Virtual Screening Using Binary Kernel Discrimination: Analysis of Pesticide Data

David J. Wilton,[†] Robert F. Harrison,[‡] and Peter Willett*[,†]

Departments of Information Studies and of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S10 2TN, U.K.

John Delaney, Kevin Lawson, and Graham Mullier

Syngenta, Jealott's Hill International Research Centre, Bracknell, Berkshire RG42 6EY, U.K.

Received September 12, 2005

This paper discusses the use of binary kernel discrimination (BKD) for identifying potential active compounds in lead-discovery programs. BKD was compared with established virtual screening methods in a series of experiments using pesticide data from the Syngenta corporate database. It was found to be superior to methods based on similarity searching and substructural analysis but inferior to a support vector machine. Similar conclusions resulted from application of the methods to a pesticide data set for which categorical activity data were available.

VIRTUAL SCREENING METHODS

Virtual screening, i.e., the prediction of the biological activities of a set of chemical compounds prior to biological testing, is widely used to increase the cost-effectiveness of lead-discovery programs.[1−3] Many different approaches to virtual screening have been described, including similarity searching,[4] 3D substructure search,[5] and docking.[6] Here, we consider approaches that can be used when heterogeneous sets of known-active and known-inactive compounds are available, making it possible to derive a qualitative model linking structure to the biological activity of interest.[7] In a previous study, we compared several machine-learning methods for this purpose[8] and found, in experiments using 2D fingerprints, that the *binary kernel discrimination* (hereafter BKD) method[9] gave favorable results when compared to similarity[4] and substructural analysis[10,11] methods. In this paper, we report further studies of the effectiveness of BKD when used for virtual screening, focusing on the comparison of BKD with another machine-learning method, the *support vector machine* (SVM).[12,13]

The virtual screening methods considered here take a training set of compounds, each of which is known to be either biologically active or inactive. The training set is used to generate a model that enables a score to be computed for each molecule, *j*, in the test set for which predictions are required. These scores are used to rank the test set compounds, so that only the highest ranking compounds, those with the highest probabilities of activity, are submitted to the bioassay for testing.

We have used the four types of virtual screening method that were described in our previous report.[8] The similarity and substructural analysis methods are long-established in chemoinformatics, and we hence describe them only briefly here. The SVM method for machine learning is of rather more recent provenance but has already been quite widely used so we again provide only a relatively brief description. The BKD machine learning method has been used far less in the chemoinformatics context to date, and we have hence described the use of this for virtual screening in more detail.

Similarity and Substructural Analysis Methods. Similarity methods are well established for virtual screening and other chemical applications.[4] The similarity of a pair of compounds *i* and *j* can be computed from a number of different similarity coefficients. One of the most commonly used is the Tanimoto coefficient: this is defined for binary fingerprints as

$$S(i,j) = \frac{a}{a+b+c}$$

where *b* and *c* are the numbers of bits set to on in *i* or in *j* but not set to on in the other, and *a* is the number of bits set to on in both *i* and *j*. The similarities between compound *j* and the compounds of a training set $\{i\}$ may be combined in different ways.[8] For example, the score by which a compound is ranked may be taken to be the similarity to the most similar training set active, i.e.

$$S_{\max}(j) = \max\{S(i,j)\}\ i \in \text{actives}$$

This is a simple approach that has been demonstrated to be effective for similarity searching when multiple active reference structures are available.[14,15] Alternatively, if information about both active and inactive molecules is available in the training set, then we may use a function such as the mean similarity to all the training set inactives subtracted from the mean similarity to all the training set actives, i.e.

$$S_{\text{A}-\text{I}}(j) = \frac{1}{N_{\text{A}}} \sum_{i \in \text{actives}} S(i,j) - \frac{1}{N_{\text{I}}} \sum_{i \in \text{inactives}} S(i,j)$$

where there are N_{A} actives and N_{I} inactives in the training set.

* Corresponding author phone: 0044-114-2222633; fax: 0044-114-2780300; e-mail: p.willett@sheffield.ac.uk.

[†] Department of Information Studies, University of Sheffield.

[‡] Department of Automatic Control and Systems Engineering, University of Sheffield.

10.1021/ci050397w CCC: $33.50
Published on Web 02/21/2006

Table 1. Weighting Schemes for Substructural Analysis[a]

weighting scheme	weight
AVID	$(A_k + 1)\Big/\left(\frac{T_k N_A}{N_T} + 1\right)$
R1	$\log\left(\frac{A_k/N_A}{T_k/N_T}\right)$
R2	$\log\left(\frac{A_k/N_A}{I_k/N_I}\right)$
WT2	$\frac{A_k - I_k}{T_k}$
MAS	$\frac{1}{T_k}\left(A_k - \frac{T_k N_A}{T_k}\right)$

[a] In this table, A_k and I_k are the numbers of active and inactive training set compounds with bit k set, T_k is the total number of compounds with bit k set (i.e., $A_k + I_k$), N_A and N_I are the total numbers of training set active and inactive compounds, and N_T is the total number of training set compounds (i.e., $N_A + N_I$).

