Artificial Neural Network–Based Analysis of High-Throughput Screening Data for Improved Prediction of Active Compounds

SWAPAN CHAKRABARTI,1 STAN R. SVOJANOVSKY,2 ROMANA SLAVIK,3 GUNDA I. GEORG,4 GEORGE S. WILSON,5 and PETER G. SMITH2,6,7

Artificial neural networks (ANNs) are trained using high-throughput screening (HTS) data to recover active compounds from a large data set. Improved classification performance was obtained on combining predictions made by multiple ANNs. The HTS data, acquired from a methionine aminopeptidases inhibition study, consisted of a library of 43,347 compounds, and the ratio of active to nonactive compounds, $R_{AN}$, was 0.0321. Back-propagation ANNs were trained and validated using principal components derived from the physicochemical features of the compounds. On selecting the training parameters carefully, an ANN recovers one-third of all active compounds from the validation set with a 3-fold gain in $R_{AN}$ value. Further gains in $R_{AN}$ values were obtained upon combining the predictions made by a number of ANNs. The generalization property of the back-propagation ANNs was used to train those ANNs with the same training samples, after being initialized with different sets of random weights. As a result, only 10% of all available compounds were needed for training and validation, and the rest of the data set was screened with more than a 10-fold gain of the original $R_{AN}$ value. Thus, ANNs trained with limited HTS data might become useful in recovering active compounds from large data sets. (Journal of Biomolecular Screening 2009:1236-1244)

Key words: pattern classification, neural networks, generalization property

INTRODUCTION

Over the past few decades, the experimental high-throughput screening (HTS) process has become the first major step toward drug discovery.1 It is essentially an expensive laboratory process that measures chemical and biological activities of a large number of molecules toward one or several target molecules. The compounds that are found active toward a target molecule are further investigated for discovering new drugs. The test set for HTS often consists of a few thousand to several hundred thousand compounds on high-density microplates, and automated robotic instruments are used to run the HTS assays.2 Due to ever increasing costs of laboratory experimentation with such a large number of samples that are mostly found to be nonactive toward a target molecule, alternative low-cost computer-assisted approaches are considered desirable. A support vector machine (SVM)–based approach has been used to analyze HTS data of a methionine aminopeptidases (MetAPs) inhibition study.3 This approach described an increased ratio of active to nonactive compounds (RA/N) by 7-fold in the classified set. This enhancement in the RA/N value was obtained when more than 10,000 examples, out of approximately 40,000 available samples, were used to train the classifier.

Among other computer-assisted approaches, the artificial neural network (ANN)–based schemes have gained significant momentum.2,4 A back-propagation ANN is well known for its inherent ability in learning to generalize from a small number of examples per class, and it often provides robust performances when the input data are corrupted with incomplete and/or noisy information.5,6 It has been noted that the methodologies that can tolerate noisy data might become useful in HTS data mining.1 This valuable information and previous successes of ANNs in classifying targets from their noisy responses have been the primary impetuses for using ANN in classifying active compounds.6,7 This article investigates the feasibility of training and validating ANNs using a relatively small experimental HTS data set and then recovering active compounds from a large test set.
The HTS data from a MetAPs inhibition study were made available by the HTS Laboratory at the University of Kansas.

A chemical library of 43,347 chemical compounds was examined by the experimental HTS process, and a normalized sorted data set was obtained in the order of decreasing activities toward cobalt. Then, 1347 molecules with inhibition activity greater than or equal to 40% toward cobalt were selected as active, and the remaining 42,000 compounds were considered nonactive. Thus, for the original data set, the ratio of active to nonactive compounds (\(R_{\text{AN}}\)) was 0.0321. Each compound of the given data set was characterized by a feature vector of 16 elements corresponding to the 16 physical and chemical properties of that compound. The overall measured inhibition activities toward cobalt for the compounds were normalized between 0 and 1.

