackling Class Distributions. Predictive modeling approaches often use an error criterion or an accuracy-based criterion to build models. This is a good decision criterion when the data have a symmetric distribution. Complications arise when the data are skewed with one of the classes being heavily underrepresented as compared to the other class. Kubat and Matwin[33] used undersampling, which eliminates examples in the majority class, making the data less imbalanced. However, there is a danger of discarding potentially useful data. Weiss et al.[34] contend that the failure of classifiers in classifying minority class examples is because the marginal probabilities in the natural training distribution are biased strongly in favor of the majority class. They advocate the use of frequency estimates that are sensitive to the training set distributions. The Laplace estimate[35] has been used in several cases and has been found to be worthwhile in terms of estimating class priors. The probability estimate will be closer to 0.5 (1/*C* where *C* is the number of classes) than the frequency-based estimate but the difference will be marginal for large sample sizes.

Since conditional probabilities need to be smoothed toward the corresponding unconditional probability, Elkan et al. suggest a corrected form of the Laplace estimate. Table 2 lists the common probability estimates used.

In the corrected Laplace estimate, *TP* represents the number of true positives while *FP* is the number of false positives; *b* is the base rate of the positive class and *m* is a parameter that controls how much scores are shifted toward *b*. As an example, if the number of minority class examples is 10 and the majority class examples is 80, the base rate would be 10:80 or 0.125. The Laplace estimates are not used to assign class labels, as it can never move the probability estimate across the threshold. Given a base rate *b*, Elkan et al.[36] suggest that the *m* value be adjusted so that $bm = 10$. As the value of *m* increases, the observed training set frequencies are shifted toward the base rate. Since the probability smoothing is similar for a wide range of *m* values, the precise value chosen for *m* does not have a great impact on performance. The choice of *m* is therefore heuristic or can be chosen using crossvalidation. Table 3 gives the probability estimates with different values.

Table 3. Corrected Frequency Estimates for Different *m* Values

TP	FP	base rate (*b*)	bm	m	estimate
25	1852	0.02245	10	445	0.01507321
25	1852	0.02245	14	624	0.01559376

Table 4. GP Parameters

terminal set	{a, b, c, d...},(10−50)
function set	AND, XOR, NOT
population size	200−1000
selection method	tournament selection (2)
generations	25−125

Methodology. We apply a three-pronged approach to the problem. Since it is impossible for any algorithm to handle so many features, it is important that relevance analysis be performed to reduce the dimensionality of the data. The first phase therefore involves selecting features maximally informative about the target. By assessing the impact of different feature subsets on the classification performance, we could direct the search for an approximate optimal subset. In the second phase, the reduced feature set is fed to the genetic algorithm, which then performs a symbolic regression to obtain solutions that cater to the fitness criteria. To avoid overfitting we reduce the number of generations and keep the number of models to a minimum.[37,38] Table 4 lists the parameters used for the GP.

Having obtained a raw rule set, we apply rule interestingness measures to filter out programs that do not satisfy the interestingness criterion. In general, unexpected rules (models) are by definition interesting.[39] Given that the user already knows the high information gain features, programs (models) containing them would be of less interest to him/her as compared to ones that contained features with lower information gain. According to Freitas,[40] attribute interactions can render an individually irrelevant feature into a relevant one, thus introducing an element of surprise.

Feature Filtering. *Step 1.* For the given set of features, compute the mutual information between the features and the target variable ("Activity") using eq 1.

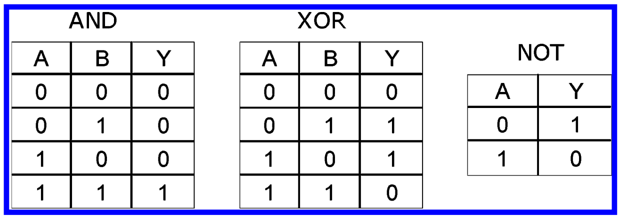
Step 2. Sort the features in terms of their mutual information gain in decreasing order.

Step 3. Filter out the required set of features that have gains less than an arbitrary threshold value.