Substructural analysis (or SSA) was first described by Cramer et al.[10] and involves computing weights for each of the bits in the fingerprints that are used to characterize the training set compounds. Those bits which occur in more actives than inactives are assigned high weights, and those that occur in more inactives than actives are assigned low weights. A number of different weighting schemes exist.[11] For a compound of unknown activity we calculate a score by summing the weights of the bits present in that compound; some weighting schemes also involve a normalization based on the number of bits that are set to on. The definitions of the weighting schemes we have used are given in Table 1.

Support Vector Machines. The SVM has rapidly established itself as one of the most powerful methods for machine learning and is becoming an increasingly popular tool for solving chemical classification problems.[16−22] Detailed discussions of the SVM approach are provided by Burges[12] and by Christianini and Shawe-Taylor;[13] here, we briefly outline how an SVM can be applied to virtual screening.

A set of two-category data represented in an M-dimensional space (such as a training set of active and inactive compounds represented by M-length binary fingerprints) can in principle be transformed into a N-dimensional space ($N \geq M$), where the two categories become linearly separated, i.e., each category lies on a different side of an $(N-1)$-dimensional hyperplane. In fact, there may be many such hyperplanes that separate the two categories of data: an SVM finds the "best" separating hyperplane, this being the one which maximizes its distance from both categories. However, because it is necessary to train the SVM on a finite set of data, it is rare that a transformation that separates the data perfectly will generalize to novel data, which is the purpose of the classifier. Instead, a "soft margin" SVM is usually preferred. This permits a number of errors to be made on the training sample to avoid overspecialization and finds a hyperplane that implements a tradeoff between accuracy and error rate. This is done, in practice, via a cross validatory choice of parameters.

It is not necessary to know the explicit transformation into the high, possibly infinite, dimensional space that the SVM operates in. If $\Phi(\boldsymbol{x})$ and $\Phi(\boldsymbol{y})$ are the transformations of data points $\boldsymbol{x}$ and $\boldsymbol{y}$, then all required calculations can be cast in terms of dot products between these, i.e., $\Phi(\boldsymbol{x})\cdot\Phi(\boldsymbol{y})$. There are a number of kernel functions $K(\boldsymbol{x},\boldsymbol{y})$, defined in terms of the original M-dimensional space, such that $K(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{x})\cdot\Phi(\boldsymbol{y})$; thus the direct computation of the dot products and explicit knowledge of the transformation can be avoided by replacing them with a suitable kernel function.

Once the SVM has learned how to separate adequately the training set data, the corresponding transformation can then be applied to new data, with predictions for individual molecules being based on which side of this hyperplane they are found to lie. If a ranking, rather than a classification, of the test set is required, then the distance from the hyperplane to the current (transformed) sample can be used as a score for ranking, with compounds on the inactive side being assigned a negative distance (so that those test set compounds on the active side furthest away from the hyperplane will be ranked highest).

Binary Kernel Discrimination. BKD was first used in the chemoinformatics domain by Harper et al.[9] If d_{ij} is the squared Euclidean distance between compounds i and j (which for binary data is also the Hamming distance, i.e., the number of bit positions in which the compounds' fingerprints differ), and λ is a smoothing parameter to be determined, then the kernel similarity function suggested by Aitchison and Aitken[23] is

$$K_\lambda(i,j) = \lambda^{n-d_{ij}}(1-\lambda)^{d_{ij}} \tag{1}$$

where n is the length of the binary fingerprints. This function can then be used in kernel density estimators to estimate the likelihoods that a compound is active or not. Following Harper et al., we have used a scoring function proportional to the ratio of these estimators—the likelihood ratio for activity

$$S_{BKD}(j) = \frac{\sum_{i\in\text{active}} K_\lambda(i,j)}{\sum_{i\in\text{inactive}} K_\lambda(i,j)} \tag{2}$$

to rank the molecules in the test set. The optimum value of λ is determined from the training set by cross-validation. This is done by computing scores for each training set compound using a number of different values of λ between 0.5 and 1.0, the summations in this case running over all other training set compounds. These scores are then used to rank the training set for each value of λ, and that value which gives the lowest sum of ranks for the actives (i.e., the value that gives the greatest clustering of the actives at the top of the ranking) is taken to be the optimum λ. It is assumed that the optimal value for the training set is also optimal for the ranked test set: this is clearly a strong assumption, but the results we have obtained[8,15] suggest that it does not result in poor predictive performance (and it is, of course, difficult to use a machine learning technique without such an assumption). The length of the binary fingerprints, n, is typically in the range 100−1000 or more, which results in very small values of the kernel functions, especially when λ is close to 0.5. Therefore, if the true optimum λ is close to 0.5 it may not be possible to find this value, as many of the kernel values may be below the tolerance of the machine used to perform the calculation. However a modified version