The original data set is first divided into a development set and a test set, and then the development set is further divided into training and validation sets. ANNs are trained using the samples selected randomly from the training set and then validated using all the members of the validation set. It is important to note that \(R_{\text{AN}}\) values, obtained from classifying the validation set, actually provide a measure of the ratio of true positives to false positives. In addition, the \(R_{\text{AN}}\) value is found to increase when the predictions made by a large number of trained ANNs are combined. However, this increase is recurrently accompanied by a decrease in the actual number of recovered active compounds. To train those large numbers of ANNs without using numerous training samples, the generalization property of a back-propagation ANN is exploited.

The ANNs that provided superior performances during the validation procedure are then applied to analyze the test set. On combining the predictions made by these ANNs, nearly a 10-fold gain is obtained in \(R_{\text{AN}}\) values while recovering about one-sixth (or nearly 16%) of all the active compounds from the test set. We also demonstrate that during the training process, an ANN basically learns to provide a gain “G” in the \(R_{\text{AN}}\) values from the initial values available in the validation or even the test set. Thus, to get an estimate of the G value from the validation set, actually provide a measure of the ratio of true positives to false positives. In addition, the \(R_{\text{AN}}\) value is found to increase when the predictions made by a large number of trained ANNs are combined. However, this increase is recurrently accompanied by a decrease in the actual number of recovered active compounds. To train those large numbers of ANNs without using numerous training samples, the generalization property of a back-propagation ANN is exploited.

The ANNs that provided superior performances during the validation procedure are then applied to analyze the test set. On combining the predictions made by these ANNs, nearly a 10-fold gain is obtained in \(R_{\text{AN}}\) values while recovering about one-sixth (or nearly 16%) of all the active compounds from the test set. We also demonstrate that during the training process, an ANN basically learns to provide a gain “G” in the \(R_{\text{AN}}\) values from the initial values available in the validation or even the test set. Thus, to get an estimate of the G value from the validation set, actually provide a measure of the ratio of true positives to false positives. In addition, the \(R_{\text{AN}}\) value is found to increase when the predictions made by a large number of trained ANNs are combined. However, this increase is recurrently accompanied by a decrease in the actual number of recovered active compounds. To train those large numbers of ANNs without using numerous training samples, the generalization property of a back-propagation ANN is exploited.

Understanding the characteristics of the available data

Sixteen different features used to characterize each compound are: molecular weight, ClogP, molar reactivity, total surface area, polar surface, volume, polar volume, atom count, bond count, rotatable counts, ring count, diversity fingerprint, atom pair paths, H-bond acceptors, H-bond donors, and hydrophobic centers. For each compound, 16 values are used to define the fingerprint, and one additional number is used to define the inhibition activity. Thus, on combining the active and nonactive compounds, 43,347 feature vectors are made available for this investigation. As previously mentioned, the 1347 feature vectors correspond to the set of active compounds, whereas the remaining 42,000 feature vectors are considered for nonactive compounds. A 2-D scatter diagram is plotted in Figure 1 using 2 randomly selected components of the feature vectors from both classes. It shows that the features from both classes vastly overlap in that 2-D space. The scatter plots using different pairs of features also demonstrate similar behavior. Because of the high degree of overlap, if a classifier is not properly trained, the number of false positives may progressively increase as we try to recover more and more active compounds. Thus, the primary objective of this investigation is to train ANNs to successfully learn about the nonlinear decision boundaries between the 2 classes in the multidimensional feature space.

**Data Partitioning**

The computer-assisted HTS process is considered a 2-class pattern classification problem where one class consists of active compounds and the other class contains the nonactive compounds. The available data are divided into 2 parts: the development set and the test set. Because the original set had 1347 active and 42,000 nonactive compounds, the development set yields 674 active and 21,000 nonactive compounds, and the test set acquires 673 active and 21,000 nonactive compounds. The development set is then divided equally into the

![FIG. 1. Scatter plot of 2 randomly selected components, out of the total 16 components, of the normalized feature vectors for the entire data set.](attachment:image)
training and validation sets, where each set consists of 337 active and 10,500 nonactive compounds. The number of samples that will be used for training diverse networks would be different, but the trained networks will be validated by all the feature vectors of the validation set.

