Hazard Assessment Modeling: An Evolutionary Ensemble Approach

David W. Opitz
Department of Computer Science
University of Montana
Missoula, MT 59812 (USA)
opitz@cs.umt.edu
406-243-2831

Subhash C. Basak **Brian D. Gute**
Natural Resources Research Institute
University of Minnesota
Duluth, MN 55811 (USA)
{sbasak, bgute}@wyle.nrri.umn.edu
218-720-4230

Abstract

This paper presents a novel and effective genetic algorithm approach for generating computational models for hazard assessment. With millions of proposed chemicals being registered each year, it is impossible to come even remotely close to completing the battery of tests needed for the proper understanding of the toxic effects of these chemicals. Computer models can give quick, cheap, and environmentally friendly hazard assessments of chemicals. Our approach works by first extracting a hierarchy of theoretical descriptors of the structure of a compound, then filtering these numerous descriptors with a genetic algorithm approach to ensemble feature selection. We tested the utility of our approach by modeling the acute aquatic toxicity (LC_{50}) of a congeneric set of 69 benzene derivatives. Our results demonstrate a very important point: that our method is able to accurately predict toxicity directly from structure.

1 INTRODUCTION

By the end of 1998 the number of chemicals registered with the Chemical Abstract Service rose to over 19 million (CAS 1999). This is an increase of over 3 million chemicals between 1996 and 1998. It is desirable to test each of these chemicals for their effects on the environment and human health (which we refer to as *hazard assessment*); however, completing the battery of tests necessary for the proper hazard assessment of even a single compound is a costly and time-consuming process. Therefore, there is simply not enough time or money to complete these test batteries for even a tiny portion of the compounds which are registered today (Menzel 1995). An alternative to these traditional test batteries is to develop computational models for hazard assessment. Computational models are fast (milliseconds per compound), cheap (less than one cent per compound), and do not run the risk of adversely affecting the environment during testing. Additionally, these computational methods can replace or limit the amount of animal testing that is necessary. Thus computational models can easily process *all* registered chemicals and flag the ones that require further testing. The central problem with this approach is developing class specific models that can be considered accurate enough to be useful. In this paper, we present a novel and effective approach for learning computational hazard assessment models by using an ensemble feature selection algorithm based on genetic algorithms (GAs) to filter numerous theoretical descriptors of chemical structure.

To better illustrate the need for effective and quick hazard assessment, we should consider the situation of the industrial chemicals "grandfathered" into continued use under the Toxic Substances Control Act (TSCA) of 1976. TSCA has required that a suite of physicochemical and toxicological screens be run on all commercial compounds (those produced or imported in volumes exceeding one million pounds annually) developed after 1976. However, there are almost 3,000 chemicals that were "grandfathered" in with the understanding that it would be the responsibility of the chemical manufacturing industry to ultimately supply information about these chemicals. Only recently, after a 20-year delay, are the chemical manufacturers talking about running 2,800 of these compounds through basic toxicity screens and while this is promising, these screens will not be completed until 2004 and at a cost of between $500 to $700 million dollars. So it will be another five years before we have basic toxicity data on compounds that have been in wide-spread use for more than twenty years (Johnson 1998).

One of the fundamental principles of biochemistry is that activity is dictated by structure (Hansch 1976). Following this principle, one can use theoretical molecular descriptors that quantify structural aspects of a molecule to quantitatively determine its activity (Basak & Grunwald 1995; Cramer, Famini, & Lowrey 1993). These theoretical descriptors can be generated directly from the known structure of the molecule and used to estimate its properties, without the need for further experimental data. This is important due to that fact that, with chemicals needing to be evaluated for hazard assessment, there is a scarcity of available experimental data that is normally required as inputs (i.e., independent variables) to traditional quantitative structure-activity relationship (QSAR) model development. A QSAR model based solely on theoretical descriptors on the other hand can process all registered chemicals for hazard assessment.