Table 2. Distributions of Values of the Kernel Function K for Five Different Test Compounds with a Set of 400 Other Compounds Using the Same Value of λ

test set compd	number of compounds with $K =$									
	10^{-19}	10^{-20}	10^{-21}	10^{-22}	10^{-23}	10^{-24}	10^{-25}	10^{-26}	10^{-27}	10^{-28}
A	0	2	15	67	90	104	78	36	6	2
B	0	0	2	5	34	100	126	90	40	3
C	0	0	0	0	1	6	93	181	104	15
D	1	2	3	25	99	99	109	47	13	2
E	0	2	11	4	10	25	63	172	105	8

of eq 1 may be used in such cases, as described by Harper[24]

$$K_\lambda(i,j) = [\lambda^{n-d_{ij}}(1-\lambda)^{d_{ij}}]^{(k/n)} \quad (3)$$

where k ($k \leq n$) is a user-defined constant.

BKD is, in essence, a similarity method that combines the dis-similarities between a single test set compound and the training set compounds to produce a score indicating the likelihood of that test set compound being active. The similarity measure used is a kernel function, as given in (1) or (3), which incorporates the squared Euclidean distance between two compounds described by binary fingerprints. The score for the test set compound is then just a summation of the kernel similarities with the training set actives divided by the summation of the kernel similarities with the training set inactives, as shown by eq 2.

The variation of the kernel function $K_\lambda(i,j)$ with respect to squared Euclidean distance and λ is shown in Figure 1, where it will be seen that the difference in magnitude between low and high values of $K_\lambda(i,j)$ increases as λ increases. The scores computed by BKD are derived from summing many different kernel similarities: as λ increases, the BKD scores are thus increasingly dominated by the largest kernel values (i.e., the most similar training set actives and inactives). Exactly how many terms in the summations, and hence how many training set compounds, make significant contributions to the scores is determined by the value of λ, so that, in essence, BKD finds the optimum number of training set nearest neighbors on which to base an activity prediction. This optimum number is not necessarily the same for each test set compound (whereas conventional nearest neighbor approaches generally use some fixed number of nearest neighbors). This characteristic is illustrated by Table 2, which shows the distribution of kernel-function values for five different test set compounds with 400 training set compounds calculated using the same value of λ. From this table, it will be seen that the sums of the kernels for each of these compounds may be written as

$$\text{sum(A)} = 2 \times 10^{-20} + 15 \times 10^{-21} + 67 \times 10^{-22} + \ldots;$$

$$\text{sum(B)} = 2 \times 10^{-21} + 5 \times 10^{-22} + 34 \times 10^{-23} + \ldots\text{etc., etc.}$$

These sums may be truncated after a few terms as the smaller terms contribute negligibly to the total. To a reasonable approximation, therefore, we can see that the number of kernels that contribute to the sums, and hence (taking the first three nonzero terms) the number of training set compounds that contribute to the scores, are 84, 41, 100, 6, and 17 for compounds A, B, C, D, and E, respectively.

COMPARISON OF METHODS

Ranking Experiment. Previous studies of BKD (and, indeed, of many of the available virtual screening methods) have used pharmaceutical data sets, many of which are publicly available. Thus, the original description of BKD by Harper et al.[9] used the well-known NCI AIDS and MAO data sets and an internal HTS enzyme-assay data set, while most of our previous experiments involved NCI AIDS and *MDL Drug Data Report* (MDDR) data.[8,15] The work reported here has used agrochemical data that has been generated in corporate pesticide discovery programs. Specifically, the first set of experiments used a Syngenta data set of 132 784 compounds that had been tested in 15 different in vivo whole organism screens and that were characterized by the 988-bit, UNITY 2D fingerprints.[25] The 7127 compounds from this data set were active in at least one screen, with the remaining 125 657 compounds having a response less than a predefined threshold value in all of the screens. The 7127 actives may be further classified as insecticides, herbicides, and fungicides, of which there were 1952, 2973, and 2778, respectively (with some compounds occurring in more than one category). In our previous paper,[8] we reported results using training sets of 713 actives (corresponding to about 10% of the total) randomly selected from all activity categories and 713 randomly selected inactives, with the remainder of the data used as the test set. We hence assessed the various virtual-screening methods on their ability to produce a model of general pesticide activity; here, we extend our analyses by focusing on the individual types of pesticidal activity, by considering the effect of training set size, and by analyzing the SVM method in more detail.