**ANN configuration and feature selection for validation**

Prior to training any ANN, it is essential to establish the requirements that an ANN needs to satisfy during the validation process. From the drug discovery point of view, identification of active compounds is the principal goal. The higher the ratio of active to nonactive compounds in the classified set, the better it is. However, following Figure 1, it is realized that the more active compounds we would want to identify, the more nonactive compounds will be misclassified as active compounds. Thus, our initial objective is to identify an optimal combination of network architectures and training features that would classify the validation set with $R_{AN}$ values higher than 0.0321.

The training and validation of ANNs are conducted interactively and iteratively while varying the design parameters of the classifier, such as network architectures, number of samples used for training, and convergence criteria. The basic architecture of an ANN used in this investigation is shown in Figure 2.

![Graphical representation of a feed-forward artificial neural network with 2 hidden layers.](image)

There are 16 fixed neurons or nodes at the input layer and 2 nodes at the output layer. The number of hidden layers is varied between 1 and 3, and the number of nodes in each hidden layer is selected interactively. All the neurons of the network possess a sigmoid input-output activation function. For a feature vector representing an active compound, the desired outputs from the first and second nodes of the output layer are set to $(1, 0)$, and the outputs are set to $(0, 1)$ for nonactive compounds. The overall mean squared error (MSE) criterion for convergence is set to be less than or equal to $10^{-4}$. These networks operate in the feed-forward mode and are trained with a back-propagation learning algorithm.

During the validation process, a feature vector from the validation set is applied to the input layer of a trained ANN, and the outputs from the 2 neurons of the output layer are computed in the feed-forward manner. If the output from the first node is greater than the output from the second node, the input pattern will be classified as active. Otherwise, the input pattern will be classified as nonactive. Basically, a winner-take-all scheme is used for classification.

Due to the presence of a disproportionate majority of nonactive compounds and a high overlap of the features from both classes in the feature space, selecting an appropriate number of training samples has been challenging. When numerous examples from both classes are used to train an ANN, the network encounters difficulty in learning to generalize, and often the network does not converge. On the other hand, training an ANN with just a few examples from both classes may cause the loss of the ability of an ANN to successfully learn about the complex nonlinear boundaries between 2 classes. Therefore, those ANNs cannot provide sufficient improvement in $R_{AN}$ values during validation.

After selecting the number of training samples to be used from both classes, actual feature vectors are collected randomly from the training set. These training vectors from both classes are treated as one set of vectors with 16 columns for each vector, and each column of that set is normalized independently between 0 and 1. Then, the normalized active and nonactive feature vectors are separated into 2 groups for training an ANN.

The maximum and minimum values for each feature that are used to normalize each column of the training vectors are also used to scale the corresponding feature of all the feature vectors of the validation set.

In addition to the use of original normalized features, the principal component analysis (PCA)–based features, discrete cosine transform (DCT)–based features, and wavelet transform–based features are extracted from the original data set for training and validating the ANNs. After training the ANNs with these new sets of features, the performances of these networks in classifying the validation set are studied, and the results are compared with those obtained from using the original normalized features. The goal here is to find an optimal combination of a set of features and network configurations so that higher $R_{AN}$ values are obtained from classifying the validation set. The compounds of the test set are then screened by that optimal combination.