Since the data are binary in nature the probabilities are estimated from the frequency counts. Thus eq 1 now becomes

$$I(A,B) = P(A = a, B = b)\log\frac{P(A = a, B = b)}{P(A = a)P(B = b)} \tag{1a}$$

Evolutionary Algorithm. The evolutionary process is driven by the fitness measure. With reference to an unbalanced data set, fitness-functions-based F-measures[41] and other accuracy-based measures fail to tackle class imbalance and end up being either too sensitive or too specific. We use the



AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NOT

A	Y
0	1
1	0

Figure 2. Semantics for AND, XOR, NOT logic gates.

Laplace-corrected estimate in its sigmoid form as estimates with better bias tend to perform better.

$$Confidence = \frac{(TP + bm)}{TP + FP + m} \qquad (2)$$

$$Fitness = \frac{1}{(1 + \exp(-1*(1 - Confidence))} \qquad (3)$$

Function Set. Since the values were binary, we implemented the program using Boolean functions (AND, XOR, NOT), the semantics of which are described in Figure 2.

Algorithm. *Step 1.* Generate N rules by randomly combining different feature vectors (terminal set) and the operators (XOR, AND, NOT) such that the rules created form a legal executable code.

Step 2. Evaluate the fitness of the population using eq 2. If the highest fitness exceeds a predefined threshold, terminate. Else, proceed to the next step.

Step 3. Evolve the population using tournament selection without replacement.

(1) Enter all individuals in the tournament.

(2) Randomly select any two individuals to compete.

(3) Compare the fitness of the two individuals; the individual with the greatest fitness is considered the victor. Choose the individual with lesser complexity if fitness values are equal.

(4) Add the victor to the mating pool.

(5) Remove the loser from the tournament.

(6) Repeat steps 2−6 until only one individual is left in the tournament.

(7) Randomly select an operation: reproduction, crossover, or mutation.

Mutation: Randomly select a rule and a random mutation point. If the token at the point is a function mutate the function else mutate the terminal.

Crossover: Randomly select two rules, say A and B, and two crossover points that are functions. Merge adjacent parts of the two rules to create new individuals.

If the individuals exceed a predefined length or form illegal programs, repeat the process until two well-formed individuals are created.

(8) Repeat step 7 until the next generation is full (mating pool is empty).

Step 4. Go to Step 2.

Step 5. Stop.

Rule Selection. Based on the confidence, the complexity, and the average information gain obtained for each rule, we choose the rules that have low complexity, low information gain, and decent confidence levels. Balancing all three at any given time is a difficult task, with the added problem of having certain rules that perform poorly and might leave out others (exceptions) that do not satisfy these criteria. By defining minimum and maximum values of confidence, the user can select those that conform to specific interest criteria. Note that these criteria may not only be objective but may also involve a certain amount of subjective decision-making. The interestingness is calculated as the average of the information gain of the features occurring in the rule. We use eq 3 as a measure of the interestingness where *InfoGain*(*F_i*) is the information gain of the i^{th} feature in the rule and *n* is the number of features occurring in the rule.

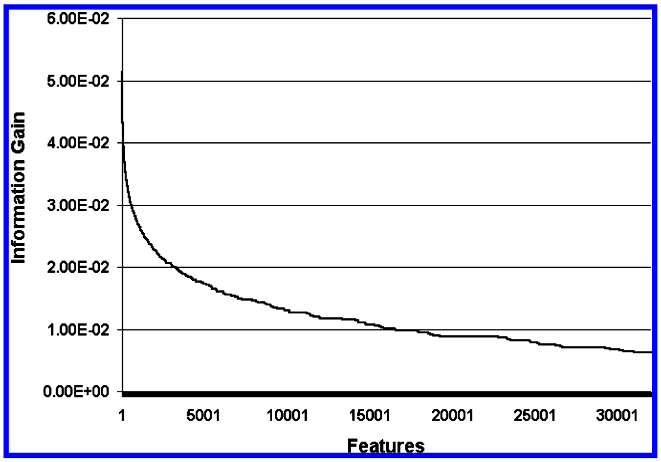


Figure 3. Mutual information gain plot for the top 30 000 features.