Our hierarchical approach examines the relative contributions of theoretical descriptors of gradually increasing complexity (structural, chemical, shape, and quantum chemical descriptors). This approach is important as none of the individual classes of parameters are very effective at predicting toxicity (Gute & Basak 1997); however, we show in this paper that we can effectively predict toxicity if we combine all levels of descriptors. One potential problem with using our hierarchical approach is that it often gives many independent variables as compared to data points since having a limited number of data points in not uncommon in hazard assessment. For instance, in our case study of predicting acute toxicity (LC_{50}) of benzene derivatives, we have 95 independent variables and 69 data points. Therefore, reducing the number of independent variables is critical when attempting to model small data sets. The smaller the data set, the greater the chance of spurious error when using a large number of independent variables (descriptors). In some of our earlier QSAR studies we have used statistical methods such as principal components analysis (PCA) and variable clustering methods to reduce the number of independent variables (Basak & Grunwald 1995; Gute & Basak 1997; Gute, Grunwald, & Basak In press).

As an alternative solution, we use our previous ensemble feature selection approach (Opitz 1999) that is based on GAs. An "ensemble" is a combination of the outputs from a *set* of models that are generated from separately trained inductive learning algorithms. Ensembles have been shown to, in most cases, greatly improve generalization accuracy over a single learning model (Breiman 1996; Maclin & Opitz 1997; Shapire *et al.* 1997). Recent research has shown that an effective ensemble should consist of a set of models that are not only highly correct, but ones that make their errors on different parts of the input space as well (Hansen & Salamon 1990; Krogh & Vedelsby 1995; Opitz & Shavlik 1996a). Varying the feature subsets used by each member of the ensemble helps promote the necessary diversity and create a more effective ensemble (Opitz 1999). We use GAs to search through the enormous space of finding a set of feature subsets that will promote disagreement among the component members of an ensemble while still maintaining the component member's accuracy.

Combining our approach of generating hierarchical theoretical descriptors with our other approach to GA-based ensemble feature selection, we are able to generate an effective model for predicting the toxicity of benzene derivatives using only a few compounds. Our results show that our model is nearly as accurate as the battery of tests necessary for the proper hazard assessment of a single compound. Our results also confirm that our new ensemble feature selection approach is more effective than previous approaches for modeling hazard assessment.

The rest of the paper is organized as follows. First we provide background and related work for both our hierarchical QSAR approach and our GA-based ensemble feature selection approach. This is followed by results of our approach applied to benzene derivatives. Finally, we discuss these results and provide future work.

2 QSAR AND THEORETICAL METHODS

QSARs have come into widespread use for the prediction of various molecular properties, as well as biological, pharmacological and toxicological responses. Traditional QSAR techniques use empirical properties (Dearden 1990; Hansch & Leo 1995; de Waterbeemd 1995); however, due to the scarcity of available data for the majority of chemicals needing to be evaluated for hazard assessment, these physicochemical properties necessary for traditional QSAR model development may not be available. When this is the case, it is imperative that there are methods available which make use of nonempirical parameters, which we term theoretical molecular descriptors.

Topological indices (TIs) are numerical graph invariants that quantify certain aspects of molecular structure (Gute & Basak 1997; Gute, Grunwald, & Basak In press). The different classes of TIs provide us with nonempirical, quantitative descriptors that can be used in place of experimentally derived descriptors in QSARs for the prediction of properties.

Our recent studies have focused on the role of different classes of theoretical descriptors of increasing levels of complexity and their utility in QSAR (Gute & Basak 1997; Gute, Grunwald, & Basak In press). Four distinct sets of theoretical descriptors have been used in this study: topostructural, topochemical, geometric, and quantum chemical indices. Gute and Basak 1997 provide the detailed list of the indices included in our study.

2.1 TOPOLOGICAL INDICES

The topostructural and topochemical indices fall into the category normally considered topological indices. Topostructural indices (TSIs) are topological indices that only encode information about the adjacency and distances of atoms (vertices) in molecular structures (graphs), irrespective of the chemical nature of the atoms involved in bonding or factors such as hybridization states and the number of core/valence electrons in individual atoms. Topochemical indices (TCIs) are parameters that quantify information regarding the topology (connectivity of atoms), as well as specific chemical properties of the atoms comprising a molecule. These indices are derived from weighted molecular graphs where each vertex (atom) or edge (bond) is properly weighted with selected chemical or physical property information.