**RESULTS**

A protracted interactive approach is used to find a set of network architectures and the number of training samples so that a practical classifier can be constructed. A fixed set of 30 nodes for the first hidden layer is used, and then varying the number of nodes in the second hidden layer from 5 to 20, $R_{AN}$ values of the order of 0.06 are obtained in several cases. The
number of training samples varied from 50 to 200 for the active compounds and from 300 to 600 for the nonactive compounds. Some of the results obtained from this part of the investigation are plotted as scattered circles in Figure 3. The maximum value of $R_{AN}$ is found to be equal to 0.09, whereas the ANN identifies only 89 active compounds. This implies that about one-fourth (26%) of all the active compounds available in the validation set are recovered with less than a 3-fold gain in the $R_{AN}$ value. In a 2-D plot, this result corresponds to a circle at the point (89, 0.09) as shown in Figure 3. A straight line is drawn from the location (89, 0.09) to the location (337, 0.0321), which represents characteristics of the validation set. The endpoints of this line are indicated by asterisks. Any result on or below this line can be produced by tweaking the training parameters of an ANN. However, results above the line may not be obtained by the chosen combination of the network architecture and training features. It has been shown that an ANN trained with 88 active and 1912 nonactive feature vectors provides a classified set with $R_{AN} = 0.1210$, and the number of active compounds found in this set was 129.12 Because the test set contained $1347 - 88 = 1259$ active compounds, the network has recovered only one-tenth of all the active compounds while providing a 4-fold gain in $R_{AN}$. Thus, the use of a large number of training samples might not be optimal in recovering a high portion of the active compounds.

**Classifier validation using different types of features**

PCA, DCT, and wavelet-based features are extracted from the original set to improve classification performance. For both PCA- and DCT-based transforms of the original feature vector, a new feature vector of 16 elements is computed for all the 43,347 compounds. For wavelet-based transforms, 43,347 feature vectors also are obtained but with only 9 wavelet coefficients per vector after the discrete wavelet transform.13 So, the input layer of the ANN consisted of 9 nodes instead of 16 when the wavelet-based features were used for training and validation. While maintaining the architectural framework of the ANNs, the aforementioned training and validation processes are repeated with each of the 3 new sets of feature vectors.

We have shown that an ANN can recover about half of the available active compounds with nearly a 2-fold gain in $R_{AN}$ values. However, it recovers less than one-fourth (89) of the available active compounds, whereas the gain in $R_{AN}$ value is slightly greater than 3-fold. So, we determine that a useful network should provide at least a 3-fold gain in the value of $R_{AN}$ while recovering at least one-third of all the available active compounds from the validation set. Thus, classification results around $(337/3), 3 \times 0.0321$ or $(112, 0.0963)$ are considered desirable. For each type of feature used to train and validate the ANNs, a line is drawn from the location $(337, 0.0321)$ to the location that comes closest to the aforementioned desirable results.

The endpoints of each line are depicted by asterisks. The networks trained with PCA-based features have provided superior performance over the others by extracting 177 active compounds with $R_{AN} = 0.0971$, as shown in Figure 3. The best results provided by the wavelet and DCT-based features are (102, 0.0631) and (107, 0.0814). Due to its superior performance, the PCA-based features are used for the rest of the investigation. In addition, all the 16 principal components are used for training, validation, and testing because the performance of the classifier deteriorates gradually as fewer and fewer principal components are used.

**Classifier training and validation using PCA-based features**

A large number of networks are trained and validated using PCA-based features. The training parameters are slowly varied from the configuration that previously provided the result (117, 0.0971). Different configurations of the trained networks and the corresponding classification results are summarized in Table 1, and the tabulated data show that the maximum value of $R_{AN}$ found during validation is 0.0996 while 125 active compounds are recovered. When the number of active compounds recovered reaches around 200, the $R_{AN}$ value decreases close to 0.06. The root cause for this problem is the considerable overlap of the features from both classes in the feature space. On the other hand, because the number of active compounds available is relatively small, there might be more common active compounds classified by the 2 different ANNs than the nonactive compounds. Therefore, in the next section, predictions made by multiple ANNs are combined and the performance of this combination process is studied.
Validation by combining outputs from multiple networks

