On the Use of Neural Network Ensembles in QSAR and QSPR

Dimitris K. Agrafiotis,* Walter Cedeño, and Victor S. Lobanov

3-Dimensional Pharmaceuticals, Inc., 665 Stockton Drive, Exton, Pennsylvania 19341

Received March 26, 2002

Despite their growing popularity among neural network practitioners, ensemble methods have not been widely adopted in structure−activity and structure−property correlation. Neural networks are inherently unstable, in that small changes in the training set and/or training parameters can lead to large changes in their generalization performance. Recent research has shown that by capitalizing on the diversity of the individual models, ensemble techniques can minimize uncertainty and produce more stable and accurate predictors. In this work, we present a critical assessment of the most common ensemble technique known as bootstrap aggregation, or bagging, as applied to QSAR and QSPR. Although aggregation does offer definitive advantages, we demonstrate that bagging may not be the best possible choice and that simpler techniques such as retraining with the full sample can often produce superior results. These findings are rationalized using Krogh and Vedelsby's decomposition of the generalization error into a term that measures the average generalization performance of the individual networks and a term that measures the diversity among them. For networks that are designed to resist over-fitting, the benefits of aggregation are clear but not overwhelming.

I. INTRODUCTION

Artificial neural networks are rapidly becoming the method of choice for structure−activity and structure−property correlation.[1−11] Neural networks are model-free mapping devices that are capable of capturing complex nonlinear relationships in the underlying data that are often missed by conventional QSAR approaches such as multilinear regression[12] and partial least squares.[13] Just like many other techniques of this kind, these systems work by correlating some experimentally determined measure of biological activity with a set of physicochemical, structural, and/or electronic parameters (descriptors) of the compounds under investigation. Their use involves a training phase in which the model parameters are determined from a set of training data, an optional but highly recommended validation phase in which the generalization ability of the model is established, and a prediction phase in which the biological properties of novel compounds are computed using the optimized model. Since it is not possible to know a priori which molecular properties are most relevant to the problem at hand, neural networks are often used in conjunction with optimization techniques for feature selection, ranging from simple greedy approaches such as forward selection or backward elimination,[14] to more elaborate methodologies such as automatic relevance determination,[15] simulated annealing,[16] evolutionary programming,[17] genetic algorithms[18−21] and, most recently, artificial ant colony systems[22,23] and particle swarms.[24]

However, neural networks are known to be unstable, in the sense that minor changes in the training data and/or training parameters can have serious consequences in the generalization ability of the resulting models. Regretfully, this instability is rarely addressed in the QSAR literature. Indeed, it is still not uncommon for studies based on a single validation set to make their way into respectable journals, even though it is well-known that the underlying predictor(s) may perform well on a particular test set and abysmally on another. *N*-fold cross-validation is a much better choice, but it too can fall prey to the random number generator. A much better approach would be to run a sufficiently large number of *n*-fold cross-validation runs (typically 50 or 100) and report the distribution of the resulting generalization statistics (minimally, the mean and standard deviation). This is particularly important in comparative studies where the differences in the generalization performance of the underlying predictors are relatively small. Still, these types of procedures can only provide a better estimate of the true generalization error of the model but do nothing to improve it.

Recent research has shown that the accuracy of unstable methods can be significantly improved through aggregation. The idea is to construct multiple instances of a predictor by perturbing the training set or the construction method and combine them into a single model using voting (classification) or averaging (regression). Although the idea has been around for a long time (its origins in the neural network literature can be traced back to 1965[25]), it has not received much attention until recently, largely because it requires significant computational resources, especially when applied to learners such as neural networks. In more formal terms, let $\boldsymbol{x}$ be a training pattern in a training set $T = \{(\boldsymbol{x}_i, y_i), i = 1, ..., N\}$, generated according to

$$y = f(\boldsymbol{x}) + \epsilon(\boldsymbol{x}) \tag{1}$$

where y is the observed response value, $f(\boldsymbol{x})$ is the output of the physical system, and $\epsilon(\boldsymbol{x})$ is noise with zero mean. The

* Corresponding author phone: (610)458-6045; fax: (610)458-8249; e-mail: dimitris@3dp.com.

10.1021/ci0203702 CCC: $22.00
Published on Web 05/22/2002

regression task is to approximate the function $f(\boldsymbol{x})$. Let $\phi(\boldsymbol{x})$ denote such an approximation. Aggregation attempts to improve this approximation by generating multiple versions of the predictor, $\phi_b(\boldsymbol{x})$, and combining their outputs in some prescribed way, typically by averaging

$$\phi_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\phi_b(\boldsymbol{x}) \tag{2}$$

where $\phi_{\text{bag}}(\boldsymbol{x})$ is the aggregated predictor, and B is the cardinality of the ensemble. Note that in the following discussion, $\phi_{\text{bag}}(\boldsymbol{x})$ will refer to a general ensemble, not necessarily one that is derived through bootstrap aggregation (see below).

Obviously, combining the output of multiple predictors is useful only if there is disagreement between them. Model diversity can be introduced by (1) manipulating the input features (feature selection), (2) randomizing the training procedure (over-fitting, under-fitting, training with different topologies and/or training parameters, etc.), (3) manipulating the response value (adding noise), or (4) manipulating the training set. The latter has received the most attention, and three techniques have come to dominance: (1) bagging,[26] (2) boosting,[27] and (3) stacking.[28,29]

Bagging is an acronym for "bootstrap aggregation" and was originally proposed by Breiman in connection to classification and regression trees.[26] The method uses the bootstrap, a very popular statistical resampling technique, to generate multiple training sets, which are used to train the members of the ensemble. If the training set T consists of N cases, each is assigned a probability of $1/N$, and a new training set, T_B, is assembled by sampling with replacement N times from the original training set, using these probabilities. Some cases in T may not appear in T_B, while others may appear multiple times. The resampled training set T_B is used to train a predictor, the process is repeated, and the results are combined to form a consensus prediction. Breiman found that bagging gave dramatic improvements when applied to classification and regression trees. Works that study bagging in the context of neural networks include refs 30−32.

Boosting is a related technique that attempts to drive the test set error rapidly to zero.[27] Unlike bagging, boosting produces a *series* of predictors. The training set used for each member of the series is based on the performance of the preceding predictor(s). The method creates new training sets by choosing patterns for which the predictions of the previous predictors were bad more frequently than those for which the predictions were good. Thus, boosting attempts to produce new predictors for its ensemble that are able to make better predictions for patterns for which the current ensemble performance is poor. As with bagging, the resampled training set is assembled using probabilistic selection, with the exception that the probability assigned to each sample depends on the prediction error for that sample by the existing ensemble.

