Predictive Toxicology: Benchmarking Molecular Descriptors and Statistical Methods

Jun Feng,[†] Laura Lurati,[‡] Haojun Ouyang,[§] Tracy Robinson,[||] Yuanyuan Wang,[⊥] Shenglan Yuan,[#] and S. Stanley Young*,[⊗]

University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, Brown University, Providence, Rhode Island 02912, University of Toledo, Toledo, Ohio 43606, North Carolina State University, Raleigh, North Carolina 27695, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, Auburn University, Auburn, Alabama 36849, and National Institute of Statistical Sciences, 3401 Caldwell Drive, Raleigh, North Carolina 27607

Received February 17, 2003

The development of drugs depends on finding compounds that have beneficial effects with a minimum of toxic effects. The measurement of toxic effects is typically time-consuming and expensive, so there is a need to be able to predict toxic effects from the compound structure. Predicting toxic effects is expected to be challenging because there are usually multiple toxic mechanisms involved. In this paper, combinations of different chemical descriptors and popular statistical methods were applied to the problem of predictive toxicology. Four data sets were collected and cleaned, and four different sets of chemical descriptors were calculated for the compounds in each of the four data sets. Three statistical methods (recursive partitioning, neural networks, and partial least squares) were used to attempt to link chemical descriptors to the response. Good predictions were achieved in the two smaller data sets; we found for large data sets that the results were less effective, indicating that new chemical descriptors or statistical methods are needed. All of the methods and descriptors worked to a degree, but our work hints that certain descriptors work better with specific statistical methods so there is a need for better understanding and for continued methods development.

INTRODUCTION

The development of drugs depends on finding compounds that have beneficial effects with minimal toxic effects. Currently, we do not have the knowledge and computational power to directly predict toxic effects, so we have to rely on structure−activity relationship (SAR) methods. Unlike the initial screening and lead optimization, in which the binding affinities to a specific receptor are typically modeled, toxicity often involves many different receptors and mechanisms, but with the same toxicology response, death, mutagenicity, etc.

Currently, there are two predictive toxicology strategies:[1] knowledge-based and statistics-based. Knowledge-based methods rely on rules from human experts or previous knowledge on structure−toxicity relationships, while statistics-based methods rely on using training data sets, chemical descriptors, and statistical methods to correlate the observed toxicity data to structural features and generate a mathematical predictive model. The predictive model is used to evaluate untested compounds. There are a number of commercially available software packages for making predictions of toxic effects: DEREK,[2] ONCOLOGIC,[3] HAZARDEXPERT,[4] TOPKAT,[5] etc. DEREK, ONCOLOGIC, and HAZARD-EXPERT are knowledge-based, and TOPKAT is statistics-based.[1]

Generally, multiple linear regression is the dominant statistical method used for predictive toxicology. Because most toxicology predictions involve heterogeneous compound classes and multiple toxic mechanisms, we expect more complex statistical methods such as recursive partitioning (RP), neural networks (NN), and partial least squares (PLS) should perform better than linear regression. We are also curious that certain types of chemical descriptors might work better with certain types of statistical analysis so we studied the performance of different chemical descriptors with different statistical methods. We want our findings to have some generality so our study used four different data sets.

Our goal is to investigate the performance of three statistical methods (PLS, NN, and RP) and four types of chemical descriptors: constitutional, topological index, BCUT,[6−9] and fragment/property descriptors (CONS, TI, BCUT, and FRAG). The statistical methods and molecular descriptors are described in more detail later. We are interested if certain types of descriptors work better with specific statistical analysis methods. For example, do TIs work better with PLS and fragments work better with NN? The three statistical methods we used are advertised to work well in complex situations. PLS is capable of dealing with the situation where there are more descriptors than observations. NN is capable of dealing with nonlinear relationships. RP is capable of dealing with multiple mechanisms. We used four publicly available data sets (three toxicology data sets and one potency data set) and calculated four sets of chemical descriptors using free-ware DRAGON.[10] These data sets, SD files and descriptors, are posted at www.niss.org. Altogether there are 48 situations, four data sets, four sets of chemical descriptors, and three statistical methods.

[†] University of North Carolina at Chapel Hill.
[‡] Brown University.
[§] University of Toledo.
[||] North Carolina State University.
[⊥] University of Waterloo.
[#] Auburn University.
[⊗] National Institute of Statistical Sciences.

10.1021/ci034032s CCC: $25.00
Published on Web 07/15/2003

Table 1. Data Sets Used, Their Size, Description, and Where They Can Be Obtained[a]

data set	size	description	response type
NCI	~29000	HIV	1/0
Yeast	~9000	PC score of six variables	continuous
Mut	~1800	Ames test	1/0
Tox	~270	LC50 aquatic organism	continuous

[a] These data files can be downloaded from www.niss.org.