The complete set of topological indices used in this study, both the topostructural and the topochemical, have been calculated using POLLY 2.3 (Basak, Harriss, & Magnuson 1988) and software developed by the authors. These indices include the Wiener index (Wiener 1947), the connectivity indices developed by Randic 1975 and higher order connectivity indices formulated by Kier and Hall 1986, bonding connectivity indices defined by Basak and Magnuson 1988, a set of information theoretic indices defined on the distance matrices of simple molecular graphs (Hansch & Leo 1995), and neighborhood complexity indices of hydrogen-filled molecular graphs, and Balaban's 1983 J indices.

2.2 GEOMETRICAL INDICES

The geometrical indices are three-dimensional Wiener numbers for hydrogen-filled molecular structure, hydrogen-suppressed molecular structure, and van der Waals volume. Van der Waals volume, V_W (Bondi 1964), was calculated using Sybyl 6.1 from Tripos Associates, Inc. of St. Louis. The 3-D Wiener numbers were calculated by Sybyl using an SPL (Sybyl Programming Language) program developed in our lab (SYBYL 1998). Calculation of 3-D Wiener numbers consists of the sum entries in the upper triangular submatrix of the topographic Euclidean distance matrix for a molecule. The 3-D coordinates for the atoms were determined using CONCORD 3.0.1 from Tripos Associates, Inc. Two variants of the 3-D Wiener number were calculated: ${}^{3D}W_H$ and ${}^{3D}W$. For ${}^{3D}W_H$, hydrogen atoms are included in the computations and for ${}^{3D}W$ hydrogen atoms are excluded from the computations.

2.3 QUANTAM CHEMICAL PARAMETERS

The following quantum chemical parameters were calculated using the Austin Model version one (AM1) semi-empirical Hamiltonian: energy of the highest occupied molecular orbital (E_{HOMO}), energy of the second highest occupied molecular orbital (E_{HOMO1}), energy of the lowest unoccupied molecular orbital (E_{LUMO}), energy of the second lowest unoccupied molecular orbital (E_{LUMO1}), heat of formation (ΔH_f), and dipole moment (μ). These parameters were calculated using MOPAC 6.00 in the SYBYL interface (Stewart 1990).

3 FILTERING DESCRIPTORS

As stated above, one potential problem with including all theoretical descriptors in the hierarchy is that it gives many independent variables when compared to the limited number of data points available for hazard assessment modeling of a particular chemical derivative. Compounding this problem is that a salient descriptor for one hazard assessment model may not be a salient descriptor for another problem. That is, the relevance of a descriptor for predicting hazard assessment is often problem dependent. This section describes our approach for automatically filtering the descriptors with a GA-based approach to ensemble feature detection. Before explaining our algorithm, we briefly cover the notion of ensembles.

3.1 ENSEMBLES

Figure 1 illustrates the basic framework of a predictor ensemble. Each predictor in the ensemble (predictor 1 through predictor N in this case) is first trained using the training instances. Then, for each example, the predicted output of each of these predictors (o_i in Figure 1) is combined to produce the output of the ensemble ($\hat{o}$ in Figure 1). Many researchers (Breiman 1996; Hansen & Salamon 1990; Krogh & Vedelsby 1995;

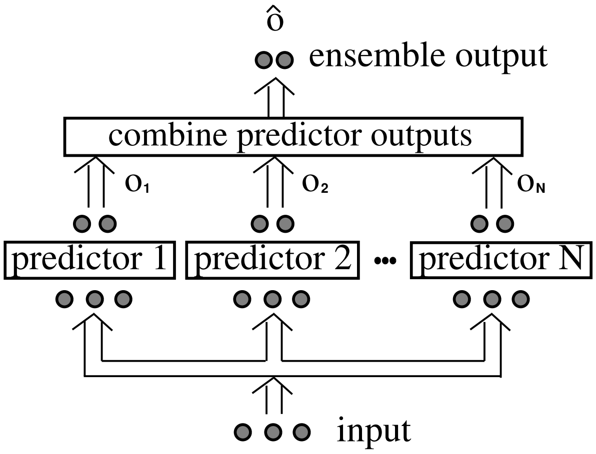

Figure 1: A predictor ensemble.