Stacking attempts to deduce the biases of the predictor(s) with respect to the learning set.[28] This deduction proceeds by generalizing in a second space using as input the predictions of the original predictors when taught with leave-one-out cross-validated partitions of the learning set and as output the correct predictions. More specifically, the method attempts to minimize the function

$$\sum_{i=1}^{N}[y_i - \sum_{b=1}^{B} c_b \phi_b^{(T-T_i)}(\boldsymbol{x}_i)] \tag{3}$$

where $\phi_b^{(T-T_i)}(\mathbf{x}_i)$ represents the leave-one-out cross-validated fit for $\phi_b(\boldsymbol{x})$ evaluated at $\boldsymbol{x} = \boldsymbol{x}_i$. This process produces estimates for the coefficients $\hat{c}_1, \hat{c}_2, ..., \hat{c}_B$, which are used to construct the ensemble prediction

$$\phi_{\text{bag}}(x) = \sum_{b=1}^{B}\hat{c}_b\phi_b(\boldsymbol{x}) \tag{4}$$

Unlike conventional cross-validation, stacking evaluates a linear combination of models rather than selecting the one with the smallest generalization error.

Theoretical analyses of ensemble techniques have focused on the bias-variance decomposition of learning error.[33,34] Under this formalism, the prediction error of a learning algorithm on a particular target function and training set size can be described as the sum of three non-negative terms: (1) a *bias* term, which measures how closely the learning algorithm's average prediction (over all possible training sets of the given training set size) matches the target, (2) a *variance* term, which measures how much the learning algorithm's predictions will vary with respect to each other, and (3) the *intrinsic target noise*, which represents the lower bound of the expected error of any learning algorithm as dictated by Bayes optimality. Although Breiman originally argued that both bagging and boosting work by reducing the variance term,[35] Freund and Schapire[27] suggested that boosting also attempts to reduce the bias component since it focuses on ill-predicted examples. This was later confirmed by Bauer and Kohavi,[36] who provided evidence that not only boosting but also bagging as well can reduce the bias component of the error. Still, it is widely accepted that both of these methods work primarily by reducing variance.

In addition to the insight that it offers in understanding the sources of error, a significant advantage of the bias-variance decomposition is the existence of the *bias-variance tradeoff*. Usually, changing one aspect of a learning algorithm will have opposite effects on the bias and variance. For example, as one increases the flexibility of a predictor (e.g. by increasing the number of synapses in a neural network), the bias decreases but the variance increases. The optimal number of degrees of freedom is the one that balances the tradeoff between bias and variance. Unfortunately, the practical use of this theory in real-world applications is limited by the fact that the function being learned is not known a priori. In such cases, these quantities must be estimated from the available sample by e.g. holding out some of the data,[36] resulting in a significant reduction in the amount of data available for training.

Although boosting and stacking were shown to outperform bagging on some data sets,[37] bagging is more stable in the sense that it will always perform at least as well as an individual predictor, as long as the predictor is unstable.[26] Moreover, the method provides a simple mechanism for computing confidence intervals[38] and is programmatically trivial to implement. In this work, we provide a systematic

study of bagging in the context of structure−activity correlation and compare it to two alternative methods for producing neural network ensembles using three classical data sets from the QSAR literature. Although aggregation does offer clear advantages, we find that bagging is not the best possible choice and that simpler techniques such as retraining with the full sample can often produce superior results. Of course, ensemble techniques are not the only method designed to address instability. An alternative strategy that is gaining momentum is to employ regularization theory.[39] Regularization attempts to convert an ill-posed approximation problem to a well-posed one, by introducing an auxiliary nonnegative functional that embeds smoothness constraints on the input−output mapping. Regularization was introduced in the neural network community by Poggio and Girosi[40] and was recently adopted by Burden and Winkler for exploring structure−activity correlations.[41]

II. METHODS

Network Modeling and Cross-Validation. Our analysis was based on three-layer fully connected multilayer perceptrons (MLPs), trained with the standard error back-propagation algorithm.[42] The logistic transfer function $f(x) = 1/(1 + e^{-x})$ was used for both hidden and output layers. Each network was trained for 300 epochs, using a linearly decreasing learning rate from 1.0 to 0.01 and a momentum of 0.8. During each epoch, the training patterns were presented to the network in a randomized order. In all cases, the descriptor data were normalized to [0,1] prior to network modeling.

Following common practice, the quality of the resulting models was assessed using *n*-fold cross-validation and quantified using the cross-validated correlation coefficient, R_{CV}

$$R_{CV} = \frac{N\sum y_i \tilde{y}_i - \sum y_i \sum \tilde{y}_i}{\sqrt{[N\sum y_i^2 - (\sum y_i)^2][N\sum \tilde{y}_i^2 - (\sum \tilde{y}_i)^2]}} \quad (5)$$

where N is the number of training patterns, and y_i and $\tilde{y}_i$ are the measured and predicted activities of the ith compound, respectively. The latter was obtained by dividing the training data into disjoint groups comprised on n patterns each, systematically removing each group from the training set, building a model with the remaining cases, and predicting the activity of the removed patterns using the optimized weights. This was done for each group of patterns in the original training set, and the resulting predictions were compared to the measured activities to determine their degree of correlation. Since cross-validation itself is known to be susceptible to the choice of initial parameters, each model was cross-validated 50 times in order to obtain reliable statistics and establish the true generalization capabilities of the resulting models. The same cross-validation procedure was used for both individual networks and network ensembles.

Aggregation. The most critical step in generating a neural network ensemble is the construction of the individual predictors prior to aggregation. In this work, we examine three different methods for constructing these predictors: (1) retraining with a bootstrapped resample, (2) retraining with the full sample, and (3) retraining with a partial sample, i.e., a subset of the original training data.

Table 1. Data Set Size and Neural Network Topology Used

data set	N[a]	M[b]	K[c]	H[d]	ref
AMA	31	53	3	3	14
BZ	57	42	6	2	44
PYR1	74	27	6	2	45
PYR2	68	38	6		46

[a] Number of samples. [b] Number of features in the original data set. [c] Number of features used in the models. [d] Number of hidden neurons.