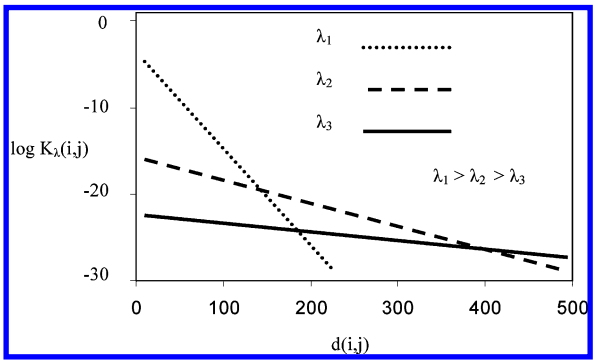

Figure 1. Variation of the kernel function, $K_\lambda(i,j)$ in eq 3, for typical values of the squared Euclidean distance, $d(i,j)$, using 988-bit binary fingerprints and $k = 100$.

A number of training sets were randomly selected using only one category of pesticide as the actives, therefore producing separate models for insecticide activity, herbicide activity, and fungicide activity. In each case the number of training set actives corresponded to 10% of the total, with the same number of inactives; the remaining actives of the same category and all of the remaining inactives comprised the test set. Six different random partitions of the data into a training set and test set were carried out for each category of pesticide.