This research followed the following plan. First, four public data sets were obtained. The point here is to determine the effectiveness of the chemical descriptors and statistical methods over a variety of data sets. Second, four classes of descriptors were computed for each data set. There is considerable interest in molecular descriptors and, again, we wanted to access the different statistical methods using different descriptors. Third, we used three popular statistical analysis methods, PLS, NN, and RP. Each data set was divided evenly at random into two data sets, a training data set for training the model and testing data set for testing the predictions of the model. A prediction equation was made using a statistical method, a type of molecular descriptor, and the training data. The quality of the prediction was evaluated using the testing set. Because an evaluation of all statistical methods on all data sets and descriptor sets would require more labor than available, we used a statistical sampling plan to select 15 of the 48 combinations of data set/descriptor type/statistical methods. Three of the 15 conditions were "replicated" by resplitting the data at random to give a new training and testing data set. Examination of the replicate evaluations can give a sense of the variability of the results.

This paper is organized as follows. First, we describe the data sets and data processing. Second, we describe four sets of chemical descriptors. Third, we describe the three statistical methods. We then use statistical design of experiments to select an informative subset from the 48 possible combinations of statistical method, descriptor type, and data set. Last, we give the results and conclude with a discussion.

METHODS (DATA, DESCRIPTORS, AND STATISTICAL METHODS)

Data and Data Processing. The data sets used are listed in Table 1. There are several tasks involved in data processing. The logistical order of these tasks is (1) collect data sets, (2) preformat data for compatibility with DRAGON, (3) compute the descriptors, and (4) postformat data for compatibility with ChemTree[11] and JMP.[12]

The descriptors were computed using DRAGON, developed by Todeschini, Consonni, and Pavan of Milano Chemometrics. The descriptors used were separated into four sets listed in Table 2. These descriptors are described in work by Todeschini and Consonni (2000)[13] and can be viewed at http://www.dist.umimib.com. DRAGON is limited to processing 1500 compounds at a time, so the files were divided, processed, and reconstituted. Incompatible electronic compound representations were also resolved, or the compound was eliminated.

Four publicly available data sets are selected for analysis (Table 1):

Table 2. Descriptor Set Name and Numbers of Descriptors[a]

descriptor set	no. of descriptors	no. of eigenvalues > 1
BCUT	64	4
CONS	47	14
TI	260	30
FRAG	247	57

[a] We also give the number of eigenvalues for each set for the NCI data set greater than 1 to indicate the approximate dimensional size of the set.

(1) Acute Toxicity data set (Tox):[14] This data set contains 278 substituted benzenes. The IC_{50} toxicity to *T. pyriformis* is used as the toxicity end point.

(2) Mutagenicity data set (Mut): The response of this data set is based on the Ames test for mutagenicity.[15] Usually four strains of bacteria are tested, with or without metabolic activation. If any of the eight tests is positive, the compound is considered positive. This data set is collected from various public sources like EPA, NIH, etc. The original number of compounds for this data set is 2018; because some compounds failed to be converted to three-dimensional structures by CONCORD in Sybyl and some compounds did not meet the requirement of DRAGON, the final number of compounds is 1863.

(3) NCI Yeast Anticancer Drug Screen data set (Yeast): This data set is obtained from http://dtp.nci.nih.gov/yacds. All compounds in this data set were screened against six kinds of mutant strains. The dose concentration is 50 μM. The activity of each compound is expressed as the growth inhibition percentage for strains treated with the compound compared to strains treated with solvent only. There are around 100 000 compounds in this data set, but to reduce the work load, we randomly picked out 10 000 compounds for analysis. Compounds that have more than 150 atoms or contain metal atoms are discarded, which resulted in 8885 compounds.

(4) NCI AntiHIV Drug Screen data set: (NCI): This data set is obtained from http://dtp.nci.nih.gov/docs/aids/aids_data.html. It has a categorical response measuring how a compound protects human CEM cells from HIV-1 infection.

Chemical Descriptors. All descriptors were computed from SD files using DRAGON.

(a) CONS Descriptors (Reference 13, pp 90 and 91). There are 47 descriptors. Examples include molecular weight, atomic weight, atomic counts, etc. Descriptors in this class are not determined by the connectivity or conformation of the molecule.

(b) Topological Information Indices (Reference 13, pp 447−456). TI descriptors are widely used in QSAR analysis because they are easy to calculate, do not depend on conformation, and are sensitive to small changes in molecular structure. DRAGON can calculate 262 kinds of TIs.

(c) BCUT.[7,8,13] Similar to TI, BCUT descriptors are also determined by the connectivity, i.e., topological relations between different atoms within the molecule. BCUT descriptors are calculated from the adjacency matrix, also called the Burden matrix. In this matrix, atomic properties are placed on the major diagonal and a measure of connection is placed in the off-diagonal cells. Through diagonalization of the adjacency matrix, eight highest eigenvalues and eight

lowest eigenvalues are used. The four atomic properties are atomic mass, van der Waals volume, atomic electronegativity, and atomic polarizability, so there are $4 \times 16 = 64$ descriptors. We found that BCUT descriptors are highly intercorrelated with each other. Details will be shown in the next section.

(d) FRAG Descriptors.[13] We also tried to use some descriptors that reflect the physicochemistry properties, like log *P*, aromatic index, etc. Fragment descriptors, which indicate what kinds of fragments are in the molecule and how many of them there are, are also included in this class.