Bootstrap aggregation or bagging[26] represents the most popular technique for constructing predictor ensembles. Bagging constructs the ensemble by training each predictor on a random redistribution of the training set. This approach is rooted on the bootstrap, a common statistical method for estimating the variability of a statistic obtained from a finite sample of data. If the real universe from which the samples are drawn is unknown or inaccessible, one needs to construct a proxy universe that embodies everything that is known about the real universe and which can be used to draw samples from. One such resampling technique is to replicate the sample data a large number of times and create a proxy universe based entirely on the available samples. An indirect and more effective way to accomplish this is to sample with replacement from the original sample. According to this method, each training pattern is selected with a probability $1/N$ each time, where N is the total number of patterns in the original sample. Many of the original examples may be repeated in the resampled training set, while others may be left out. The method constructs a proxy universe equal in size to the original sample and is, in effect, equivalent to sampling without replacement from an infinitely large replicated universe.

The other two methods for constructing ensembles involve retraining with the full sample and retraining with a partial sample, i.e., a subset of the original training set. The former generates multiple instances of the predictor by simply retraining the neural network using a different initial set of synaptic parameters, capitalizing on the susceptibility of the back-propagation algorithm to the presence of multiple local minima. The latter introduces diversity by training each neural network with a randomly chosen subset of the training patterns, in a manner similar to cross-validation. In this work, we used random subsets containing 90% of the original training patterns. As we discuss below, this number was chosen to be somewhere between the 100% coverage of simple retraining and the 63.2% (on average) coverage of bootstrapping.

Data Sets. The methods were tested on four well-studied data sets: antifilarial activity of antimycin analogues (AMA),[43] binding affinities of ligands to benzodiazepine/$GABA_A$ receptors (BZ),[44] and two sets of pyrimidine inhibitors of dihydrofolate reductase (PYR1[45] and PYR2[46]). These data sets have been the subject of extensive QSAR studies and have served as a test bed for many feature selection algorithms. To allow comparison with previous neural network-based approaches, the number of input features and hidden neurons were taken from the literature. These details are summarized in Table 1. The parameters have been specifically chosen to minimize over-fitting by maintaining

Table 2. Top Models Selected by the Artificial Ant Algorithm[a]

data set	selected variables (names)
AMA	NSDL8, MOFI_Y, LOGP, NSDL8, MOFI_Z, LOGP, MOFI_Y, LOGP, SUM_F
BZ	μ_7, π_7, σ_{m7}, MR_1, R_1, μ_2, μ_7, π_7, F_7, MR_1, σ_{p1}, σ_{m2}, μ_7, π_7, F_7, MR_1, σ_{m2}, π_6
PYR1	SZ_3, FL_3, Hd_3, ΠA_3, SZ_5, HA_5, Hd_3, ΠA_3, FL_4, PO_4, SZ_5, HA_5, SZ_3, FL_3, ΠA_3, SZ_5, Hd_5, HA_5
PYR2[a]	MW, VOL, MW_3, P_CH_2, P_CH_4, P_CH_SUM

[a] See ref 23. [b] Features selected by the ANFIS system described in ref 46.

a favorable ratio between the number of training cases and the number of freely adjustable parameters in the model.[2] To ensure that the results do not depend on a particular choice of descriptors, we examined three different models for each data set, comprised of the subsets of features that produced the highest cross-validation scores in a related study of a novel feature selection algorithm based on artificial ants[23] (Table 2).

Implementation. All programs were implemented in the C++ programming language and are part of the Directed-Diversity[47] software suite. They are based on 3-Dimensional Pharmaceuticals' Mt++ class library[48] and are designed to run on all Posix-compliant Unix and Windows platforms. All calculations were carried out on a Dell Inspiron 8000 laptop computer equipped with a 1 GHz Pentium IV Intel processor running Windows 2000 Professional.

III. RESULTS AND DISCUSSION

Our main goal was to determine whether bagging offered any significant advantages compared to the two alternative retraining schemes. Five factors that could affect the results were examined: (1) the nature of the chemical compounds and the biological assay, (2) the nature of the input descriptors, (3) the flexibility of the model, (4) the size of the ensemble, and (5) the nature of the cross-validation procedure. The results for the AMA, BZ, PYR1, and PYR2 data sets are summarized in Tables 3−6, respectively. These tables list the mean and standard deviation of the cross-validated correlation coefficients, R_{CV}, obtained from 50 independent *n*-fold cross-validation runs of the three best models discovered by the artificial ant feature selection algorithm described in ref 23. Though this procedure was computationally intensive, it was necessary in order to ensure that the results would reflect the true generalization ability of the models and would not be an artifact of a particular arbitrary shuffling of the training data. Each table is divided into three sets of rows marked as F, P, and B, which stand for retraining with the full sample, a partial sample, and a bootstrapped resample, respectively.

To ensure that our interpretation of these results accounted for the variability in the observed data, the effects of model, number of hidden units and ensemble size on the cross-validated correlation coefficient, R_{CV}, were examined using single-factor ANOVA (analysis of variance), ANOVA is a well-established statistical technique that attempts to test the null hypothesis that two or more means are equal, by comparing the variation observed within each category against the one observed across the entire sample. The key parameters of an ANOVA study include the *F* statistic which is compared to the critical value of the *F* distribution, F_{crit}, and the *P*-value which represents the smallest level of significance for which the observed sample information becomes significant, provided the null hypothesis is true. The results are summarized in Tables 7−9. Inspection of these results leads to several interesting observations.

(1) Regardless of the method used to construct the base models, the average generalization error of the network ensembles is always lower than that of the individual predictors (lower generalization error is manifested by a higher R_{CV}), and the same is true for the standard deviation. This means that ensembles are more stable and less sensitive to the choice of the validation set, which is consistent with the results reported by other authors.

(2) For ensembles constructed by retraining with the full sample (rows labeled *F* in Tables 3−6), the improvement of aggregation is consistent but relatively small (no more than 0.01 on the R_{CV} scale). The same is true for models constructed with partial samples (P).