Table 5. Results for the Thrombin dataset. Table shows the number of features given to the genetic program, the number of cycles, the population size, and the model statistics in terms of its complexity and weighted accuracy

features	generations	population size	weighted accuracy	best accuracy	complexity (no. of features)
10	25	200	70.59	73.16	4−8
20	50	400	63.2	66.65	10−14
30	75	600	75.6	77.7	18−20
40	100	700	66.8	68.52	24−28
50	125	1000	62	63.07	35−40

$$Interestingness = \frac{\sum InfoGain(F_i)}{n} \qquad (3)$$

RESULTS AND DISCUSSION

As can be seen from Figure 3, very few features have a strong correlation with the output. By varying the threshold, we could select the top 10−50 features that had the maximum relevance to the target. Adding too many features would increase the program space, which in turn can affect the efficiency of the genetic program in evolving a solution. Table 5 gives an account of the performances of the different runs. The performance seems to drop as the number of features increases. Our best results were produced with 10 features (terminal set) achieving a weighted accuracy of 70.59% over the top 20 models that performed well in the testing. The loss in accuracy with the increase in the number of features could be attributed to overfitting in the absence of any complexity penalty.

Our best model achieved a balanced test accuracy of 73.16% with just 4 features. Figure 4 shows the model represented as a Boolean logic circuit. Our implementation of the model as a logic circuit has the advantage of an easily realized hardware, which can bring about a manifold

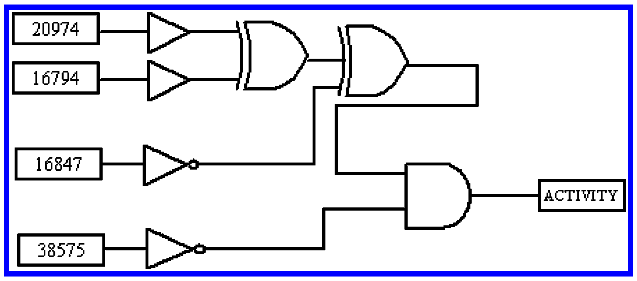


Figure 4. Logic circuit of the best performing model (Numbers in the boxes indicate the feature vector numbers).

Table 6. Comparison of the Naïve Bayes with Our Method

	TP	TN	FP	FN	best accuracy (%)	weighted accuracy (%)	complexity (no. of features)
GP + MI	120	321	163	30	73.16	70.59	4
naïve Bayes (tan)	95	356	128	55	71.1	68.4	4

Table 7. Sensitivity and Specificity Comparison

	sensitivity (%)	specificity (%)
GP + MI	80	66.32
naïve Bayes (tan)	63.33	73.55

acceleration in the prediction process. In the KDD competition the winner[42] used a tree-augmented Naïve-Bayes network. Table 6 shows the comparison of our algorithm with that of the Naïve-Bayes method. Their result on the test set was a weighted accuracy of 68.4% over 5 models. Their best model identified more negatives as compared to positives. From Table 7, one can observe that the higher-balanced accuracy estimates are due to greater sensitivity with a slight reduction in the specificity. The only similarity between the two methods, other than the use of the information gain metric, was the absence of crossvalidation. The runners-up in the competition used decision trees[43] to achieve a weighted accuracy of 64%.

The candidate models generated by the genetic program were selected on the basis of their confidence, complexity, and entropy. When comparing their performances on the test set, there were still a few exceptions that did perform well despite not having satisfied our criteria. This suggests the use of a more refined form of selection criteria. The models created with just 10 features were all quite simple, with just 4−8 features, but varied in their prediction accuracies. With an increase in the population size along with the generations, there was a manifold increase in time complexity, which was further affected by the absence of complexity penalties except in terms of selection. Having gained awareness of the overfitting problem, we can now divert our attention toward improving the performance with a higher set of features.