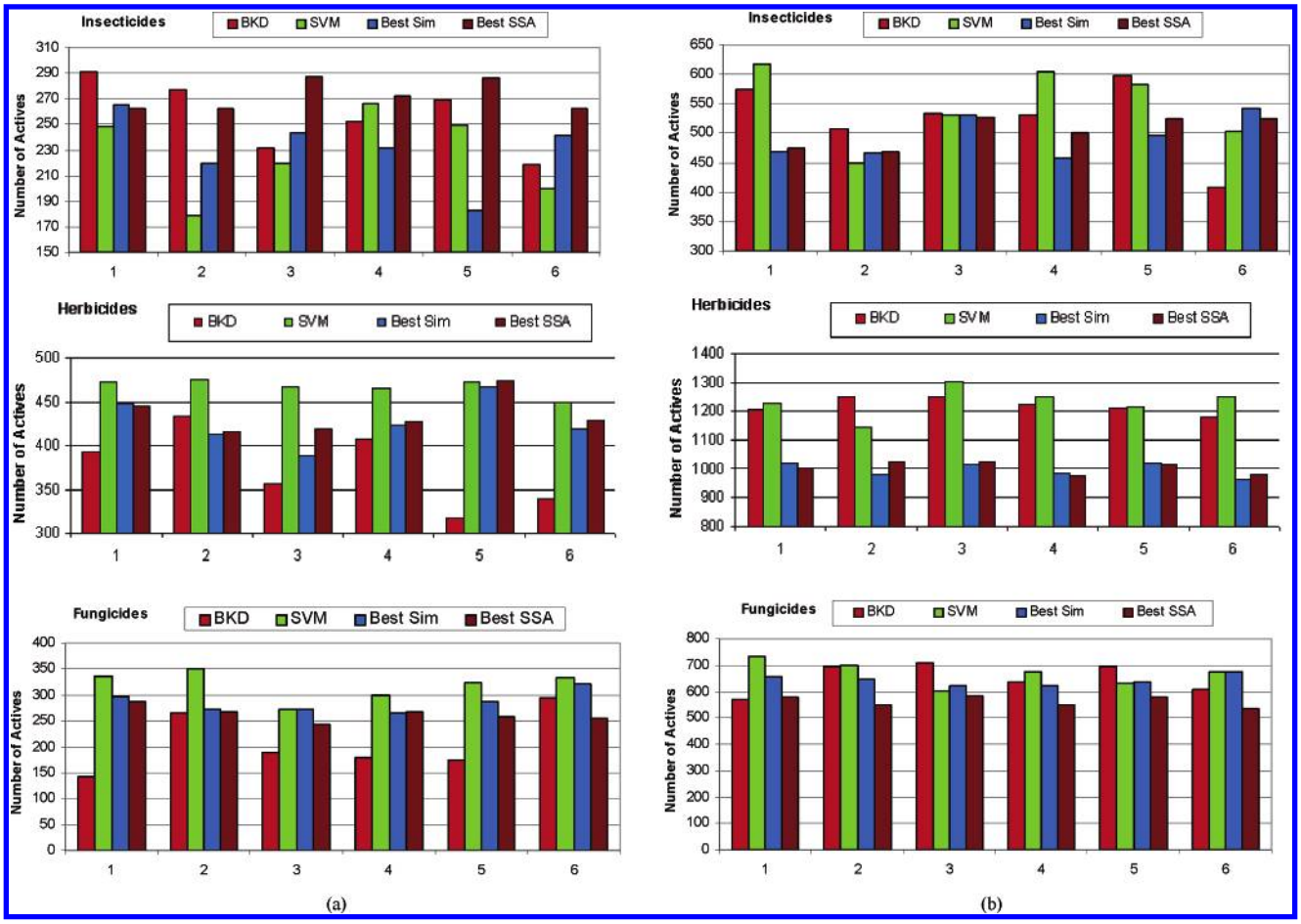

Figure 2. Numbers of active compounds found in (a) the top 1% and (b) the top 5% of the ranked test sets, using six different test sets and training sets for each type of pesticide.

The six test sets were ranked by each of the virtual screening methods. Our BKD rankings used the form of the kernel function given in eq 3 with $k = 100$, with the optimum values of λ being found to be in the range 0.55−0.60. The SVM results were obtained using the program SVM^{light}.[26] Initial experiments with the standard SVM^{light} radial basis function and polynomial kernels showed that the latter gave better results in terms of ranking the test set actives. This polynomial kernel has the general form $(a\,\boldsymbol{x}\cdot\boldsymbol{y} + b)^c$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the binary fingerprints representing data set compounds; detailed comparative experiments were carried out with the best results being obtained using $a = b = 0.1$ and $c = 5$, with all the other settings for the SVM being the SVM^{light} defaults.

The results of these experiments are shown in Figure 2, which shows the numbers of actives retrieved in the top 1% or the top 5% of the ranked test set; results are presented for binary kernel discrimination (BKD), support vector machine (SVM), the best similarity method for that test set (Best Sim), and the best substructural analysis method for that test set (Best SSA). The best similarity method was always S_{A-I} for the insecticide and herbicide data sets, but was the S_{max} method in all but one of the fungicide data sets. There was more variation in the best SSA weighting scheme: AVID and WT2 tended to be best with the insecticides and herbicides in the top 5% experiments, whereas R2 was always the best for the fungicides, while the results were still more varied in the top 1% experiments.

Analysis of the 18 top 1% runs (three types of pesticidal activity and six data set partitions for each activity) shows that BKD performs quite erratically with respect to the similarity and substructural analysis methods and that while SVM is the best or equal-best method for all the herbicide and fungicide results it is always the worst for the insecticides. However, analysis of the 18 top 5% runs shows that the two machine-learning methods perform far better: BKD outperforms the best similarity and substructural analysis in 16 of the 18 runs; SVM outperforms the best similarity in 13 and outperforms the best substructural analysis in 16 of the 18 runs.

When several ranking methods are available, data fusion[27] can be used to yield a combined ranking that is superior to individual rankings.[28] We hence investigated the fusion of some of the BKD and SVM rankings using the SUM and MAX fusion rules: with the SUM rule, test set compounds are reranked using the sum of the BKD and SVM ranks; while with the MAX rule, the compounds are reranked using the better of the BKD and SVM ranks. Such approaches have been found to be effective in previous work,[21] but the results here were inconclusive, and where there were improvements these were quite small.

The initial experiments showed that the BKD and SVM methods could perform effectively even with a very limited amount of training data. However, the experiments were rather unrealistic in two ways. First, in real virtual screening programs, there may often be a large amount of such data

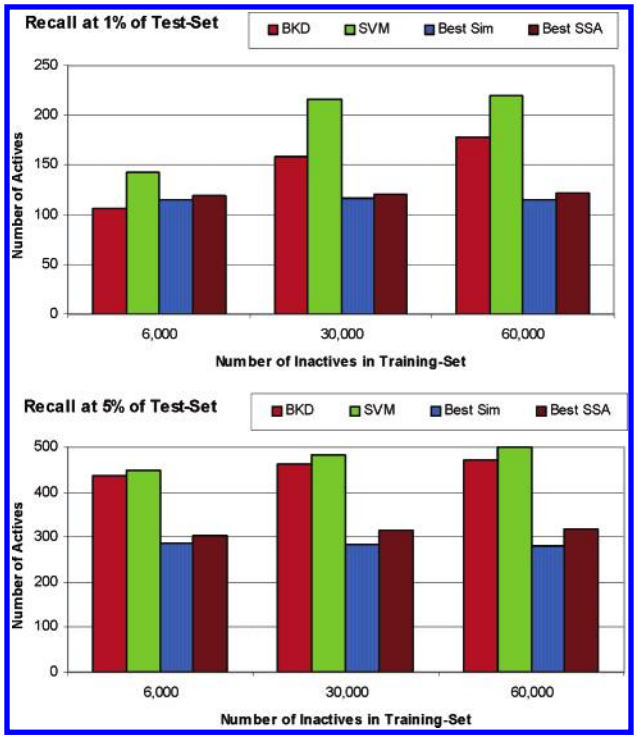

Figure 3. Numbers of active compounds found in the top 1% and top 5% of the test set ranked using three different training sets, each containing 6000 actives compounds and either 6000 or 30 000 or 60 000 inactive compounds.