(3) In most cases, there is no statistically significant difference in generalization performance between ensembles trained with full and partial samples (see column labeled F−P in Table 9). The only exceptions were the TOP3 model in PYR1 where the full sample ensemble was better than the partial sample one by 0.015 R_{CV} units, and, to a lesser extent, the TOP1 model of PYR2 where the difference was marginally significant. Conversely, significant differences between bootstrapped and nonbootstrapped ensembles were observed for three sets of models (AMA TOP2, PYR1 TOP1, and PYR1 TOP3) and to a lesser extent for AMA TOP2, BZ TOP1, BZ TOP3, and PYR2 TOP1. The only case where bootstrapped ensembles performed better than nonbootstrapped ones was the AMA TOP2 model, but the difference was relatively modest (0.003 R_{CV} units). In all other cases, nonbootstrapped ensembles were favored, sometimes by a wide margin (as high as 0.05 on the R_{CV} scale).

(4) The gains from aggregation are most impressive for individuals trained with bootstrap resamples (B), sometimes reaching as high as 0.04 on the R_{CV} scale. This difference is to a large extent due to the drop in the generalization performance of the respective base networks, which for the PYR1 data set can degrade by as much as 0.08 R_{CV} units. Thus, whatever advantages are afforded by aggregation are lost due to the deterioration of the underlying models. Conversely, the minimal gains afforded by simple retraining with full or partial samples reflect the substantially lower diversity of the constituent models, as manifested by the variances of their generalization errors.

(5) The features that are employed by the model can have a significant impact on the effectiveness of bootstrapping as a means of constructing ensembles. This is most evident in the PYR1 data set, where the difference between bootstrap resampling and full sampling for the first and third model (TOP1 and TOP3, respectively) differ by ~0.04 R_{CV} units, whereas for the second model (TOP2) it is statistically insignificant. In all other cases, the effects are similar across the various models.

(6) The choice of the cross-validation methodology has no qualitative effect on the relative merits of bootstrapping versus the full sampling and partial sampling procedures. In all cases, 1-fold and 10-fold cross-validation produced similar results for the top model. Not surprisingly, the former

Table 3. Cross-Validated *R* for Individual Networks and Network Ensembles for the AMA Data Set

				TOP1[a]				TOP2[b]				TOP3[c]			
				net		bag		net		bag		net		bag	
method[d]	models[e]	hidden[f]	CV[g]	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$
B	10	3	1	0.7642	0.0396	0.801	0.013								
B	10	3	3	0.7517	0.0513	0.7901	0.0217	0.7909	0.035	0.8181	0.014	0.7804	0.0404	0.8187	0.0144
B	10	5	3	0.7598	0.0454	0.7971	0.016								
B	10	7	3	0.753	0.0496	0.7913	0.0184								
B	15	3	3	0.7547	0.0518	0.797	0.0171								
B	5	3	3	0.7616	0.0457	0.7947	0.0205								
F	10	3	1	0.792	0.0104	0.7954	0.003								
F	10	3	3	0.7896	0.0168	0.7932	0.013	0.8078	0.0121	0.809	0.0105	0.8142	0.0155	0.8205	0.0092
F	10	5	3	0.7787	0.0183	0.7813	0.0157								
F	10	7	3	0.7745	0.0186	0.7766	0.0165								
F	15	3	3	0.7899	0.018	0.7934	0.0147								
F	5	3	3	0.7873	0.0162	0.7904	0.0134								
P	10	3	1	0.7847	0.0177	0.7928	0.0049								
P	10	3	3	0.7793	0.0237	0.7879	0.0141	0.8003	0.015	0.8052	0.0096	0.8101	0.0199	0.814	0.0113
P	10	5	3	0.7705	0.0243	0.7784	0.0161								
P	10	7	3	0.7646	0.0238	0.7729	0.013								
P	15	3	3	0.7739	0.0277	0.7834	0.0175								
p	5	3	3	0.7757	0.0264	0.7839	0.0162								

[a] Models derived from features NSDL8, MOFI_Y, and LOGP. [b] Models derived from features NSDL8, MOFI_Z, and LOGP. [c] Models derived from features MOFI_Y, LOGP, and SUM_F. [d] Method used to construct the ensemble (B: bootstrapping; F: full sample; P: partial sample). [e] Number of models in the ensemble. [f] Number of hidden neurons. [g] Size of cross-validation set (*n* in leave-*n*-out cross-validation). [h] Mean cross-validated *R* of individual networks over 50 cross-validation runs. [i] Standard deviation of cross-validated *R* of individual networks over 50 cross-validation runs. [j] Mean cross-validated *R* of network ensembles over 50 cross-validation runs. [k] Standard deviation of cross-validated *R* of network ensembles over 50 cross-validation runs.

Table 4. Cross-Validated *R* for Individual Networks and Network Ensembles for the BZ Data Set

				TOP1[a]				TOP2[b]				TOP3[c]			
				net		bag		net		bag		net		bag	
method[d]	models[e]	hidden[f]	CV[g]	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$
B	10	2	1	0.8606	0.0237	0.8945	0.0057								
B	10	2	5	0.8556	0.0265	0.8901	0.01	0.8521	0.0303	0.892	0.0093	0.848	0.03	0.8883	0.0108
B	10	4	5	0.8687	0.0255	0.9032	0.0093								
B	10	6	5	0.8754	0.0244	0.9069	0.0078								
B	15	2	5	0.8567	0.0295	0.8951	0.0093								
B	5	2	5	0.8557	0.03	0.8886	0.0093								
F	10	2	1	0.8896	0.0089	0.8982	0.0032								
F	10	2	5	0.8863	0.0122	0.8948	0.0086	0.8836	0.0135	0.8934	0.0089	0.8848	0.0137	0.8946	0.0098
F	10	4	5	0.8987	0.0107	0.9069	0.0064								
F	10	6	5	0.9037	0.01	0.9095	0.0066								
F	15	2	5	0.8873	0.0125	0.8961	0.0085								
F	5	2	5	0.8864	0.0124	0.8937	0.0095								
P	10	2	1	0.8869	0.0105	0.8981	0.0035								
P	10	2	5	0.8829	0.0133	0.8948	0.0074	0.8784	0.0155	0.892	0.0095	0.8795	0.0158	0.8934	0.0078
P	10	4	5	0.894	0.014	0.9062	0.0078								
P	10	6	5	0.899	0.0121	0.9085	0.0078								
P	15	2	5	0.886	0.0129	0.8978	0.0074								
P	5	2	5	0.8853	0.0135	0.8956	0.0087								

[a] Models derived from features μ_7, π_7, σ_{m7}, MR_1, R_1, and μ_2. [b] Models derived from features μ_7, π_7, F_7, MR_1, σ_{p1}, and σ_{m2}. [c] Models derived from features μ_7, π_7, F_7, MR_1, σ_{m2}, and π_6. [d] Method used to construct the ensemble (B: bootstrapping; F: full sample; P: partial sample). [e] Number of models in the ensemble. [f] Number of hidden neurons. [g] Size of cross-validation set (*n* in leave-*n*-out cross-validation). [h] Mean cross-validated *R* of individual networks over 50 cross-validation runs. [i] Standard deviation of cross-validated *R* of individual networks over 50 cross-validation runs. [j] Mean cross-validated *R* of network ensembles over 50 cross-validation runs. [k] Standard deviation of cross-validated *R* of network ensembles over 50 cross-validation runs.

produced slightly lower generalization error estimates than the latter, for both the individual networks and their ensembles. To minimize the computational burden associated with our intense validation procedure, all subsequent studies were restricted to 10-fold cross-validation.