In the KDD competition only 7% of the 114 contest entrants achieved a weighted accuracy of 60% and above. Since then, other algorithms have looked to integrate information in the test data set. Weston et al.[44] achieved an 81% success rate with just 10 features by employing a transductive inference algorithm that used unlabeled test feature vectors in the training stage. Since laboratory testing can be quite expensive and time-consuming, chemists would prefer to test the most positive predictions first and proceed down an ordered list. Having identified a satisfactory number of binding compounds, they would be able to avoid other expenses. Forman used an incremental learn and retrain algorithm thta obviates the concept drift between the training and testing sets.[45]

In drug design, ideally, the rules discovered through QSAR analysis should be successful in predicting the properties of previously unseen compounds. A priori knowledge of the underlying chemistry may also be considered. Extracting useful information from highly interlinked data is often a time-consuming and difficult task. Predictive models often fail in domains that have one of the classes in the minority, which gets worse with the skewness. In the absence of well-identified misclassification costs and having to handle high dimensional data, most learning methods end up with very poor results.

Although mutual information was successful in alienating most of the features, there were several variables that exhibited similar information gain values. Ideally, we would have liked the features to be nondominated or noninferior.[46] Such inconsistencies in the selection process can lead to a flawed feature subset selection. Recent studies[47] have suggested that mutual information is not an altogether invariant measure, as it contains marginal entropies, and advocate the use of normalized mutual information as a better measure of the prediction that one variable can do about the other.

The combination of information-theoretic approaches with genetic programming has yielded promising results that are accurate and interpretable. The problem of data imbalance has been solved to some extent using a Laplace-corrected frequency estimate, which has been successfully applied in the past in several algorithms. Other invariant forms of feature selection based on unbalanced correlation[48] and joint mutual information[49] have been more effective in removing redundancies and could be applied in future in QSAR.

Imamura et al.[50] have used N-version genetic programming that tends to avoid overfitting and the lack of generalization problems caused by small samples. Ensemble approaches have been able to improve prediction accuracies in many cases[51,52] ahead of individual learners. From a rule-interestingness point of view, rules would need to satisfy both objective and subjective criteria; both of which will have to be incorporated into the evolutionary mechanism.

Active learning[53] heuristics have also performed quite well in the drug discovery process. By performing selective sampling, data requirements have been reduced considerably. Given that evolutionary algorithms operate on a search-based principle, further research can be directed toward integrating active learning techniques in genetic programming.

CONCLUSIONS

In this paper, a novel method for feature selection in QSAR by means of a combination of information theory and evolutionary computation is presented. The applicability of the method is demonstrated using a highly skewed dataset. In particular, it has been shown that the method can be used to realize a logic circuit capable of predicting the structure−function relationship. Pronounced differences have been observed in the prediction of active compounds during comparisons with results obtained from the KDD cup winners. The work carried out in this research recognizes the significance of factors such as data distribution and rule interestingness in the process of feature selection.

ACKNOWLEDGMENT

The authors would like to thank DuPont Pharmaceuticals for providing the thrombin dataset and the organizers of the 2001 KDD cup for arranging the competition. We also thank the reviewers for their helpful comments in modifying the paper.

REFERENCES AND NOTES

(1) Topliss, J. G.; Edwards, R. P. Chance factor in studies of quantitative structure−activity relationships. *J. Med. Chem.* **1979**, *22*, 1238−1244.

(2) Whitley, D. C.; Martyn, F. G.; David, L. J. Unsupervised Forward Selection: A Method for Eliminating Redundant Variables. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1160−1168.

(3) Eriksson, L.; Johansson, E.; Kettaneh-Wold, N.; Wold, S. *Multi- and Megavariate Data Analysis − Principles and Applications*; Umetrics: 2001.

(4) Rogers, D.; Hopfinger, A. J. Application of Genetic Function Approximation to Quantitative Structure−Activity Relationships and Quantitative Structure−Property Relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854−866.

(5) Hasegawa, K.; Funatsu, K. GA Strategy for Variable Selection in QSAR Studies: GAPLS and D-optimal Designs for Predictive QSAR Model. *J. Mol. Struct. (THEOCHEM)* **1998**, *425*, 255−262.

(6) Trotter, M.; Burbidge, R.; Buxton, B.; Holden, S. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Comput. Chem.* **2001**, *26*, 5−14.

(7) Burden, F. R.; Winkler, D. Q. Robust QSAR Models Using Bayesian Regularized Bayesian Networks. *J. Med. Chem.* **1999**, *42*, 3183−3187.

(8) Zheng, W.; Tropsha, A. A Novel Variable Selection Quantitative Structure−Property Relationship Approach Based on the k-Nearest-Neighbour Principle. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 185−194.