Opitz & Shavlik 1997) have demonstrated the effectiveness of combining schemes that are simply the weighted average of the predictors (i.e., $\hat{o} = \sum_{i \in N} w_i \cdot o_i$ and $\sum_{i \in N} w_i = 1$), and this is the type of ensemble on which we focus in this article.

Combining the output of several predictors is useful only if there is disagreement on some inputs. Obviously, combining several identical predictors produces no gain. Hansen and Salamon 1990 proved that for an ensemble, if the average error rate for an example is less than 50% and the predictors in the ensemble are independent in the production of their errors, the expected error for that example can be reduced to zero as the number of predictors combined goes to infinity; however, such assumptions rarely hold in practice.

Krogh and Vedelsby 1995 later proved that the ensemble error can be divided into a term measuring the average generalization error of each individual predictor and a term called diversity that measures the disagreement among the predictors. Formally, they define the diversity term, d_i, of predictor i on input x to be:

$$d_i(x) \equiv [o_i(x) - \hat{o}(x)]^2. \quad (1)$$

The quadratic error of predictor i and of the ensemble are, respectively:

$$\epsilon_i(x) \equiv [o_i(x) - f(x)]^2, \quad (2)$$

$$e(x) \equiv [\hat{o}(x) - f(x)]^2, \quad (3)$$

where $f(x)$ is the target value for input x. If we define $\hat{E}$, E_i, and D_i to be the averages, over the input distribution, of $e(x)$, $\epsilon(x)$, and $d(x)$ respectively, then the ensemble's generalization error can be shown to consist of two distinct portions:

$$\hat{E} = \bar{E} - \bar{D}, \quad (4)$$

where $\bar{E}$ ($= \sum_i w_i E_i$) is the weighted average of the individual predictor's generalization error and $\bar{D}$ ($= \sum_i w_i D_i$) is the weighted average of the diversity among these predictors. What the equation shows then, is that an ideal ensemble consists of highly correct predictors that disagree as much as possible. Opitz and Shavlik 1996a; 1996b empirically verified that such ensembles generalize well.

Regardless of theoretical justifications, methods for creating ensembles center around producing predictors that disagree on their predictions. Generally, these methods focus on altering the training process in the hope that the resulting predictors will produce different predictions. For example, neural network techniques that have been employed include methods for training with different topologies, different initial weights, different parameters, and training only on a portion of the training set (Alpaydin 1993; Freund & Schapire 1996; Hansen & Salamon 1990; Maclin & Shavlik 1995).

Numerous techniques try to generate disagreement among the classifiers by altering the training set each classifier sees. The two most popular techniques are Bagging (Breiman 1996) and Boosting (Freund & Schapire 1996). Bagging is a bootstrap ensemble method that trains each network in the ensemble with a different partition of the training set. It generates each partition by randomly drawing, with replacement, N examples from the training set, where N is the size of the training set. As with Bagging, Boosting also chooses a training set of size N and initially sets the probability of picking each example to be $1/N$. After the first network, however, these probabilities change to emphasize misclassified instances. A large number of extensive empirical studies have shown that these are highly successful methods that nearly always generalize better than their individual component predictors (Bauer & Kohavi 1998; Maclin & Opitz 1997; Quinlan 1996). Neither approach is appropriate for our domain since we are data poor and cannot afford to waste training examples; however, we are feature rich and can afford to create diversity by instead varying the inputs to the learning algorithms. *Varying the feature subsets to create a diverse set of accurate predictors is the focus of the next section.*

3.2 THE GEFS ALGORITHM

The goal of our algorithm is to find a set of feature subsets that creates an ensemble of classifiers (neural networks in this study) that maximize equation 1 while minimizing equation 2. The space of candidate sets is enormous and thus is particularly well suited for ge-

Table 1: The GEFS algorithm.