(7) The ANOVA results suggest that there is a very strong interaction between the generalization performance and the number of hidden units for the nonbootstrapped samples (F and P, respectively). While the effect is statistically very significant, the direction and magnitude depends on the particular data set and the extent of over-fitting induced by the increased flexibility of the underlying models. For the AMA data set, increasing the number of hidden nodes decreases generalization performance, whereas the opposite is observed for BZ and PYR1. In the case of bootstrapped resamples, generalization performance increases with increasing number of hidden units for BZ and PYR1 but remains statistically unchanged for AMA. These results seem

Table 5. Cross-Validated *R* for Individual Networks and Network Ensembles for the PYR1 Data Set

				TOP1[a]				TOP2[b]				TOP3[c]			
				net		bag		net		bag		net		bag	
method[d]	models[e]	hidden[f]	CV[g]	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$	$\mu(R_{CV})^j$	$\sigma(R_{CV})^k$
B	10	2	1	0.7244	0.055	0.7646	0.0152								
B	10	2	7	0.7164	0.0592	0.7561	0.0236	0.7259	0.0472	0.7713	0.0138	0.6786	0.0768	0.7305	0.0289
B	10	4	7	0.7408	0.0495	0.779	0.0167								
B	10	6	7	0.7421	0.0501	0.7776	0.0134								
B	15	2	7	0.7185	0.0586	0.7596	0.0216								
B	5	2	7	0.719	0.0588	0.7559	0.0232								
F	10	2	1	0.7943	0.0191	0.8009	0.0062								
F	10	2	7	0.7955	0.0247	0.8024	0.0128	0.7633	0.0304	0.7745	0.0184	0.7602	0.0473	0.7735	0.0171
F	10	4	7	0.8077	0.0223	0.8136	0.0115								
F	10	6	7	0.8164	0.0206	0.8208	0.0129								
F	15	2	7	0.7943	0.0283	0.8017	0.0137								
F	5	2	7	0.7945	0.0263	0.8004	0.0156								
P	10	2	1	0.7937	0.0281	0.806	0.0087								
P	10	2	7	0.79	0.0354	0.8037	0.0149	0.7571	0.0324	0.7734	0.0179	0.7374	0.0578	0.7586	0.0235
P	10	4	7	0.8023	0.0288	0.8142	0.0134								
P	10	6	7	0.8111	0.031	0.8213	0.0164								
P	15	2	7	0.7912	0.0372	0.8045	0.0166								
P	5	2	7	0.7944	0.0277	0.8035	0.0149								

[a] Models derived from features SZ_3, FL_3, Hd_3, ΠA_3, SZ_5, and HA_5. [b] Models derived from features Hd_3, ΠA_3, FL_4, PO_4, SZ_5, and HA_5. [c] Models derived from features SZ_3, FL_3, ΠA_3, SZ_5, Hd_5, and HA_5. [d] Method used to construct the ensemble (B: bootstrapping; F: full sample; P: partial sample). [e] Number of models in the ensemble. [f] Number of hidden neurons. [g] Size of cross-validation set (*n* in leave-*n*-out cross-validation). [h] Mean cross-validated *R* of individual networks over 50 cross-validation runs. [i] Standard deviation of cross-validated *R* of individual networks over 50 cross-validation runs. [j] Mean cross-validated *R* of network ensembles over 50 cross-validation runs. [k] Standard deviation of cross-validated *R* of network ensembles over 50 cross-validation runs.

Table 6. Cross-Validated *R* for Individual Networks and Network Ensembles for the PYR2 Data Set

				TOP1[a]			
				net		bag	
method[b]	models[c]	hidden[d]	CV[e]	$\mu(R_{CV})^f$	$\sigma(R_{CV})^g$	$\mu(R_{CV})^h$	$\sigma(R_{CV})^i$
B	10	10	1	0.7655	0.0301	0.8171	0.01
B	10	10	7	0.7565	0.0333	0.809	0.0136
B	15	10	7	0.7605	0.0329	0.8162	0.014
B	5	10	7	0.76	0.0305	0.8057	0.016
F	10	10	1	0.8123	0.0092	0.8213	0.0035
F	10	10	7	0.8073	0.0149	0.8158	0.0112
F	15	10	7	0.8044	0.0191	0.8131	0.016
F	5	10	7	0.809	0.0151	0.8163	0.0126
P	10	10	1	0.8042	0.0131	0.8179	0.0046
P	10	10	7	0.7971	0.0176	0.8108	0.0107
P	15	10	7	0.7995	0.0191	0.8141	0.0122
P	5	10	7	0.7981	0.0198	0.8104	0.0144

[a] Models derived from features MW, VOL, MW_3, P_CH_2, P_CH_4, and P_CH_SUM. [b] Method used to construct the ensemble (B: bootstrapping; F: full sample; P: partial sample). [c] Number of models in the ensemble. [d] Number of hidden neurons. [e] Size of cross-validation set (*n* in leave-*n*-out cross-validation). [f] Mean cross-validated *R* of individual networks over 50 cross-validation runs. [g] Standard deviation of cross-validated *R* of individual networks over 50 cross-validation runs. [h] Mean cross-validated *R* of network ensembles over 50 cross-validation runs. [i] Standard deviation of cross-validated *R* of network ensembles over 50 cross-validation runs.

to suggest that for the AMA data set the architecture used in the original study was prudently chosen. Conversely, for BZ and PYR1 additional hidden units lead to improved generalization performance both at the individual and the ensemble levels, though the difference is relatively small.