(9) Agrafiotis, D. K.; Izrailev, S. A new method for building regression tree models for QSAR based on artificial ant colony systems. *J. Chem. Inf. Comput. Sci.* **2001**, *41*(1), 176−180.

(10) Shannon, C. E. A mathematical theory of communication. *AT&T Technol. J.* **1948**, *27*, 379−423.

(11) Cover, T. M.; Thomas, J. A. *Elements of Information Theory*; Wiley: New York, 1991.

(12) Kullback, S. *Information Theory and Statistics*; Wiley: New York, 1959.

(13) Tourassi, G. D.; Frederick, E. D.; Markey, M. K.; Floyd, C. E., Jr. Application of the mutual information criterion for feature selection in computer-aided diagnosis. *J. Med. Phys.* **2001**, *28*, 2394−2402.

(14) Ding, C.; Peng, H. Minimum Redundancy Feature Selection for Gene Expression Data. *IEEE Computer Society Bioinformatics Conference (CSB '03)*, Stanford, CA, 2003; 523−529.

(15) Lin, K.; May, C. W. A.; Taylor, W. R. Threading Using Neural network (TUNE): the measure of protein-sequence structure compatibility. *Bioinformatics* **2002**, *18*, 1350−1357.

(16) Domingoes, P. A process-oriented heuristic for model selection. *Data. Min. Knowl. Discuss.* **1999**, *3*(4), 409−425.

(17) Terfloth, L.; Gasteiger, J. Neural networks and genetic algorithms in drug design. *Drug Discov. Today* **2001**, *6*, 102−108.

(18) Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Longman: Reading, MA, 1989.

(19) Nicolotti, O.; Gillet, V. J.; Fleming, J. P.; Green, V. S. D. Multi-objective Optimization in Quantitative Structure−Activity Relationships: Deriving Accurate and Interpretable QSARs. *J. Med. Chem.* **2002**, *45*(23), 5069−5080.

(20) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford Press: 1995.

(21) Punch, W. F.; Goodman, E. D.; Min Pei; Lai Chia-Shun; Hovland, P.; Enbody, R. In *Fifth International Conference on Genetic Algorithms*; Forrest, S., Ed.; Morgan Kaufmann: San Mateo, 1993; p 557.

(22) So, S.-S.; Karplus, M. Evolutionary Optimization in Quantitative Structure−Activity Relationship: An Application of Genetic Neural Networks. *J. Med. Chem.* **1996**, *39*(7), 1521−1530.

(23) Kailin, T.; Li, T. Combining PLS with GA-GP for QSAR. *Chemom. Intell. Lab. Syst.* **2002**, *64*, 55−64.

(24) Yang, Z. R.; Zwolinski, M. Mutual information theory for adaptive mixture models. *IEEE Trans. Pattern. Anal.* **2001**, *23*(4), 396−403.

(25) Padmanabhan, B.; Tuzhilin, A. A belief-driven method for discovering unexpected patterns. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD '98)*, New York, 1998; p 94.

(26) *KDD 2001*, Annual KDD cup. http://www.cs.wisc.edu/~dpage/kddcup2001.

(27) Koza, J. R. *Genetic Programming*; MIT Press: Cambridge, 1995.

(28) Gilbert, R. J.; Goodacre, R.; Woodward, A. M.; Kell, D. B. Genetic Programming: A Novel Method for the Quantitative Analysis of Pyrolysis Mass Spectral Data. *Anal. Chem.* **1997**, *69*, 4381−4389.

(29) Hasegawa, K.; Kimura, T.; Funatsu, K. GA Strategy for Variable Selection in QSAR Studies: Application of GA-Based Region Selection to a 3D-QSAR Study of Acetylcholinesterase Inhibitors. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 112−120.

(30) Perkis, T. Stack Based Genetic Programming. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*; IEEE Press: Orlando, 1994.

(31) Yang, Z. R.; Thomson, C.; Hodgman, C.; Doyle, A. K. Extracting decision rules from protein sequences using genetic programming methods. *BioSystems* **2003**, *72*, 159−176.