GOAL: Find a set of input subsets to create an accurate and diverse classifier ensemble.

1. Using varying inputs, create and train the initial population of classifiers.
2. Until a stopping criterion is reached:
   (a) Use genetic operators to create new networks.
   (b) Measure the diversity of each network with respect to the current population.
   (c) Normalize the accuracy scores and the diversity scores of the individual networks.
   (d) Calculate fitness of each population member.
   (e) Prune the population to the N fittest networks.
   (f) Adjust λ.
   (g) The current population is the ensemble.

---

netic algorithms. Table 1 summarizes our recent algorithm (Opitz 1999) called GEFS (for Genetic Ensemble Feature Selection) that uses GAs to generate a set of classifiers that are accurate and diverse in their predictions. GEFS starts by creating and training its initial population of networks. The representation of each individual of our population is simply a dynamic length string of integers, where each integer indexes a particular feature. We create networks from these strings by first having the input nodes match the string of integers, then creating a standard single-hidden-layer, fully connected neural network. Our algorithm then creates new networks by using the genetic operators of crossover and mutation.

GEFS trains these new individuals using backpropogation. It adds new networks to the population and then scores each population member with respect to its prediction accuracy and diversity. GEFS normalizes these scores, then defines the fitness of each population member (i) to be:

$$Fitness_i = Accuracy_i + \lambda\ Diversity_i \qquad (5)$$

where λ defines the tradeoff between accuracy and diversity. Finally, GEFS prunes the population to the N most-fit members, then repeats this process. At every point in time, the current ensemble consists of simply averaging (with equal weight) the predictions of the output of each member of the current population. Thus as the population evolves, so does the ensemble.

We define accuracy to be network i's training-set accuracy. (One may use a validation-set if there are enough training instances.) We define diversity to be the average difference between the prediction of our component classifier and the ensemble. We then separately normalize both terms so that the values range from 0 to 1. Normalizing both terms allows λ to have the same meaning across domains.

It is not always clear at what value one should set λ; therefore, we automatically adjust λ based on the discrete derivatives of the ensemble error $\hat{E}$, the average population error $\bar{E}$, and the average diversity $\bar{D}$ within the ensemble. First, we never change λ if $\hat{E}$ is decreasing; otherwise we (a) increase λ if $\bar{E}$ is not increasing and the population diversity $\bar{D}$ is decreasing; or (b) decrease λ if $\bar{E}$ is increasing and $\bar{D}$ is not decreasing. We started λ at 1.0 for the experiments in this article. The amount λ changes is 10% of its current value.

We create the initial population by randomly choosing the number of features to include in each feature subset. For classifier i, the size of each feature subset (N_i) is independently chosen from a uniform distribution between 1 and twice the number of original features in the dataset. We then randomly pick, with replacement, N_i features to include in classifier i's training set. Note that some features may be picked multiple times while others may not be picked at all; replicating inputs for a neural network may give the network a better chance to utilize that feature during training. Also, replicating a feature in a genome encoding allows that feature to better survive to future generations.

Our crossover operator uses dynamic-length, uniform crossover. In this case, we chose the feature subsets of two individuals in the current population proportional to fitness. Each feature in both parent's subset is independently considered and randomly placed in the feature set of one of the two children. Thus it is possible to have a feature set that is larger (or smaller) than the largest (or smallest) of either parent's feature subset. Our mutation operator works much like traditional genetic algorithms; we randomly replace a small percentage of a parent's feature subset with new features. With both operators, the network is trained from scratch using the new feature subset; thus no internal structure of the parents are saved during the crossover.

4 RESULTS

We tested the utility of combining our approach for generating numerous hierarchical theoretical descriptors of compounds with our approach for filtering these descriptors with GEFS by modeling the acute aquatic toxicity (LC_{50}) of a congeneric set of 69 benzene derivatives. The data was taken from the work of Hall, Kier and Phipps 1984 where acute aquatic toxicity was measured in fathead minnow (*Pimephales promelas*). Their data was compiled from eight other sources, as well as some original work which was conducted at the U.S. Environmental Protection Agency (USEPA) Environmental Research Laboratory in Duluth, Minnesota. This set of chemicals was composed of benzene and 68 substituted benzene derivatives.