Thus far, one ANN at a time has been used to classify the validation set, with limited success. We anticipate that during the training process, a single ANN learns some aspects of the shape of the boundary between 2 classes at a certain location of the feature space. As a result, it only succeeds to extract some active compounds while misclassifying a large number of nonactive compounds. Next, we train several ANNs with different sets of initial conditions so that these ANNs could learn about the boundaries between the 2 classes at different locations of the feature space. During validation, we expect that there will be more active compounds in common between the predictions made by 2 different ANNs because there are fewer active compounds available than the nonactive compounds. After combining the predictions made by different ANNs, the changes in the classification performance are investigated.

Table 1 shows an ANN consisting of 24 nodes in the first hidden layer and 10 nodes in the second hidden layer, trained with 60 and 393 examples from active and nonactive compounds, respectively, provided comparatively superior performance during validation. Subsequently, several networks with identical architectural design are constructed. To minimize the number of samples needed to train these ANNs, we have used the generalization property of the back-propagation ANNs. Consequently, different ANNs with the identical design, trained with the same examples after being initialized with different sets of randomized weights, would behave differently during the validation process. This property arises mainly because the converged weights are located at dissimilar positions of the multidimensional weight space for different networks, even with equivalent training examples. As a result, those different ANNs would provide similar output for the training vectors, but different outputs for other feature vectors are not used during training process.

After several ANNs of identical architecture are trained with the same 60 active and 393 nonactive compounds, all these ANNs are used to classify the validation set. From this set of a large number of ANNs, the first 10 networks that individually satisfy the criterion of a useful classifier are then selected to improve the overall performance of the classification process. The number of times a compound is predicted as active by these 10 networks is computed, and then a threshold number, nth, is used for classifying that compound. For example, if the compound number Nc has been identified as an active compound by P different networks, then the compound number Nc will be classified as active if P ≥ nth; otherwise, the compound number Nc will be considered nonactive. Using this approach, the classification results are computed for different values of nth. The outcome is presented in the left half of Table 2. This result shows that the procedure succeeds in classifying 139 active compounds for nth = 3 and R_{AN} = 0.1531. Thus, nearly a 5-fold gain in R_{AN} value is obtained while recovering more than one-third of all active compounds in the validation set. The gains in R_{AN} values are plotted in Figure 4 as a function of the number of recovered active compounds. It can be seen that the value of R_{AN} becomes 14-fold higher as compared to its original value when 63 active compounds are correctly classified. Higher R_{AN} values can be obtained as higher threshold values are used for combining outputs from these networks. In addition, the trend of recovering fewer active compounds with higher gains is also illustrated.

Screening of the test set

Ten networks that provided comparatively superior performance during the validation process are used to screen the 673 active and 21,000 nonactive compounds of the test set. The results obtained from combining the outputs of these networks are also presented in Table 2 as a function of the threshold. This outcome shows that the procedure succeeds in classifying 252 active compounds for nth = 3 while providing R_{AN} = 0.13. This R_{AN} value is about 4-fold higher than the value in the test set. The gain in R_{AN} values is plotted in Figure 4 as a function of the number of active compounds recovered from the test set.
It can be seen from Figure 4 that the gain in \( R_{AN} \) values becomes 10-fold when 104 active compounds are correctly classified, and a nearly 17-fold gain is obtained when 17 active compounds are correctly classified.

This approach combined the results from several useful classifiers to enhance or boost the performance of the overall classification process. However, this procedure differs from the popular “boosting” algorithm/procedure\(^4\) used in data mining publications. In that boosting algorithm, the training set of the \( i^{\text{th}} \) classifier is selected following the performances of the previous \((i – 1)\) classifiers. The goal is to enable each new classifier to predict those examples better that are poorly classified by the previous classifiers. In our approach, the individual classifiers are trained independent of each other, where each classifier satisfies the basic criterion of a useful classifier.