already available from previous testing; second, in such cases, there will normally be many more inactive compounds than active compounds available for training. The second set of experiments hence studied the effect of using much larger training sets and of increasing the ratio of inactives to actives in these sets. Specifically, training sets of 6000 actives, randomly selected from all pesticide categories, were used with between 6000 and 60 000 randomly selected inactives, with the larger training sets always including the inactives that were in the smaller training sets. For each training set exactly the same test set of the 1127 remaining actives and 65 657 inactives was then used, to ensure that any variations in the results did not arise from differences in the test sets.

Each virtual screening method was employed as described previously. The optimum value of λ used for the BKD method was 0.64 for each training set. Note that due to the time that would be required, no optimization was carried out for the 6000 active/60 000 inactive training set: having found an optimum λ of 0.64 for the 6000 active/6000 inactive and 6000 active/30 000 inactive training sets, this value was assumed to be at least near-optimal for the largest training set.

Results for the number of actives found in the top 1% and 5% of the test set when ranked by each method are shown in Figure 3. We have again shown only the best similarity and substructural analysis method results: for the similarity results this was S_{A-I} in all cases, while for substructural analysis this was either WT2 or AVID. As in the previous experiments, BKD does not find as many actives as similarity and substructural analysis at 1% recall with equal numbers of training set actives and inactives (i.e. 6000/6000), and SVM is the best method. However at 5% recall and with larger numbers of training set inactives, both BKD and SVM significantly outperform the other methods (and the same results are also obtained if one considers the top 10% of the rankings). Therefore, BKD and SVM improve much more than the other methods as more inactive compounds are included and as more training set information becomes available. It is worth repeating here that BKD and SVM are being compared with the best similarity and substructural analysis methods. A particularly stiff basis of comparison has thus been set for the two machine-learning methods, in that they have been parametrized using just the training set data but are being compared here with the methods that proved to be most effective on the test set data. Moreover, their relative performance is more consistent than is the case with the other two approaches, where there is some degree of variation in the method that performs best.

Classification Experiment. Thus far, both in this paper and in the work reported in ref 8, we have considered the use of the various methods to produce a ranking of a test set in order of decreasing likelihood ratio for activity. We now consider the application of the BKD method to categorization, i.e., predicting the category membership for each member of the test set, rather than ranking.[29] This second set of experiments involved a further Syngenta data set containing 3792 compounds for which soil decomposition rates had been determined and which were again characterized by UNITY 2D fingerprints.

Each compound had been categorized as having a decomposition rate that was very short (VS), short (S), medium (M), long (L), or very long (VL), and the experiments sought to predict to which each of the five categories a particular compound belonged. The data set was randomly divided into a training set of 792 compounds, of which 158 were VS, 158 S, 160 M, 158 L, and 158 VL, leaving a test set of 3000 compounds. We compared BKD and SVM to a simple nearest neighbor classifier, in which a test set compound was predicted to be in the same category as its single nearest neighbor (NN) in the training set.

For the BKD approach we produced a model for each of the five categories. If C denotes the current category of interest, then we set the actives in eq 2 to be those molecules in the training set that occur in category C, and the inactives in eq 2 to be those molecules in the training set that occur in any of the other four categories. A compound is then predicted to be in the category of the model that gives it the highest score. This requires that the same value of λ is used for each model as the magnitude of λ can greatly affect the resulting scores (as discussed previously). The optimum value for λ was derived by computing scores for each training set compound, as before, using different values of λ. Then, for a particular value of λ, the category of each training set compound was predicted, and the optimum λ was taken to be that which yielded the most correct predictions.

The SVM approach here was similar to that for BKD in that we constructed a model for each category of compound. This was done by labeling training set compounds of a particular category as actives and those of any other category as inactives and training an SVM (implemented as previously described) to distinguish between the two categories. This process was repeated for each of the five categories. The five resulting models were then used to produce scores for each test set compound, with each such compound being predicted to be in the category for which the model gave it the highest rank. Ranks rather than scores were used in this

Table 3. Number of Correct Predictions by the NN, BKD, and SVM Methods for the Soil-Decomposition Categorization Experiments

	correct predictions		
run	NN	BKD	SVM
1	1059	1157	1189
2	1032	1125	1135
3	1037	1086	1152