Table 2 lists the number of descriptors in each class and the number of eigenvalues greater than 1 for each class as an indication of the number of "real" variables in each class. We were surprised at the relatively small number of greater than 1 eigenvalues for each class and particularly surprised that there were only four eigenvalues greater than 1 for BCUTs.

Statistical Methods. The statistical programs used for this analysis were JMP and ChemTree. Some of the data processing was done in SAS. ChemTree has internally computed fragement-based descriptors, but these were not used for this study. JMP contains a binary RP method, but ChemTree was used because it can perform multiway splitting and has a random tree generation function.

PLS. PLS is a statistical method for the analysis of systems of independent and response variables. Weighted linear combinations of the predictor variables are used to predict the response variable(s). PLS is a predictive technique that can handle more independent than response variables and can relate the set of independent variables to a set of multiple dependent (response) variables. PLS regression has been used in various disciplines such as chemistry, economics, medicine, psychology, and pharmaceutical science, where predictive linear modeling, especially with a large number of predictors, is necessary. PLS regression is probably the least restrictive of the various multivariate extensions of multiple linear regression. This flexibility allows it to be used in situations where traditional multivariate methods fail, such as when there are fewer observations than predictor variables. PLS regression can also be used as an exploratory analysis tool to help select suitable predictor variables and to identify outliers before classical linear regression is used.

PLS is an extension of multiple linear regression. In multiple linear regression, a model specifies the (linear) relationship between a dependent (response) variable *Y* and a set of independent (predictor) variables, the *X*'s, so that

$$Y = b_0 + b_1X_1 + b_2X_2 + ... + b_pX_p$$

where b_0 is the regression coefficient for the intercept and the b_i values are the regression coefficients (for variables $1-p$) computed from the observed data.

In our case we use PLS to build a linear model on each training set, $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where **Y** is an *m* cases by *r* variable(s) response matrix, where *m* is the number of compounds in the training set. In our case $r = 1$. For the yeast data, we initially chose two principal components, i.e., $r = 2$. To simplify this paper, we restricted our reported analysis to only one response. **X** is an *m* cases by *p* variable predictor (design) matrix, **B** is a *p* by *r* regression coefficient matrix, and **E** is a error term for the model which has the same dimensions as **Y**. Usually, the variables in **X** and **Y** are centered by subtracting their means and scaled by dividing by their standard deviation.

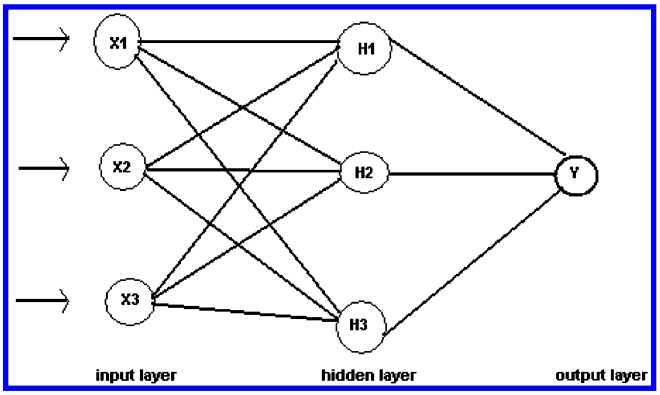

Figure 1. Prototypical feedforward NN with one hidden layer (three hidden units).

PLS simplifies the descriptors by use of factor scores, linear combinations of the original predictor variables. There is no correlation between the factor score variables used in the predictive regression model. For example, suppose we have a data set with response variables **Y** (in matrix form) and a large number of predictor variables **X** (in matrix form), some of which are highly correlated. A regression, using factor extraction for these types of data, computes the factor score matrix $\mathbf{T} = \mathbf{XW}$ for an appropriate weight matrix **W** and then considers the linear regression model $\mathbf{Y} = \mathbf{TQ} + \mathbf{E}$, where **Q** is a matrix of regression coefficients (loadings) for **T** and **E** is an error (noise) term. Once the loadings **Q** are computed, the above regression model is equivalent to $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B} = \mathbf{WQ}$, which can be used as a predictive regression model.

PLS produces the weight matrix **W** reflecting the covariance structure between the predictor and response variables. PLS produces a *p* by *c* weight matrix **W** for **X** such that $\mathbf{T} = \mathbf{XW}$; i.e., the columns of **W** are weight vectors for the **X** columns producing the corresponding *n* by *c* factor score matrix **T**. These weights are computed so that each of them maximizes the covariance between responses and the corresponding factor scores. Ordinary least-squares procedures for the regression of **Y** on **T** are then performed to produce **Q**, the loadings for **Y** (or weights for **Y**) such that $\mathbf{Y} = \mathbf{TQ} + \mathbf{E}$. Once **Q** is computed, we have $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B} = \mathbf{WQ}$, and the prediction model is complete. We used the PLS platform in JMP for the analysis work in this paper.