![Table 2](https://example.com/table2.png)

**Table 2.** Results after Combining Outputs from 10 High-Performance Artificial Neural Networks

<table>
<thead>
<tr>
<th>Threshold Used for Acceptance as an Active Compound (nth)</th>
<th>Actual Number of Active Compounds Found (NA)</th>
<th>Actual Number of Nonactive Compounds Found (NN)</th>
<th>( R_{AN} = \frac{\text{Active Compounds}}{\text{Nonactive Compounds}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>198</td>
<td>2338</td>
<td>0.0847</td>
</tr>
<tr>
<td>2</td>
<td>161</td>
<td>1463</td>
<td>0.1000</td>
</tr>
<tr>
<td>3</td>
<td>139</td>
<td>908</td>
<td>0.1531</td>
</tr>
<tr>
<td>4</td>
<td>109</td>
<td>537</td>
<td>0.2030</td>
</tr>
<tr>
<td>5</td>
<td>84</td>
<td>308</td>
<td>0.2727</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>149</td>
<td>0.4228</td>
</tr>
<tr>
<td>7</td>
<td>38</td>
<td>79</td>
<td>0.4810</td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>35</td>
<td>0.5429</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>5</td>
<td>0.6000</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>—</td>
</tr>
</tbody>
</table>

**DISCUSSION**

Thus far, we have used about half of all the available samples (21,674 out of 43,347 samples) to train and validate the ANNs. The remaining half, which consists of 21,673 compounds, has been screened by 10 selected high-performance networks. That set of 10 ANNs can now be employed to classify a very large industrial library of chemicals. It is also worthwhile to investigate the possibility of training and validating ANNs with fewer samples than what we have used, and the rationale is described below.

**Possible reduction in the size of the training and validation sets**

We demonstrate that an individual ANN can be designed and trained with only 60 examples of active and 393 nonactive compounds to improve the ratio of \( R_{AN} \) in the classified set by 3-fold from the original data set. Furthermore, it takes several hours to train ANNs with more than a total of 1000 examples. Even if a network converges with a large number of training samples, its classification performance is not that useful. In addition, when the ratio of the nonactive to active compounds increases beyond 10 during the training procedure, the value of \( R_{AN} \) becomes less than 0.06 during validation. Thus, we conclude that a maximum of randomly selected 100 active and 1000 nonactive samples should be adequate for the training set. We could subsequently select 60 active and 393 nonactive compounds from this training set. So, we construct a new training set using a total of only 1100 samples. To determine an acceptable size for the validation set, we have used the following procedure.

After training many different configurations of backpropagations ANNs, we have observed that each ANN learns something very basic through the training process. It actually learns to classify an untrained data set by making an “F-fold” improvement in the \( R_{AN} \) value, and an almost inverse relationship...
exists on the number of active compounds recovered by that ANN. For example, if an ANN improves the value of RA/N value by 3-fold from classifying an untrained set of data, almost one-third of all active compounds will then be recovered from that set. This observation is illustrated in Table 2. So, to estimate the gain in RA/N that a trained ANN would provide during testing, it is not necessary to analyze and classify a large validation set, but the ratio of the active to nonactive compounds in the validation set should be kept very similar to that of the test set. Thus, a validation set is constructed with 100 active and 3000 nonactive compounds.

An ANN possessing 24 nodes in the first hidden layer and 10 nodes in the second hidden layer, as before, is trained with 60 active and 393 nonactive compounds collected from the new training set of 1100 compounds. After training, the ANN is used to analyze the validation set that consists of 100 active and 3000 nonactive compounds, and the ANN recovers 31 active compounds with an RA/N value equal to 0.0940. This results in a 3-fold gain in RA/N value, and about one-third of the active compounds are recovered from the validation set. Ten identically designed ANNs are then trained using the same training samples, and the predictions made by these ANNs in analyzing the validation set are combined using a threshold value as discussed before. The results are presented in the left half of Table 3, where we used a total of 200 active and 4000 nonactive compounds by combining training and validation sets. The remaining data set of 1137 active and 38,000 nonactive compounds is then tested with those 10 ANNs. The results obtained on combining the outcomes from the 10 ANNs are presented in the right half of Table 3. Furthermore, the results obtained from analyzing this large test set of 1137 + 38,000 = 39,137 compounds and that obtained from the previously classified test set with 673 + 21,000 = 21,673 samples are plotted in Figure 5. It is shown that smaller training and validation sets allow a larger pool of test data to be analyzed, and more active samples are recovered at gains lower than 6 in RA/N values.