(8) Finally, the size of the ensemble does not appear to have a significant impact on its performance, at least for the range examined in this work. The differences between 5, 10, and 15 models were in most cases statistically insignificant, or barely significant but marginal in magnitude, suggesting that ~10 models would be sufficient for good generalization. A weak interaction was observed in the BZ and PYR2 bootstrapped ensembles, which is probably due to the relatively greater diversity of the underlying models. We need to point out, however, that the ensemble sizes examined in this study are smaller that those typically used by other researchers, particularly in connection with classification and regression trees. This reflects practical considerations and was dictated by the computational requirements of neural networks and the intensive nature of our cross-validation procedure. Although we cannot draw any definitive conclusions, it appears doubtful that larger ensembles can lead to any noticeable improvements in generalization performance. This finding is consistent with previous empirical evidence.[49]

These results can be easily rationalized using Krogh and Vedelsby's[50] theoretical framework for analyzing neural network ensembles. Let us define the *ambiguity* of a single member of the ensemble on a prediction for a pattern $(\boldsymbol{x}, y)$ as

$$\alpha_b(\boldsymbol{x}) = (\phi_b(\boldsymbol{x}) - \phi_{\text{bag}}(\boldsymbol{x}))^2 \tag{6}$$

and the *generalization error* as

$$e_b(\boldsymbol{x}) = (\phi_b(\boldsymbol{x}) - y)^2 \tag{7}$$

Similarly, we can define the *ensemble ambiguity* as

$$\alpha_{bag}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}(\phi_b(\boldsymbol{x}) - \phi_{bag}(\boldsymbol{x}))^2 \tag{8}$$

and the *ensemble generalization error* as

$$e_{\text{bag}}(\boldsymbol{x}) = (\phi_{\text{bag}}(\boldsymbol{x}) - y)^2 \tag{9}$$

Table 7. ANOVA Analysis (Fixed Main Effects Model at the 0.05 Significance Level) of the Effects of the Method Used To Construct the Ensemble on the Generalization Ability, R_{CV}

		B−F[a]				B−P[b]				F−P[c]			
data	model[d]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]
AMA	TOP1	0.709	3.938	0.402		0.362	3.938	0.549		3.681	3.938	0.058	
	TOP2	13.073	3.938	5.0E−04	**	28.411	3.938	6.3E−07	***	3.682	3.938	0.058	
	TOP3	0.525	3.938	0.471		1.002	3.938	0.319		0.167	3.938	0.684	
BZ	TOP1	6.524	3.938	0.012	*	7.184	3.938	0.009	*	0.001	3.938	0.971	
	TOP2	0.609	3.938	0.431		0.001	3.938	0.976		0.543	3.938	0.463	
	TOP3	9.087	3.938	0.003	*	7.061	3.938	0.009	*	0.464	3.938	0.497	
PYR1	TOP1	145.653	3.938	4.3E−21	***	142.504	3.938	8.2E−21	***	0.219	3.938	0.641	
	TOP2	0.976	3.938	0.326		0.426	3.938	0.516		0.096	3.938	0.757	
	TOP3	80.423	3.938	2.1E−14	***	27.914	3.938	7.7E−07	***	12.935	3.938	5.0E−04	**
PYR2	TOP1	7.236	3.938	8.0E−03	*	0.571	3.938	0.452		4.891	3.938	0.029	*

[a] Comparison of methods B (bootstrapping) and F (full sample). [b] Comparison of methods B (bootstrapping) and P (partial sample). [c] Comparison of methods F (full sample) and P (partial sample). [d] Model examined (for description, see Tables 3−6). [e] Flag indicating whether the difference between the means is statistically significant (numbers of stars indicate the relative strength of the statistical significance).

Table 8. ANOVA Analysis (Fixed Main Effects Model at the 0.05 Significance Level) of the Effects of the Number of Hidden Units on the Generalization Ability, R_{CV}

		B[a]				F[b]				P[c]			
data	model[d]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]
AMA	TOP1	1.913	3.058	0.151		15.573	3.058	7.3E−07	***	13.605	3.058	3.8E−06	***
BZ	TOP1	46.678	3.058	2.0E−16	***	58.818	3.058	5.3E−19	***	44.908	3.058	6.0E−16	***
PYR1	TOP1	23.86	3.058	1.1E−09	***	27.536	3.058	7.0E−11	***	17.23	3.058	1.9E−07	***

[a] Bootstrapping method. [b] Full sample method. [c] Partial sample method. [d] Model examined (for description, see Tables 3−6). [e] Flag indicating whether the difference between the means is statistically significant (numbers of stars indicate the relative strength of the statistical significance).

Table 9. ANOVA Analysis (Fixed Main Effects Model at the 0.05 Significance Level) of the Effects of Ensemble Size on the Generalization Ability, R_{CV}

		B[a]				F[b]				P[c]			
data	model[d]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]	*F*	F_{crit}	*P*-value	signif[e]
AMA	TOP1	1.539	3.058	0.218		0.741	3.058	0.478		1.162	3.058	0.316	
BZ	TOP1	6.301	3.058x	0.002	*	0.894	3.058	0.411		1.912	3.058	0.151	
PYR1	TOP1	0.404	3.058	0.670		0.24	3.058	0.787		0.06	3.058	0.941	
PYR2	TOP1	6.602	3.058	0.002	*	0.805	3.058	0.449		1.312	3.058	0.272	

[a] Bootstrapping method. [b] Full sample method. [c] Partial sample method. [d] Model examined (for description, see Tables 3−6). [e] Flag indicating whether the difference between the means is statistically significant (numbers of stars indicate the relative strength of the statistical significance).