Table 4. Predictions Made by NN, BKD, and SVM Methods for Five-Category Soil-Decomposition Data[a] Using the Second Random Partition of Data into Training Set and Test Set

predicted categories		percentage of actual categories in each predicted category				
		VS	S	M	L	VL
NN	VS	42.1	19.2	9.6	7.1	6.7
	S	23.4	30.8	16.6	11.9	8.0
	M	14.5	20.9	29.2	21.6	20.1
	L	9.3	18.0	26.0	30.9	24.3
	VL	10.7	11.1	18.6	28.6	40.8
BKD	VS	54.7	23.5	13.9	8.2	11.4
	S	19.0	34.4	19.4	13.8	8.9
	M	11.4	17.5	26.6	19.7	14.3
	L	6.9	15.7	24.8	32.0	23.2
	VL	8.1	8.9	15.3	26.4	42.1
SVM	VS	60.4	27.1	16.7	10.8	11.1
	S	16.4	30.4	14.4	7.8	4.7
	M	8.6	16.8	26.7	21.2	15.1
	L	6.0	14.5	23.7	30.5	24.5
	VL	8.4	11.2	18.5	29.4	44.6

[a] VS = very short, S = short, M = medium, L = long, and VL = very long.

case as the scores for different SVM models correspond to the distances from different hyperplanes in different spaces and are, therefore, not directly comparable.

Using a training set composed of the previously stated numbers of compounds from each category, the categories of the test set compounds were predicted, and then the accuracies of the predictions using the NN, BKD, and SVM methods were compared. The experiment was run three times with different random partitions of the data into training and test sets, but with the proportion of compounds from each category remaining the same. In each run the SVM method resulted in more correct predictions than BKD, which in turn resulted in more than NN, as exemplified in Table 3.

The results of these predictions can be further analyzed by what proportion of each actual category is found in each predicted category. This is shown for the second of the three experiments in Table 4; similar comments also apply to the other two runs. There are marked differences in performance for the VS compounds (and, to a much less extent, the VL compounds), with SVM performing best, then BKD, and finally NN. The results for the intermediate categories are less well-marked; these results also vary from test set to test set to some extent, with BKD always being best for S, NN always best for M, and none of them being consistently best for L. It is also possible to consider not just those compounds correctly predicted but also those with a predicted category one higher or one lower than the correct one. Doing this, between 60% and 80% of test set compounds have such a prediction for all methods, and SVM is still better than NN and BKD for the VS and VL categories. Overall, therefore, the SVM method is best at placing compounds into the two extreme categories, VS and VL, but BKD is normally better for the nonextreme categories. NN is best overall for the M category, but all of the methods predict less than 30% of this category correctly.

It is not really surprising that all the methods work best for the extreme categories. Taking SVM as an example, the rankings produced by this method are derived from a training procedure that learns to classify compounds as active or inactive. For the extreme category models it is quite reasonable to consider one extreme range of a property value as active and all other higher/lower values as inactive. For the S, M, and L categories, however, training set compounds from different ends of the range of values are being grouped together as inactives. In these cases, therefore, the SVM will try and learn a model that groups quite different compounds together in the inactive class, which is clearly a much more difficult problem than when attention can be focused on a particular extreme category.

CONCLUSIONS

Machine learning methods are being increasingly used for the analysis of chemical data sets; here, we have studied the use of binary kernel discrimination (BKD) for virtual screening of such databases and compared the results obtained with other methods based on support vector machines (SVM), similarity searching, and substructural analysis. Simulated virtual-screening experiments with corporate pesticide data show that BKD performs well, particularly when large amounts of training set data are available (as is often the case in a corporate environment where extensive HTS bioactivity data are available), and we have also demonstrated that BKD can be applied to categorical data. However, the results in both cases are generally inferior to those obtained with an SVM approach based on the widely available SVM^{light} software.

ACKNOWLEDGMENT

We thank Syngenta for funding, Gavin Harper for advice, the referees for their comments on an earlier version of this paper, and the Royal Society, Tripos Inc., and the Wolfson Foundation for hardware, laboratory, and software support.