### Table 3. Results from Using a Limited Number of Compounds for Training and Validation

<table>
<thead>
<tr>
<th>Threshold Used for Acceptance as an Active Compound (nth)</th>
<th>Actual Number of Active Compounds Found (NA)</th>
<th>Actual Number of Nonactive Compounds Found (NN)</th>
<th>RA/N = NA/NN</th>
<th>Actual Number of Active Compounds Found (NA)</th>
<th>Actual Number of Nonactive Compounds Found (NN)</th>
<th>RA/N = NA/NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46</td>
<td>575</td>
<td>0.0800</td>
<td>587</td>
<td>6809</td>
<td>0.0862</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>317</td>
<td>0.1198</td>
<td>447</td>
<td>3966</td>
<td>0.1127</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>162</td>
<td>0.1975</td>
<td>336</td>
<td>2283</td>
<td>0.1471</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>81</td>
<td>0.3209</td>
<td>229</td>
<td>1236</td>
<td>0.1853</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>33</td>
<td>0.4545</td>
<td>144</td>
<td>610</td>
<td>0.2360</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>6</td>
<td>0.6667</td>
<td>75</td>
<td>208</td>
<td>0.3605</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>13</td>
<td>24</td>
<td>0.5416</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>—</td>
</tr>
</tbody>
</table>

FIG. 5. Gain in the ratio of the active to nonactive compounds achieved from analyzing small and large test sets.

However, similar numbers of active compounds are recovered from both the test sets at any gain greater than 6. Thus, the use of only 4200 samples for the combined training and validation sets should be adequate to construct ANNs that would meaningfully improve the values of RA/N from classifying the test set.

### Comparison with statistical k-NN–based classification scheme

A straightforward nonparametric k-nearest neighbor (k-NN)–based classification algorithm is also used to classify the large aforementioned test set.10,11 The same 60 examples from the active class and 393 examples from the nonactive class that were used to train the ANNs are selected to form the training set for the k-NN–based classification scheme. Thus, a total of
60 + 393 = 453 training samples are used to screen the large test set that consists of 39,137 compounds. For a given test vector, the k-NN algorithm computes the Euclidean distances of the test vector with all the training vectors from both classes and generates a list of distances in the ascending order. The algorithm then takes the lowest "k" distances from the set and finds the class for each of those "k" training vectors. The input test vector is assigned to that class "i" that appears more frequently in the list of k minimum distances. We used i = 1 and 2 to represent active and nonactive classes, respectively. In this investigation, the value "k" is varied from 11 to 51 in steps of 4, and the gain in $R_{an}$ values is also plotted in Figure 5 as a function of the number of active compounds recovered. In this case, higher gains in $R_{an}$ values are obtained for higher values of k. It can be seen from Figure 5 that the ANN-based classification scheme has clearly provided better performance over the k-NN–based algorithm. Improved performance is expected from the k-NN–based approach if more training samples are used in the training set. However, the focus of this investigation has been the comparison of the performances of these 2 methods using identical training samples.

CONCLUDING REMARKS

The HTS data were subjugated by the nonactive compounds. The initial ratio of the active to nonactive compounds ($R_{an}$) was 0.0321, and the actual numbers of active and nonactive compounds were 1347 and 42,000, respectively. In addition, the feature vectors of the compounds from both classes exceedingly overlapped in the 16-dimensional feature space. The challenge was to develop an ANN-based design that would provide a $R_{an}$ value considerably higher than 0.0321 while correctly identifying a large number of active compounds.