The ensemble ambiguity, $\alpha_{bag}(\boldsymbol{x})$, measures the diversity of the ensemble, i.e., the extent to which the models in the ensemble disagree on their predictions for a single pattern $(\boldsymbol{x}, y)$. If we average over the entire test set, we obtain

$$\begin{aligned}\bar{\epsilon}(\boldsymbol{x}) &= \frac{1}{B}\sum_{b=1}^{B}(\phi_b(\boldsymbol{x}) - y)^2 \\ &= \frac{1}{B}\sum_{b=1}^{B}(\phi_b(\boldsymbol{x}) - \phi_{bag}(\boldsymbol{x}) + \phi_{bag}(\boldsymbol{x}) - y)^2 \\ &= \frac{1}{B}\sum_{b=1}^{B}(\phi_b(\boldsymbol{x}) - \phi_{bag}(\boldsymbol{x}))^2 + (\phi_{bag}(\boldsymbol{x}) - y)^2 \quad (10)\end{aligned}$$

which, using eqs 8 and 9, can be rewritten as

$$\bar{\epsilon}(\boldsymbol{x}) = \alpha_{bag}(\boldsymbol{x}) - \epsilon_{bag}(\boldsymbol{x}) \quad (11)$$

or

$$\epsilon_{bag}(\boldsymbol{x}) = \bar{\epsilon}(\boldsymbol{x}) - \alpha_{bag}(\boldsymbol{x}) \quad (12)$$

If we now define the average ambiguity of the *b*th predictor over the entire distribution *P* from which the samples are drawn as

$$A_b = \int dx P(\boldsymbol{x})\alpha_b(\boldsymbol{x}) \quad (13)$$

and the generalization error as

$$E_b = \int dx P(\boldsymbol{x}) e_b(\boldsymbol{x}) \quad (14)$$

and if we denote the average ambiguity and generalization error across the entire ensemble as *A* and *E*, respectively, then we can define the average ensemble generalization error over *P* as

$$E = \int dx P(\boldsymbol{x}) e_{bag}(\boldsymbol{x}) \quad (15)$$

and by virtue of eqs 11 and 14

$$E = \bar{E} - \bar{A} \quad (16)$$

Equation 16 is a valuable expression that relates the average ensemble generalization error to the diversity of the ensemble. According to this expression, an ideal ensemble

consists of highly accurate predictors that disagree as much as possible. Indeed, eq 16 asserts that increasing diversity ($\bar{A}$) will improve the generalization performance of the ensemble *as long as* the average generalization performance of the individual predictors ($\bar{E}$) is not compromised. Our simulations demonstrate that while bootstrap ensembles are the ones that benefit the most from aggregation, these gains are not sufficient to offset the loss of predictivity of the base predictors. Indeed, the probability that an individual training case from T will not be selected as part of a bootstrapped training set is $(1-1/N)^N \approx 0.368$, where N is the number of training samples in T. This means that a typical bootstrap resampled training set contains on average only 63.2% of the samples in the original training set. While this increases the diversity of the resulting models, it also decreases their generalization performance since the predictions are now based on a much smaller subset of the training data. Our results suggest that these two components are tightly connected: the more individual predictivity is sacrificed, the greater the diversity of the models and the gains from aggregation. Indeed, the only case where bootstrapping produced a superior ensemble was the TOP2 model in the AMA data set (Table 3), where the loss of individual predictivity and gains from aggregation were minimal.

One final cautionary note. The ensembles used in this work were based on models that were designed to resist overfitting. As a consequence, the individual predictors are much more stable than those derived from e.g. typical classification and regression trees (i.e. they have higher bias and lower variance), a fact that is reflected in the variances listed in Tables 3−6. It has long been recognized that bagging is most beneficial when the predictors are inherently unstable. Should therefore one aim at creating unstable models hoping to recover through aggregation? Although the results presented here do not definitively answer this question, they suggest that this may not be the best possible strategy.

IV. CONCLUSIONS

The results presented herein cast doubts on the usefulness of bootstrapping in creating neural network ensembles for QSAR and QSPR. Although this method is becoming increasingly popular in the machine learning community, simpler methods such as retraining with the full sample seem to be much more effective. Our results suggest that while ensembles clearly and consistently outperform individual predictors, the results are not as overwhelming as originally thought, particularly when the predictors are designed to resist over-fitting. To maximize the gain from aggregation, the base models need to be not only highly predictive but also maximally diverse. Unfortunately, it turns out that diversity is not easy to achieve, because all models are trained to do essentially similar tasks. Nevertheless, ensemble modeling can improve generalization performance and should be more widely adopted by the QSAR community since the cost does not appear to be prohibitively high. Of course, as with most empirical studies of this kind, these results should be viewed with caution. It is possible that our conclusions may reflect the idiosyncrasies of these particular data sets and may not extend to other types of biological data. Although the present work adds to a substantial body of existing empirical evidence in favor of ensemble techniques, the reader should remember that the theory of how, why, and when these methods work is still in its infancy.

ACKNOWLEDGMENT

The authors would like to thank Dr. Raymond F. Salemme of 3-Dimensional Pharmaceuticals, Inc. for his insightful comments and support of this work.