NN. Historically, the model structure of NN was based on our understanding of a biological neuron system. The human brain is a highly connected set of neurons. A neuron has inputs from other neurons and outputs to other neurons. Like neurons, a NN model tries to determine the linear combination of the explanatory variables that can predict a target response as functions of input features.[16]

Turning to mathematics, a NN provides a way to approximate a general nonlinear function. Study and use of NN has evolved to a large number of methods and algorithms, and there is less emphasis on the biological origins of the methodology. A feedforward NN with one hidden layer is the simplest but most common form in use.[17] This form is available in JMP, and we use this form for our predictions.

Figure 1 gives a diagram of a simple, feedforward NN with one hidden layer (three hidden units). It has three input

(explanatory) variables and a single output (response) variable. Between the input and output layers is a hidden layer. The hidden layer can have any number of units; here it has just three.

The output units, y_k's, are functions of input information, x_i's:[18]

$$y_k = G_0\left[a_k + \sum_h w_{hk} G_h\left(a_h + \sum_i w_{ih} x_i\right)\right]$$

Usually, the function G_h on the hidden layers is a logistic function defined as

$$G(x) = \frac{\exp(x)}{1 + \exp(x)}$$

The appropriate regression or classification network is constructed with a corresponding continuous or categorical response. The parameters a_k's, a_h's, w_{hk}'s, and w_{ih}'s can be determined by minimizing some error function such as the least-squares criterion:

$$R(a_k\text{'s}, a_h\text{'s}, w_{hk}\text{'s}, w_{ih}\text{'s}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Next, we comment on the training of a NN. In practice, because NN often have a large number of weights, the optimization of an error function of $R(a_k\text{'s}, a_h\text{'s}, w_{hk}\text{'s}, w_{ih}\text{'s})$ is very computationally intensive and the global minima will often overfit the data. In effect the NN can memorize the training data set. JMP attempts to get around an excessive number of weights and overfitting by adding a penalty to the error function:

$$K(a_k\text{'s}, a_h\text{'s}, w_{hk}\text{'s}, w_{ih}\text{'s}) = R(a_k\text{'s}, a_h\text{'s}, w_{hk}\text{'s}, w_{ih}\text{'s}) + \lambda L(a_k\text{'s}, a_h\text{'s}, w_{hk}\text{'s}, w_{ih}\text{'s})$$

where λ is called a weight decay parameter; the function L will become larger with larger effects of weights. The optimization algorithm will tend to minimize the penalty function K instead of R and a large value of λ will shrink the weights toward zero.

Because the function K is nonconvex, there are many local minima. With fixed $\lambda = 0.01$ (default), JMP randomly selects 20 (default) starting points and finds the best local minimum that has the least value of K as a final solution.

NN fitted in JMP have only one hidden layer with three hidden units (default). Many books argue that it is better to have too many hidden unites than too few. Hastie et al. (2001)[16] recommend that the number of hidden units be in the range of 5−100, with the number increasing with the number of inputs and the number of training cases.

Specifically, in our application we use all of the JMP default options because we believe that is what most people would try first. It should provide a fair comparison to the results from other methods where we use default options as well. Note that RP, NN, and PLS can all be tuned to the problem at hand. Each software vendor has set the defaults at presumably good settings. It is beyond the scope of this research to exhaustively tune the methods.

NN are touted as very powerful learning methods with widespread application in many fields. Unfortunately, it is essentially uninterruptible because it is difficult to determine which input variables are influential. There are conflicting claims in the literature on the superiority of one method or type of chemical descriptors, so it will be of interest to compare statistical methods using different types of chemical descriptors. It is obvious that we will not resolve this controversy because each statistical method can be tuned by experts and there is no end of chemical descriptors that can be computed and endless ways to mix and match them to the statistical method. Our study, by necessity, is limited, but it does point to a way to benchmark statistical methods and chemical descriptors.

RP. RP is a data mining method for finding predictive patterns in large, complex data sets. RP progressively divides a data set into smaller and more homogeneous subsets. The two main purposes of using trees are to aid in identifying the underlying data structure and to predict values of future observations. A tree is a *classification tree* if the predicted variable is a category, e.g., active/inactive, and a *regression tree* if the end point is a continuous number, e.g., LD50. There are a number of RP algorithms;[19] we used ChemTree, which is specifically designed for the analysis of chemistry data sets.

RP works from a response variable and a measurement vector **x**, which contains information about an observation on many different variables (x_1, x_2, ..., x_p). In this case, the measurement vector consists of different compound descriptors. The sample that we use to create a tree is referred to as the learning or training sample.

The data set and descriptor combinations used with RP are as follows: NCI with TI, NCI with BCUT, mutagenicity with CONS, mutagenicity with FRAG, and toxicity with FRAG. The training and test sets were a random selected division of each entire set. A generic tree from ChemTree is given in Figure 2.

In each interior node, the information consists of the splitting variable information, the number of observations in the node (n), the toxicity mean (u), the standard deviation (s), and the Bonferroni adjusted P value for the split. The algorithm finds the best "cut points" for each continuous descriptor to best divide the observations into homogeneous groups. It then selects among variables for the best variable. This is computationally intensive. The adjusted p value is used to help ensure that the split reflect cause rather than chance.