REFERENCES AND NOTES

(1) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual Screening − an Overview. *Drug Discovery Today* **1998**, *3*, 160−178.
(2) *Virtual Screening for Bioactive Molecules*; Bohm, H.-J., Schneider, G., Eds.; Wiley-VCH: Weinheim, 2000.
(3) *Virtual Screening: an Alternative or Complement to High Throughput Screening*; Klebe, G. Ed.; Kluwer: Dordrecht, 2000.
(4) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(5) *Pharmacophore Perception, Development and Use in Drug Design*; Guner, O. Ed.; International University Line: La Jolla, CA, 2000.
(6) Halperin, I.; Ma, B.; Wolfson, H.; Nussinov, R. Principles of Docking: An Overview of Search Algorithms and a Guide to Scoring Functions. *Proteins* **2002**, *47*, 409−443.
(7) Gedeck, P.; Willett, P. Visual and Computational Analysis of Structure−Activity Relationships in High-Throughput Screening Data. *Curr. Opin. Chem. Biol.* **2001**, *5*, 389−395.
(8) Wilton, D.; Willett, P.; Lawson, K.; Mullier, G. Comparison of Ranking Methods for Virtual Screening in Lead-Discovery Programs. *J. Chem. Inf. Comput. Sci.* **2003,** *43*, 469−474.
(9) Harper, G.; Bradshaw, J.; Gittins, J. C.; Green, D. V. S.; Leach, A. R. Prediction of Biological Activity for High-Throughput Screening Using Binary Kernel Discrimination. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1295−1300.

(10) Cramer, R. D.; Redl, G.; Berkoff, C. E. Substructural Analysis. A Novel Approach to the Problem of Drug Design. *J. Med. Chem*. **1974**, *17*, 533−535.
(11) Ormerod, A.; Willett, P.; Bawden, D. Comparison of Fragment Weighting Schemes for Substructural Analysis. *Quant. Struct.-Act. Relat.* **1989**, *8*, 115−129.
(12) Burges, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Know. Discovery* **1998,** *2*, 121−167.
(13) Christianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning* Methods; Cambridge Universtity Press: 2000.
(14) Schuffenhauer, A.; Floersheim, P.; Acklin, P.; Jacoby, E. Similarity Metrics for Ligands Reflecting the Similarity of the Target Proteins. *J. Chem. Inf. Comput. Sci*. **2003**, *43*, 391−405.
(15) Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. Comparison of Fingerprint-Based Methods for Virtual Screening Using Multiple Bioactive Reference Structures. *J. Chem. Inf. Comput. Sci.* **2004,** *44*, 1177−1185.
(16) Burbridge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Comput. Chem.* **2001**, *26*, 5−14.
(17) Yu-Dong, C.; Xiao-Jun, L.; Xue-Biao, X.; Kuo-Chen, C. Support Vector Machines for Predicting HIV Protease Cleavage Sites in Protein. *J. Comput. Chem.* **2002,** *23*, 267−274.
(18) Warmuth, M. K.; Liao, J.; Rätsch, G.; Mathieson, M.; Putta, S.; Lemmem, C. Active Learning with Support Vector Machines in the Drug Discovery Process. *J. Chem. Inf. Comput. Sci.* **2003,** *43*, 667−673.
(19) Anderson, D. C.; Weiqun, L.; Payan, D. G.; Noble, W. S. A New Algorithm for the Evaluation of Shotgun Peptide Sequencing in Proteomics: Support Vector Machine Classification of Peptide MS/MS Spectra and SEQUEST Scores. *J. Proteome Res.* **2003,** *2*, 137−146.
(20) Sadik, O.; Land, W. H.; Wanekaya, A. K.; Uematsu, M.; Embrechts, M. J.; Wong, L.; Leibenspereger, D.; Volykin, A. Detection and Classification of Organophosphate Nerve Agent Simulants Using Support Vector Machines with Multisensor Arrays. *J. Chem. Inf. Comput. Sci.* **2004,** *44*, 499−507.
(21) Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases using a Support Vector Machine. *J. Chem. Inf. Model.* **2005**, *45*, 549−561.
(22) Saeh, J. C.; Lyne, P. D.; Takasaki, B. K.; Cosgrove, D. A. Lead Hopping Using SVM and 3D Pharmacophore Fingerprints. *J. Chem. Inf. Model*. **2005**, *45*, 1122−1133.
(23) Aitchison, J.; Aitken, C. G. G. Multivariate Binary Discrimination by the Kernel Method. *Biometrika* **1976,** *63*, 413−420.
(24) Harper, G. The Selection of Compounds for Screening in Pharmaceutical Research. Ph.D. Thesis, University of Oxford, 1999.
(25) The UNITY software is available from Tripos Inc. at http://www.tripos.com.
(26) Joachims, T. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods − Support Vector Learning*; Schölkopf, B., Burges, C., Smola, A., Ed.; MIT-Press: 1999; pp 41−56.
(27) Klein, L. A. Sensor and Data Fusion Concepts and Applications; *SPIE The International Society for Optical Engineering*, 2nd ed.; Bellingham, WA, 1999.
(28) Ginn, C. M. R.; Willett, P.; Bradshaw, J. Combination of Molecular Similarity Measures Using Data Fusion. *Perspect. Drug Discovery Des.* **2000**, *20*, 1−16.
(29) Hall, P. Biometrika Centenary: Nonparametrics. *Biometrika* **2001**, *88*, 143−165.

CI050397W