Initially, the given data set was divided into training, validation, and test sets. Both the training and validation sets contained 337 active and 10,500 nonactive compounds, while the test set contained 673 active and 21,000 nonactive compounds. Among the original, PCA, DCT, and wavelet-based features, the PCA-based features provided the best classification performance. When a single ANN was trained with the PCA-based features, it provided a 3-fold gain in the $R_{an}$ value while recovering about one-third of all the active compounds available for the validation set. The overall classification performance for the validation set was substantially improved on combining the predictions made by 10 different networks, where each ANN provided about a 3-fold gain in $R_{an}$ values. The generalization property of the ANNs was exploited when all those 10 ANNs were trained with the same training examples after being initialized with diverse sets of random weights. As a result, we used a total of only 60 + 393 = 453 samples for training all the ANNs, which is about 1% of the available samples. These 10 networks were also used to classify the test set. As expected, combining the outputs from these networks resulted in approximately a 10-fold gain in $R_{an}$ value while recovering 104 or about one-sixth of all active compounds available for testing. The number of active compounds extracted from the test set decreased as higher $R_{an}$ values were obtained. This fact could become particularly useful if our intention is to collect a limited number of active compounds with high precision. For example, Table 2 showed that we could extract 47 active compounds with the value $R_{an}$ equal to 0.4393, which is about a 14-fold improvement.

The back-propagation ANN’s ability to learn and classify features from multiple classes, using a limited number of training examples, has enabled us to develop a training set with only 1100 examples. We also empirically established the fact that once an ANN was trained, it basically learned to classify an untrained set by F-fold improvement in the $R_{an}$ value while recovering a fraction $1/F$ of the available active compounds. As a result, we could use a smaller validation set maintaining the original ratio of $R_{an}$. We selected 100 examples of active and 3000 of nonactive compounds to create the validation set. A total of 1100 + 3100 = 4200 samples for training and validating the ANNs were used, and the remaining 1137 examples of active and 38,000 of nonactive compounds formed a new test set of 39,137 samples. Two important facts were revealed from the comparative analysis of this large test set and previous test sets. First, considerable improvement was obtained in recovering the number of active compounds from the larger test set at small gains in $R_{an}$ values. Then, at higher gains in $R_{an}$ values, the number of active compounds recovered became almost equal for both test sets. Thus, using only 4200 samples (or 10%) for the combined training and validation sets, similar classification performances were obtained when 21,674 samples (or 50%) of the data were used for the combined training and validation sets.

In addition, because fewer active compounds were successfully identified with higher $R_{an}$ values, this procedure could be used to collect a small number of active compounds with high precision from a large library of compounds. Therefore, the ANN-based procedure has a vital potential in providing an efficient and low-cost solution to identify active compounds from large data sets.

ACKNOWLEDGMENTS

This work was supported by the K-INBRE Bioinformatics Core, NIH grant number P20 RR016475 and Kansas IDDRC, P30 NICHD HD07526. We thank Dr. Gerald H. Lushington (Molecular Graphics and Modeling Laboratory, The University of Kansas, Lawrence, KS) for providing all the molecular descriptors. We also thank Dr. Qi-Zhuang Ye, former director of HTS Laboratory, for his assistance. The ITTC and the EECS Department of the University of Kansas have provided the computing resources.
REFERENCES

8. High Throughput Screening Laboratory at the University of Kansas [Online]. Retrieved from http://www.hts.ku.edu

Address correspondence to:
Swapan Chakrabarti
Department of Electrical Engineering and Computer Science
University of Kansas
2001 Eaton Hall, 1520 W. 15th Street, Lawrence, KS 66045

E-mail: chakra@eecs.ku.edu