The first step in constructing a tree is to determine the selection of splits. The basic idea is to split at each node in a way that will make the descendant nodes more homogeneous based on their toxicity value. In other words, as we split into more nodes, the observations in each node will be more similar to each other than they are to observations in other nodes. If the response variable is binomial, then a χ^2 test is performed to measure the dissimilarity of the two daughter groups, and if it is continuous, then the dissimilarity is measured by a function based on maximum likelihood equations (see *ChemTree Manual*, 2002; p 38). P values are calculated for either of these tests and then adjusted by multiplying by the Bonferroni corrector factor, which is the number of independent variables that could actually be used to split the node. A node is terminal when the split P values are no longer significant for any predictor variable. For the binomial toxicity response (1 = toxic, 0 = not), if the mean of the terminal node is greater than 0.5, then the node's

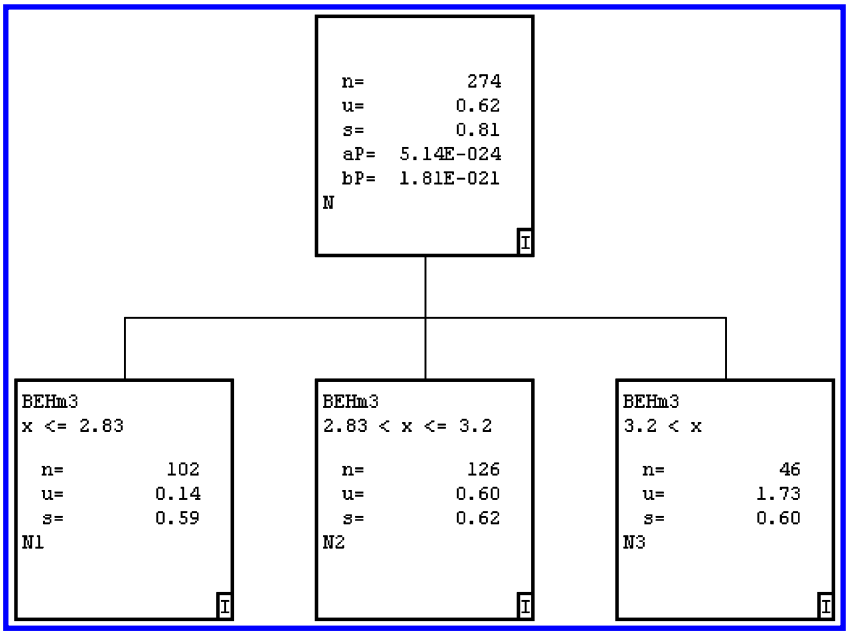

Figure 2. First split for RP tree for Tox data set and BCUT descriptors.

activity is classified as toxic and not toxic otherwise. For the mutagenicity data set, about half the compounds are measured as mutagens, so using 50% to classify a node makes sense; when the positive effect is very rare, e.g., compound screening, then we would declare a node positive with a far lower positive rate. For a continuous toxicity response, the toxicity of a terminal node is the mean value of that node. Once the tree is created, the test data set is "sent" through the tree so the predictive quality of the tree can be measured. For our continuous response data sets, an R^2 value is calculated for the observed (x) versus the predicted (y) values. For our binomial response data sets, a χ^2 value is calculated.

$$R^2 = \Sigma(x_{\text{bar}} - x_{\text{I}})(y_{\text{bar}} - y_{\text{I}})/\sqrt{\Sigma(y_{\text{bar}} - y_{\text{I}})\Sigma(x_{\text{bar}} - x_{\text{I}})}$$

$$\chi^2 = \Sigma[\Sigma(\text{actual} - \text{expected})^2/\text{expected}]$$

DESIGN OF EXPERIMENTS

Now, we have three factors: data sets, molecular descriptors, and statistical methods. The goal of this project is to find out the main effects of descriptors and methods and the interaction between them within the blocking effect of the data sets. The ideal situation is that we can figure out which descriptors or methods are consistently good or bad across the data sets and how the statistical methods and descriptors interact with each other. Thus, our result can be a guide to further study.

Here are the experimental factors with their levels:

(i) Data sets (four levels): Tox, Mut, Yeast, NCI

(ii) Descriptors (four levels): CONS, TI, BCUT, FRAG

(iii) Methods (three levels): PLS, NN, RP

The data sets should be considered as "block" in that the statistical methods and chemical descriptors are run within each data set. Each data set has a specified number of compounds, and the biological potencies are more likely to be measured consistently within a data set. To run a complete experimental design would require $4 \times 4 \times 3 = 48$ runs.

Table 3. Design of Experiments

run	data set	descriptor type	statistical method
1[a]	Mut	CONS	RP
2	Yeast	CONS	NN
3	Tox	FRAG	NN
4	Tox	BCUT	PLS
5	NCI	TI	RP
6	Mut	CONS	PLS
7	Tox	FRAG	PLS
8	Mut	BCUT	NN
9[a]	NCI	FRAG	PLS
10	NCI	BCUT	RP
11	Yeast	TI	PLS
12[a]	Mut	TI	NN
13	Mut	FRAG	RP
14	Yeast	BCUT	PLS
15	Tox	FRAG	RP

[a] Each of these conditions was replicated twice.