Table 2 gives our results. We studied three approaches for modeling toxicity: (1) giving all theoretical descriptors to a neural network, (2) reducing the feature set in a traditional previously published (Gute & Basak 1997) manner, and (3) using our new genetic algorithm technique on the entire feature set to create a neural network ensemble. Results for our approaches are from leave-one-out experiments (i.e., 69 training/test set partitions). Leave-one-out works by leaving one data point out of the training set and giving the remaining instances (68 in this case) to the learning algorithms for training. (It is worth noting that each member of the ensemble sees the same 68 training instances for each training/test set partition and thus ensembles have no unfair advantage over other learners.) This process is repeated 69 times so that each example is a part of the test set once and only once. Leave-one-out tests *generalization* accuracy of a learner, whereas training set accuracy tests only the learner's ability to memorize. Generalization error from the test set is the true test of accuracy and is what we report here.

We first trained neural networks using all 95 parameters. The networks contained 15 hidden units and we trained the networks for 1000 epochs. We normalized each input parameter to a values between 0 and 1 before training. Additional parameter settings for the neural networks included a learning rate of 0.05, a momentum term of 0.1, and weights initialized randomly between -0.25 and 0.25. With all 95 input parameters, the neural networks obtained a test-set correlation coefficient between predicted toxicity and measured toxicity (explained variance) of $R^2 = 0.868$ and a standard error of 0.29. Target toxicity measurements ranged from 3.04 to 6.37.

Our first method for feature-set reduction follows the work of Gute and Basak 1997 on toxicity domains. Their method begins by using the VARCLUS method of SAS 1998 to select subsets of topostructural and topochemical parameters for QSAR model development. With this method, the set of topological indices is first partitioned into two distinct sets, the topostructural indices and the topochemical indices.

Table 2: Relative effectiveness of statistical and neural network methods in estimating LC_{50} of 69 benzene derivatives.

Method	R^2	Standard Error
NN with 95 inputs	0.868	0.29
VARCLUS	0.825	0.32
NN with GEFS	0.893	0.27

To further reduce the number of independent variables for model construction, the sets of topostructural and topochemical indices were further divided into subsets, or clusters, based on the correlation matrix using the VARCLUS procedure. This procedure divides the set of indices into disjoint clusters, such that each cluster is essentially unidimensional. From each cluster we selected the index most correlated with the cluster, as well as any indices which were poorly correlated with their cluster ($R^2 < 0.70$). The variable clustering and selection of indices was performed independently for both the topostructural and topochemical indices. This procedure resulted in a set of five topostructural indices and a set of nine topochemical indices. These indices were combined with the three geometric and six quantum chemical parameters described earlier. Their approach then applied linear regression to these 23 parameters. This study found that an accurate linear regression model for acute aquatic toxicity required descriptors from all four levels of the hierarchy: topostructural, topochemical, geometrical and quantum chemical. This model utilized seven descriptors and obtained an explained variance (R^2) of 0.863 and a standard error of 0.30 on the whole data set used as a training set. Our leave-one-out experiment gave an $R^2 = 0.825$ and a standard error of 0.32.

Finally we applied our genetic algorithm technique, GEFS, using all 95 parameters. The parameter settings for the networks in the ensemble were the same as the settings for the single networks in the first experiment. Parameter settings for the genetic algorithm portion of GEFS includes a mutation rate of 50%, a population size of 20, a $\lambda = 1.0$, and a search length of 100 networks (20 networks for the initial population and 80 networks created from crossover and mutation). While the mutation rate may seem high as compared with traditional genetic algorithms, certain aspects of our approach call for a higher mutation rate (such as the criterion of generating a population that cooperates as well as our emphasis on diversity); other mutation values were tried during our pilot studies. With this approach, we obtained a test-set correlation coefficient of $R^2 = 0.893$ and a standard error of 0.27; the initial population of 20 networks obtained a test-set

$R^2 = 0.835$ and a standard error of 0.31.