REFERENCES AND NOTES

(1) Aoyama, T.; Suzuki, Y.; Ichikawa, H. Neural networks applied to structure−activity relationships. *J. Med. Chem*. **1990**, *33*, 905−908.
(2) Andrea, T. A.; Kalayeh, H. Application of neural networks in quantitative structure−activity relationships of dihydrofolate reductase inhibitors. *J. Med. Chem*. **1991**, *34*, 2824−2836.
(3) So, S.-S.; Richards, W. G. Application of neural networks: quantitative structure−activity relationships of the derivatives of 2,4-diamino-5-(substituted-benzyl) pyrimidines as DHFR inhibitors. *J. Med. Chem*. **1992**, *35*, 3201−3207.
(4) Ajay, A unified framework for Using neural networks to build QSARs. *J. Med. Chem*. **1993**, *36*, 3565−3671.
(5) Wikel, J. H.; Dow, E. R. The use of neural networks for variable selection in QSAR. *Bioorg. Med. Chem. Lett*. **1993**, *3*, 645−651.
(6) Gasteiger, J.; Zupan, J. Neural networks in chemistry. *Angew. Chem., Int. Ed. Engl*. **1993**, *105(4)*, 503−527.
(7) Zupan, J.; Gasteiger, J. *Neural Networks for Chemists: An Introduction*; VCH: Weinheim, 1993.
(8) Burns, J. A.; Whitesides, G. M. Feed-forward neural networks in chemistry: mathematical systems for classification and pattern recognition. *Chem. Rev*. **1993**, *93*, 2583.
(9) Manallak, D. T.; Ellis, D. D.; Livingston, D. J. Analysis of linear and nonlinear QSAR data using neural networks. *J. Med. Chem*. **1994**, *37*, 3758−3767.
(10) *Neural Networks in QSAR and Drug Design*; Devillers, J., Ed.; Academic Press: New York, 1996.
(11) So, S.-S.; Karplus, M. Evolutionary optimization in quantitative structure−activity relationship: an application of genetic neural networks. *J. Med. Chem*. **1996**, *39*, 1521−1530.
(12) Hansch, L.; Leo, C. *Exploring QSAR. Fundamentals and applications in chemistry and biology*; American Chemical Society: Washington, DC, 1996.
(13) Dunn, W. J.; Wold, S.; Edlund, U.; Hellberg, S. Multivariate structure−activity relationships between data from a battery of biological tests and an ensemble of structure descriptors: the PLS method. *QSAR* **1984**, *3*, 131−137.
(14) Selwood, D. L.; Livingstone, D. J.; Comley, J. C. W.; O'Dowd, A. B.; Hudson, A. T.; Jackson, P.; Jandu, K. S.; Rose, V. S.; Stables, J. N. *J. Med. Chem.* **1990**, *33*, 136−142.
(15) Burden, F. R.; Ford, M. G.; Whitley, D. C.; Winkler, D. A. Use of automatic relevance determination in QSAR studies using Bayesian neural networks. *J. Chem. Info. Comput. Sci*. **2000**, *40*, 1423−1430.
(16) Sutter, J. M.; Dixon, S. L.; Jurs, P. C. Automated descriptor selection for quantitative structure−activity relationships using generalized simulated annealing. *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 77−84.
(17) Luke, B. T. Evolutionary programming applied to the development of quantitative structure−activity relationships and quantitative structure−property relationships. *J. Chem. Inf. Comput. Sci*. **1994**, *34*, 1279−1287.
(18) Kubinyi, H. Variable selection in QSAR studies. I. An evolutionary algorithm. *QSAR* **1994**, *13*, 285−294.
(19) Rogers, D. R.; 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.
(20) Yasri, A.; Hartsough, D. Toward an optimal procedure for variable selection and QSAR model building. *J. Chem. Inf. Comput. Sci*. **2001**, *41*, 1218−1227.
(21) Hasegawa, K.; Miyashita, Y.; Funatsu, K. GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists. *J. Chem. Inf. Comput. Sci*. **1997**, *37*, 306−310.
(22) Izrailev, S.; Agrafiotis, D. K. 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.
(23) Izrailev, S.; Agrafiotis, D. K. Variable selection for QSAR by artificial ant colony systems. *SAR QSAR Environ. Res*. **2002**, *13*, 417−423.
(24) Agrafiotis, D. K.; Cedeno, W. Feature selection for structure−activity correlation using binary particle swarms. *J. Med. Chem.* **2002**, *45*, 1098−1107.
(25) Nilsson, N. J. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill: New York, 1965.

(26) Breiman, L. Bagging predictors. *Machine Learning* **1996**, *24*, 123−140.
(27) Feund, Y.; Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*; Springer-Verlag: 1995; pp 23−37.
(28) Wolpert, D. H. Stacked generalization. *Neural Networks* **1996**, *8*, 1341−1390.
(29) Breiman, L. Stacked regressions. *Machine Learning* **1996**, *24*, 49−64.
(30) Carney, J. G.; Cunningham, P. The NeuralBag algorihm: optimizing generalization performance in bagged neural networks. In *Proceedings of the 7th European Symposium on Neural Networks*; Verleysen, M., Ed.; Brussels, Belgium, 1997; pp 35−40.
(31) Heskes, T. Balancing between bagging and bumping. In *Advances in Neural Information Processing Systems*; Mozer, M., Jordan, M., Petsche, T., Eds.; MIT Press: 1997; Vol. 9, pp 176−182.
(32) Zhang, J. Developing robust nonlinear models through bootstrap aggregated neural networks. *Neurocomputing* **1999**, *25*, 93−113.
(33) Geman, S.; Bienenstock, E.; Doursat, R. Neural networks and bias/variance dilemma. *Neural Computation* **1992**, *4*, 1−58.
(34) Breiman, L. *Some infinity theory fr predictor ensembles*; Technical Report 577; Statistics Department, University of California: Berkeley, CA, 2000.
(35) Breiman, L. Bias, variance and arcing classifiers; Technical Report 460; Statistics Department, University of California: Berkeley, CA, 1996.
(36) Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning* **1999**, *36*, 105−139.
(37) Maclin, R.; Opitz, D. An empirical evaluation of bagging and boosting. In *Proceedings of the 14th National Conference on Artificial Intelligence*; Providence, RI, 1997.
(38) Carney, J. G.; Cunningham, P. Confidence and prediction intervals for neural network ensembles. In *Proceedings of the International Joint Conference on Neural Networks*; 1999.
(39) Tikhonov, A. N.; Arsenin, V. Y. *Solutions of ill-posed problems*; W. H. Winston: Washington, DC, 1977.
(40) Poggio, T.; Girosi, F. Networks for approximation and learning. *Proc. IEEE* **1990**, *78*, 1481−1497.
(41) Burden, F. R.; Winkler, D. A. Robust QSAR models using Bayesian regularized neural networks. *J. Med. Chem.* **1999**, *42*, 3183−3187.
(42) Haykin, S. *Neural networks*; Macmillan: New York, 1994.
(43) Selwood, D. L.; Livingstone, D. J.; Comley, J. C. W.; O'Dowd, A. B.; Hudson, A. T.; Jackson, P.; Jandu, K. S.; Rose, V. S.; Stables, J. N. *J. Med. Chem.* **1990**, *33*, 136−142.
(44) Maddalena, D. J.; Johnson, G. A. R. *J. Med. Chem.* **1995**, *38*, 715−724.
(45) Hirst, J. D.; King, R. D.; Sternberg, M. J. E. *J. Comput.-Aided Mol. Design* **1994**, *8*, 405−420.
(46) Loukas, Y. L. Adaptive neuro-fuzzy inference system: an instant and architecture-free predictor for improved QSAR studies. *J. Med. Chem.* **2001**, *44*, 2772−2783.
(47) Agrafiotis, D. K.; Bone, R. F.; Salemme, F. R.; Soll, R. M. United States Patents 5, 463, 564, **1995**; 5, 574, 656, **1996**; 5, 684, 711, **1997**; and 5, 901, 069, **1999**.
(48) Copyright 3-Dimensional Pharmaceuticals, Inc., 1994−2000.
(49) Opitz, D.; Maclin, R. Popular ensemble methods: an empirical study. *J. Artific. Intell. Res.* **1999**, *11*, 169−198.
(50) Krogh, A.; Vedelsby, J. Neural network ensembles, cross-validation and active learning. In *Advances in Neural Information Processing Systems 7*; Tesauro, G., Touretzky, D., Lean, T., Eds.; MIT Press: 1995; pp 231−238.

CI0203702