Because we had only limited resources for our data analysis and our investigation is considered to be preliminary, we decided to run 18 experiments. The detailed plan is as follows: we select 15 out of 48 runs so that we could estimate all of the main effects and two-way interactions between descriptors and statistical methods. We effectively treat the data sets as blocks. We then select 3 out of 15 selected runs to replicate. One replicate is selected at random within each of the three statistical methods. The 15-point design is constructed using JMP.

Table 3 shows the 15 experiments; an *a* is used to mark conditions that are replicated. To form a replicate, we resplit the data at random to form a new training and testing set.

RESULTS

RP. Using the topological descriptors with the NCI data set resulted in a tree with 18 terminal nodes, of which only two were classified as being potent. In the original test data set, 301 compounds are potent, while the tree only predicted that 11 compounds are potent. If we had tested each compound individually for potency, we would have a hit rate of $301/14568 = 0.020\,66$, but if we test only the ones

Table 4. Result of RP for NCI Data Set[a]

		actual		
	count total (%)	0	1	
predicted	0	14528	299	14577
		97.87	2.05	99.92
	1	9	2	11
		0.06	0.01	0.08
		14267	301	14568
		97.93	2.07	

[a] NCI Data with Topological Descriptor Contingency Table.

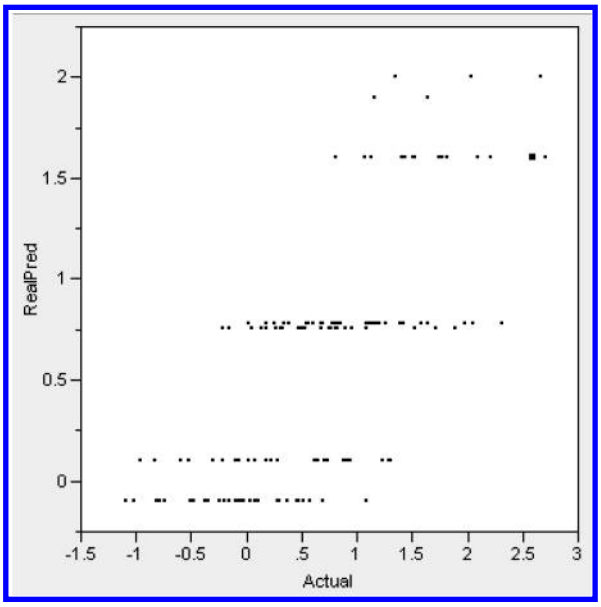

Figure 3. Observed versus predicted for Tox data set, FRAG descriptors, and RP analysis.

that are predicted to be potent, we have a hit rate of 0.181 82, an 8.8-fold increase in hit rate. To compare this method with the other methods, a two-way contingency table, Table 4, is created. χ^2 for this 2 × 2 table is 5.473. A total of 5 of the 19 terminal nodes are classified as potent for the tree created using the BCUT descriptors on the NCI data set. The hit rate for this set increased from 0.020 66 to 0.527 03, and the χ^2 test value is 206.687. The next data we worked with were the mutagenicity data. The CONS descriptors were used to create trees for two different random splits of test versus treatment. The first tree had 24 terminal nodes, with 13 of those nodes being classified as toxic. The hit rate increased from 0.485 95 to 0.6825, and the χ^2 test value is 119.101. The second tree had a hit rate increase from 0.498 35 to 0.748 48 and a χ^2 test value of 134.090. A total 10 of its 23 terminal nodes are classified as toxic. The last binomial response tree was created using the FRAG descriptor set on the mutagenicity data. With 9 out of 16 terminal nodes being toxic, the hit rate for this tree increased from 0.485 95 to 0.754 72 and it has a χ^2 test value of 197.519. The last data set is the toxicity data set, which has a continuous response variable. The R^2 value for using the FRAG descriptor set to create a tree on this set is 0.494 835. An observed versus predicted graph is provided (Figure 3).

PLS. Our data sets have two types of response: continuous response and binary response. For binary response, logistic regression may be a better tool, but because the goal of our study is to investigate the performance of PLS analysis, we have to convert the predicted continuous response to binary response by applying an arbitrary cutoff. The selection of the cutoff will solely depend on how good it works on the training set; the cutoff value that gives the best classification of active and inactive compounds in the training set will be used for the testing set. Any prediction that is larger than this cutoff value will be considered as 1, while any prediction that is less than this cutoff value will be considered as 0.

Table 5. Least-Squares Means of −log(*P* value) for Interaction of the Statistical Method and Descriptor Type

	descriptor type			
statistical method	CONS	FRAG	TI	BCUT
PLS	28.17	43.15	18.58	18.67
NN	21.26	19.09	28.45	3.38
RP	22.76	38.26	12.52	56.95

There are six PLS analyses with different combinations of descriptors and methods: BCUT descriptors on a Tox data set, BCUT descriptors on a Yeast data set, CONS descriptors on a Mut data set, TI descriptors on a Yeast data set, FRAG descriptors on a Tox data set, and FRAG descriptors on a NCI data set. The cutoff value used for CONS descriptors on a Mut data set is 0.95, and the cutoff value used for FRAG descriptors on a NCI data set is 0.10.