5 DISCUSSION AND FUTURE WORK

The correlation coefficient between the predicted value from the computational model and the target value derived from the toxicity test is an extremely informative metric of accuracy in this case. The exact numeric value of most toxicity tests is not as important as the relative ordering and spread of these values. Thus, a perfect correlation ($R^2 = 1.0$) between the computation model and target toxicity shows the computational model is as informative as the toxicity obtained from a battery of expensive and time-consuming tests – regardless of the standard error. Note the standard error of 0.27 is fairly good, given the toxicity measurements ranged from 3.04 to 6.37.

While the neural network technique and the standard data-reduction technique obtained decent correlation with measured toxicity, our ensemble technique was about 20% closer to perfect correlation. Note that GEFS produces an accurate initial population and that running GEFS longer with our genetic operators can further increase performance. Thus our approach can be viewed as an "anytime" learning algorithm. Such a learning algorithm should produce a good concept quickly, then continue to search concept space, reporting the new "best" concept whenever one is found (Opitz & Shavlik 1997). This is important since, for most hazard assessment, an expert is willing to wait for days, or even weeks, if a learning system can produce an improved model for predicting toxicity.

Our results demonstrate a very important point: that our method is able to accurately predict toxicity directly from structure. Compared to the actual battery of tests necessary to measure toxicity, a computer model is much cheaper, much faster, and does not have a negative impact on the environment. It is important to also note that the computer model does not have to be the final measurement for hazard assessment; additional tests can be run on compounds that are either flagged by the model, or require more tests by the nature of their use (such as a benzene derivative that may become a standard fuel). Not only can good computer models become filters, they will probably be the only viable option for processing all registered chemicals.

While the method proposed here has proven effective, there is much future work that needs to be completed. For instance, we plan to test our method on other data sets of chemical derivatives; investigate other ensemble feature selection techniques; investigate variants to our genetic algorithm approach, and finally investigate the utility of other descriptors, such as bio-descriptors.

6 CONCLUSIONS

In this paper we presented a novel approach for creating a computer model for hazard assessment. Our approach works by first extracting a hierarchy of theoretical descriptors derived from the structure of a compound, then filtering the numerous possible descriptors with a genetic algorithm approach to ensemble feature selection. We tested the utility of our approach by modeling the acute aquatic toxicity (LC_{50}) of a congeneric set of 69 benzene derivatives. Our results demonstrate the ability of our approach to accurately predict toxicity directly from structure. Thus our new algorithm further increases the applicability of computer models to the problem of predicting chemical activity directly from its structure.

Acknowledgements

This work was partially supported by National Science Foundation grant IRI-9734419, a University of Montana MONTS grant, U.S. Air Force grant F49620-96-1-0330, and is contribution number 246 for the Center for Water and the Environment of the Natural Resources Research Institute.

References

Alpaydin, E. 1993. Multiple networks for function learning. In *Proceedings of the 1993 IEEE International Conference on Neural Networks*, volume I, 27–32. San Fransisco: IEEE Press.

Balaban, A. 1983. Topological indices based on topological distances in molecular graphs. *Pure and Appl. Chem.* 55:199–206.

Basak, S., and Grunwald, G. 1995. Estimation of lipophilicity from molecular structural similarity. *New Journal of Chemistry* 19:231–237.

Basak, S., and Magnuson, V. 1988. Determining structural similarity of chemicals using graph-theoretic indices. *Discrete Appl. Math.* 19:17–44.

Basak, S.; Harriss, D.; and Magnuson, V. 1988. Polly 2.3. Copyright of the University of Minnesota.

Bauer, E., and Kohavi, R. 1998. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning.*

Bondi, A. 1964. Van der waals volumes and radii. *J. Phys. Chem.* 68:441–451.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

CAS. 1999. The latest cas registry number and substance count. http://www.cas.org/cgi-bin/regreport.pl.