(32) Blickle, T.; Thiele, L. A Mathematical Analysis of Tournament Selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA95)*; Eshelman, L. J., Ed.; Morgan Kaufmann: 1995.

(33) Kubat, M.; Matwin, S. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *Proceedings of the 14th International Conference on Machine Learning*; Morgan Kaufmann: 1997; p 179.

(34) Weiss, G.; Provost, F. *The Effect of Class Distribution on Classifier Learning: An Empirical Study*; Technical Report ML-TR-44 2001; Department of Computer Science, Rutgers University.

(35) Good, I. J. *The Estimation of Probabilities*; M. I. T Press: Cambridge, 1965.

(36) Elkan, C.; Zadrozny, B. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD)*; 2001; p 204.

(37) Becker, L. A.; Seshadri, M. *Comprehensibility & Overfitting Avoidance in Genetic Programming for Technical Trading Rules*; Computer Science Technical Report WPI-CS-TR-03-09 2003; Worcester Polytechnic Institute.

(38) Schaffer, C. *Machine Learning* **1993**, *10*, 153−178.

(39) Piatetsky-Shapiro, G.; Matheus, C. J. The interestingness of deviations. In *Proceedings of the 11th International Conference on Artificial Intelligence*; AAAI Press: 1994.

(40) Freitas, A. A. On objective measures of rule surprisingness. In *Proceedings of the Second European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD'98)*; Springer-Verlag: Nantes, 1998; p 1.

(41) Van Rijsbergen, C. J. *Information Retrieval*; Butterworth: London, 1979.

(42) Cheng, J.; Hatzis, C.; Hayashi, H.; Krogel, M. A.; Morishita, S.; Page, D.; Sese, J. KDD Cup 2001 Report. *ACM SIGKDD Explorations* **2001**, *3*(2), 47.

(43) Sarawagi, S.; Anurandha, B.; Janakiraman, A.; Haritsa, J. Building classifiers with unrepresentative training instances: Experiences from the KDD cup 2001 competition. In *Proceedings of Workshop on Data Mining, Lessons Learnt held in conjunction with the International Conference on Machine Learning*; Sydney, 2002.

(44) Weston, J.; Pérez-Cruz, F.; Bousquet, O.; Chapelle, O.; Elisseeff, A.; Schölkopf, B. Feature Selection and Transduction for Prediction of Molecular Bioactivity for Drug Design. *Bioinformatics* **2003**, *19*, 764−771.

(45) Forman, G. A Method for Discovering the Insignificance of One's Best Classifier and the Unlearnability of a Classification Task. In *Data Mining Lessons Learned Workshop 19th International Conference on Machine Learning (ICML)*, Sydney, 2002; Hewlett-Packard Tech Report HPL; p 123.

(46) Billings, S.; Zheng, G. L. Radial Basis Function Network Configuration Using Mutual Information and the Orthogonal Least Squares Algorithm. *Neural Networks* **1996**, *9*(9), 1619−1637.

(47) Hernández, A.; Edgar, A. G. E. C.; Coello, C.; Carlos, A. Synthesis of Boolean Functions using Information Theory. In *Lecture Notes in Computer Science*; Tyrell, A. M., Haddow, P. C., Torresen, J., Eds.; Springer: Norway, 2003, p 218.

(48) Hall, M. A. Correlation-based Feature Selection for Machine Learning. Ph.D. Thesis, Waikato University, New Zealand, 1999.

(49) Yang, H.; Moody, J. Feature selection based on joint mutual information. In *Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*; Rochester, NY, 1999.

(50) Imammura, K.; Robert, B.; Heckendorn, Soule, T., Foster, A. J. N−Version Genetic Programming via Fault Masking. In *Proceedings of the 5th European Conference on Genetic Programming*; 2002, p 172.

(51) Langdon, W. B.; Buxton, B. F. Genetic Programming for Combining Classifiers. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*; Morgan Kaufmann: San Francisco, 2001.

(52) Zhang, B. T.; Joung, J. G. Building Optimal Committees of Genetic Programs. *Lect. Notes Comput. Sci.* **2000**, *1917*, 231.

(53) Warmuth, M. K.; Liao, J.; Rätsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active Learning with SVMs in the Drug Discovery Process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*(2), 667−673.

CI049933V