The PLS method can handle the situation when there are more descriptors than number of observations, but for most of our test cases, there are usually many more observations than descriptors. The best results were obtained with the Tox data set, which contains 270 compounds. We noted that, as the size of the data set increased, the result became less satisfactory, especially for the NCI and Yeast data sets. We suspect that those data sets are too structurally diverse, and the compounds may have operated by different mechanisms. The results, −log(*p* value) of the association of predicted versus observed for test data sets, are included in Table 5.

NN. As given in the experimental plan, a NN was applied to three data sets once—Yeast data with CONS descriptors (YC), Tox data with FRAG descriptors (TF), and Mut data with BCUT descriptors (MB)—and one data set twice—Mut data with TI descriptors (MT).

All of these four data sets have many explanatory variables, which is a computational disaster to training NN models. For example, even with the YC data having only 47 descriptors, NN with only three hidden units have to estimate about 150 weights. It is almost impossible to fit NN for this many descriptors in JMP. Therefore, for each data set, we calculate the first 10 principal components and used these as the inputs, which made the computation feasible. In this way we only need to estimate about 40 parameters. PLS and RP used all variables to derive their models; there could be loss of information in using PC scores for NN, but it was not computationally feasible to use the large number of predictors with NN.

For YC, TF, and MB data, we randomly split data into training and test sets. For MT, two random divisions were applied. So, we have five data sets, each having the training and test sets. We used NN, built on the training set, to predict the test set and calculate R^2 (for regression) or χ^2 (for classification) for linear regression between the prediction and the observation of the response on the test set.

Statistical Analysis. For each of the 18 experimental conditions, data set, chemical descriptor, and statistical

Table 6. −log(*P* value) for Fit of Observed versus Predicted for Test Set Using a Model from the Training Set

ID	data set	descriptor	statistical method	−log(*p* value)
1	Mut	CONS	RP	27.00[a]
2	Mut	CONS	RP	30.28[a]
3	Yeast	CONS	NN	42.52
4	Tox	FRAG	NN	2.76
5	Tox	BCUT	PLS	2.34
6	NCI	TI	RP	1.71
7	Mut	CONS	PLS	34.05
8	Tox	FRAG	PLS	26.82
9	Mut	BCUT	NN	9.26
10	NCI	FRAG	PLS	36.22[a]
11	NCI	FRAG	PLS	28.45[a]
12	NCI	BCUT	RP	46.14
13	Yeast	TI	PLS	39.84
14	Mut	TI	NN	29.00[a]
15	Mut	TI	NN	39.67[a]
16	Mut	FRAG	RP	44.14
17	Yeast	BCUT	PLS	39.93
18	Tox	FRAG	RP	21.93

[a] Replicate pairs.

Table 7. Analysis of Variance

source	DF	sum of squares	mean square	*F* ratio
model	14	3614.0438	258.146	8.3732
error	3	92.4901	30.830	Prob > *F*
C. total	17	3706.5339		0.0528

Table 8. Effect Tests

source	Nparm	DF	sum of squares	*F* ratio	Prob > *F*
data	3	3	973.4229	10.5246	0.0422
Des	3	3	239.9667	2.5945	0.2272
Stat	2	2	256.7875	4.1646	0.1363
Des*Stat	6	6	1672.4314	9.0411	0.0492

method, we computed the association between the predicted and observed values for the test data set. The prediction model was constructed using the training data set. The *p* value for this association was transformed using the negative log base 10 (Table 6). For continuous response data sets, the *p* values are obtained from R^2, and for binary response data sets, the *p* values are obtained from χ^2. The analysis of variance for these data is given in Table 7. Although just missing the nominal statistical significance of 0.05, we see that there is an indication of significant effects when the three replicates are used to test the remaining effects. The model includes a number of different factors, so we examine those factors in the Effects Tests (Table 8). We see that there are significant differences among the data sets. It is expected that data sets will differ in the ability to be modeled. One should conclude that multiple data sets should be used for benchmarking studies. There is an indication of an interaction between the chemical descriptors and the statistical methods, $p < 0.0492$. The presence of an interaction is not unexpected. The interaction effect is just barely statistically significant, so more benchmarking should be done. If the interaction is true, the implication is that particular descriptors will work more effectively with particular statistical methods. Interactions can be examined in a number of ways. Table 9 gives the least-squares means for the statistical method by molecular descriptor type. It appears that BCUT descriptors do not work well with NN and appear to work quite well for RP. (It should be noted that these are BCUTs from DRAGON and differ from those of Pearlman and Smith.[6])