Cramer, C.; Famini, G.; and Lowrey, A. 1993. Use of calculated quantum chemical properties as surrogates for solvatochromic parameters in structure-activity relationships. *Acc. Chemical Research* 26:599–605.

de Waterbeemd, H. V. 1995. Discriminant analysis for activity prediction. In *Chemometric Methods in Molecular Design*, 283–294. VCH Publishers, Inc.

Dearden, J. 1990. Physico-chemical descriptors. In *Environmental Chemistry and Toxicology*, 25–59. Kluwer Academic Publisher.

Freund, Y., and Schapire, R. 1996. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 148–156. Morgan Kaufmann.

Gute, B., and Basak, S. 1997. Predicting acute toxicity (LC50) of benzen derivatives using theoretical molecular descripors: A hierarchical QSAR approach. *SAR and QSAR in Environmental Research* 7:117–131.

Gute, B.; Grunwald, G.; and Basak, S. In press. Prediction of the dermal penetration of polycyclic aromatic hydrocarbons (PAHs): A hierarchical QSAR approach. In *SAR and QSAR in Environmental Research*.

Hall, L.; Kier, L.; and Phipps, G. 1984. Structure-activity relationship studies on the toxicities of benzene derivatives: I. an additivity model. *Environ. Toxicol. Chem.* 3:355–365.

Hansch, C., and Leo, A. 1995. Exploring QSAR: Fundamentals and applications in chemistry and biology. *American Chemical Society* 557.

Hansch, C. 1976. On the structure of medicinal chemistry. *Journal of Medicinal Chemistry* 19:1–6.

Hansen, L., and Salamon, P. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12:993–1001.

Johnson, J. 1998. Pact triggers tests: Thousands of chemicals may be tested under toxicity screening program. *Chemical Engineering News* 76(44):19–20.

Kier, L., and Hall, L. 1986. *Molecular Connectivity in Structure-Activity Analysis*. Hertfordshire, UK: Research Studies Press.

Krogh, A., and Vedelsby, J. 1995. Neural network ensembles, cross validation, and active learning. In Tesauro, G.; Touretzky, D.; and Leen, T., eds., *Advances in Neural Information Processing Systems*, volume 7, 231–238. Cambridge, MA: MIT Press.

Maclin, R., and Opitz, D. 1997. An empirical evaluation of bagging and boosting. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 546–551. Providence, RI: AAAI/MIT Press.

Maclin, R., and Shavlik, J. 1995. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*.

Menzel, D. 1995. Extrapolating the future: research trends in modeling. *Toxicology Letters* 79:299–303.

Opitz, D., and Shavlik, J. 1996a. Actively searching for an effective neural-network ensemble. *Connection Science* 8(3/4):337–353.

Opitz, D., and Shavlik, J. 1996b. Generating accurate and diverse members of a neural-network ensemble. In Touretsky, D.; Mozer, M.; and Hasselmo, M., eds., *Advances in Neural Information Processing Systems*, volume 8. Cambridge, MA: MIT Press.

Opitz, D., and Shavlik, J. 1997. Connectionist theory refinement: Searching for good network topologies. *Journal of Artificial Intelligence Research* 6:177–209.

Opitz, D. 1999. Feature selection for ensembles. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

Quinlan, J. R. 1996. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 725–730. AAAI/MIT Press.

Randic, M. 1975. On characterization of molecular branching. *Journal of American Chemical Society* 97:6609–6615.

SAS. 1998. Cary, NC: SAS Institute Inc. chapter SAS/STAT User's Guide, Release 6.03 Edition.

Shapire, R.; Freund, Y.; Bartlett, P.; and Lee, W. 1997. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 322–330. Nashville, TN: Morgan Kaufmann.

Stewart, J. 1990. Mopac version 6.00. qcpe #455. US Air Force Academy, CO: Frank J. Seiler Research Laboratory.

SYBYL. 1998. Sybyl version 6.1. Tripos Associates, Inc.

Wiener, H. 1947. Structural determination of paraffin boiling points. *Journal of Am. Chem. Soc.* 69:17–20.