Table 9. Least-Squares Means

	CONS	TI	BCUT	FRAG
PLS	28.17	18.58	18.67	43.15
NN	21.26	28.45	3.38	19.09
RP	22.76	12.52	56.95	38.26

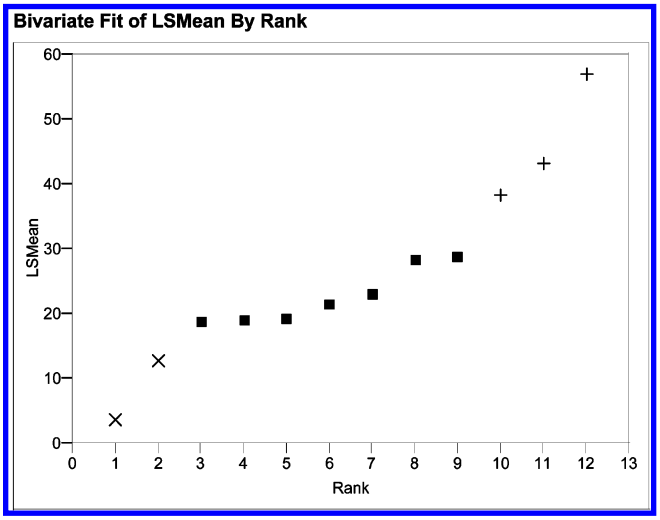

Figure 4. Plot of least-squares means versus their rank for interaction of the statistical method and chemical descriptor.

It is often useful to plot the ranked values against the integers (Figure 4). Three values in the plot appear to be larger than expected: BCUT and RP (56.95), FRAG and PLS (43.15), and FRAG and RP (38.26). Two values in the plot appear to be smaller than expected: BCUT and NN (3.38) and TI and RP (12.52).

COMMENTS AND CONCLUSIONS

The focus of this work is to demonstrate a statistical way of benchmarking, analyzing the interaction among data sets, descriptor sets, and statistical methods. Our conclusions are tentative. More data sets should be used in benchmarking, and undoubtedly the particular statistical methods could be improved by expert tuning.

All of the statistical methods were effective in the sense that *p* values were small for all types of molecular descriptors. It was surprising that there was not a clear winner either for the statistical method or for the molecular descriptor. It is not clear which statistical method should have won because each of the methods is highly touted. That the descriptors work at all, given their simplicity or abstractness in the case of BCUTs and TIs, might be considered surprising. We chose not to use all of the descriptors at one time because we wanted to get a sense of the utility of the rather different types of descriptors. It is of interest if more descriptors will do better. We are dubious given the high redundancy among these descriptors. The *p* values for the three replicates differed by more than we expected; this suggests that cross-validation studies need to be replicated. (Typically, replication of cross validation is not done.) The apparent interaction between the chemical descriptors and the statistical methods suggests that simple validation studies using one type of chemical descriptor and one statistical method might miss good opportunities. The apparent interaction also suggests that specific descriptors and statistical methods might be exploited to one's advantage.

Data Availability. Four sets of descriptors are posted at www.niss.org for each of the four data sets used in this study. We also post a SD file for each data set.

ACKNOWLEDGMENT

This paper is based on work carried out as part of the 2002 CRSC/SAMSI Industrial Mathematical Modeling Workshop sponsored by the Center for Research in Scientific Computation (CRSC) and the Statistical and Applied Mathematical Sciences Institute (SAMSI) partially funded by the National Science Foundation under grants DMS-0204515 and DMS-0112069, respectively. The authors would also like to gratefully acknowledge the support and assistance of H. T. Banks of the CRSC and Terry Byron of the Department of Statistics at NCSU during this project.

REFERENCES AND NOTES

(1) Greene, N. *Adv. Drug Delivery Rev.* **2002**, *54*, 417−431.
(2) DEREK is described at http://lhasa.harvard.edu.
(3) Dearden, J. C.; et al. *ATLA, Altern. Lab. Anim.* **1997**, *25*, 223−252.
(4) HAZARDEXPERT is described at http://www.compudrug.hu/hazard.html.
(5) TOPKAT is described at http://www.accelrys.com/products/topkat.
(6) Pearlman, R. S.; Smith, K. M. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 28−35.
(7) Burden, F. R. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 225−227.
(8) Burden, F. R. *Quant. Struct.−Act. Relat.* **1997**, *16*, 309−314.
(9) Stanton, D. T. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 11−20.
(10) DRAGON can be downloaded from http://www.dist.umimib.com.
(11) ChemTree is described at http://www.goldenhelix.com.
(12) JMP is described at http://www.jmpdiscovery.com.
(13) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Viley-VCH: Weinheim, Germany, 2000.
(14) Burden, F. R.; Winkler, D. A. *Chem. Res. Toxicol.* **2000**, *13*, 436−440.
(15) Young, S. S.; Gombar, V. K.; Emptage, M. R.; Cariello, N. F.; Lambert, C. *Chemom. Intell. Lab. Syst.* **2002**, *60*, 5−11.
(16) Hastie, T.; Tibshirani, R.; Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*; Springer: New York, 2001.
(17) Ripley, B. D. *Pattern Recognition and Neural Networks*; Cambridge University Press: Cambridge, U.K., 1996.
(18) Venables, W. N.; Ripley, B. D. *Modern Applied Statistics with S-plus*, 3rd ed.; Springer: New York, 1999.
(19) Rusinko, A.; Farmen, M. W. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017−1026.

CI034032S