Ensemble Methods for Classification in Cheminformatics

Christian Merkwirth,*,‡ Harald Mauser,† Tanja Schulz-Gasch,† Olivier Roche,† Martin Stahl,† and Thomas Lengauer‡

Computational Biology & Applied Algorithmics Group, Max-Planck-Institut für Informatik, Stuhlsatzenhauseg 85, 66123 Saarbrücken, Germany, and Roche Pharma Research, Basel, Switzerland

Received May 4, 2004

We describe the application of ensemble methods to binary classification problems on two pharmaceutical compound data sets. Several variants of single and ensembles models of k-nearest neighbors classifiers, support vector machines (SVMs), and single ridge regression models are compared. All methods exhibit robust classification even when more features are given than observations. On two data sets dealing with specific properties of drug-like substances (cytochrome P450 inhibition and "Frequent Hitters", i.e., unspecific protein inhibition), we achieve classification rates above 90%. We are able to reduce the cross-validated misclassification rate for the Frequent Hitters problem by a factor of 2 compared to previous results obtained for the same data set with different modeling techniques.

1. INTRODUCTION

Ensemble methods have gained increasing attention over the past years, from simple averaging of individually trained neural networks[1] for regression problems over the combination of thousands of decision trees to *Random Forests*[2] to the boosting of weak classifiers where the training of each subsequent classifier depends on the results of all previously trained classifiers.[3]

In this study we concentrate on several binary classification problems with a large number of input variables (descriptors). Both input variables and binary output labels are related to properties of the chemical compound under consideration. One of these problems is to correctly identify *Frequent Hitters* (FH). Frequent Hitters are defined as molecules generating hits in many different assays covering a wide range of targets[4] and therefore are not specific enough to be suitable drug candidates.

Section 2 introduces the ensemble methods which we are using in this study. In section 3 we describe our approaches to feature reduction, while section 4 details the data sets on which we perform our study and the concrete classification procedures. In section 5 we discuss the results of the evaluation of the proposed methods.

Throughout this document, the words *variable*, *descriptor*, and *feature* will be used as synonyms. The same holds for *observation*, *sample*, and *data point*. A *model* denotes a classifier. An ensemble is a collection of models but can be treated as a model itself.

2. ENSEMBLE METHODS

Ensemble methods were already considered in combination with classification and regression problems arising in cheminformatics,[5−7] though with varying conclusions. With this study we would like to compare several ensemble approaches with single models and with linear models which are widely used in cheminformatics. Due to the very high number of possible combinations of ensemble techniques and underlying model types we chose a cross-section of interesting variants for this comparison. By this we want to give the reader hints as to what combination could be useful for a particular problem setting. We consider the following model types which are described in greater detail in the Appendix:

- k-Nearest Neighbor Classifiers (k-NN)
- Support Vector Machines (SVM)
- Linear regression with ridge penalty (Ridge Regression)

We did not construct ensembles of linear ridge regression models since we observed no noticeable benefit due to the very low computational variance of this model type. Also k-NN are used only in combination with preprocessing methods that reduce the actual number of variables beforehand since they are not able to cope directly with the very high number of variables given in some of the data sets investigated.

2.1. Ensemble Methods for Regression. For the regression setting, the combined output can either be a simple average $\bar{f}(\vec{x}) = 1/K\sum_{k=1}^{K} f_k(\vec{x})$ or a weighted average $\bar{f}(\vec{x}) = \sum_k w_k f_k(\vec{x})$ with $\sum_k w_k = 1$, where $f_k(\vec{x})$ denotes the output of the kth model for input vector $\vec{x}$. According to Krogh et al.[8] the following error decomposition of the pointwise ensemble error $e(\vec{x})$ can be done:

$$e(\vec{x}) = (y(\vec{x}) - \bar{f}(\vec{x}))^2$$

$$\bar{\epsilon}(\vec{x}) = \frac{1}{K}\sum_{k=1}^{K}(y(\vec{x}) - f_k(\vec{x}))^2$$

$$\bar{a}(\vec{x}) = \frac{1}{K}\sum_{k=1}^{K}(f_k(\vec{x}) - \bar{f}(\vec{x}))^2$$

$$e(\vec{x}) = \bar{\epsilon}(\vec{x}) - \bar{a}(\vec{x})$$

Here $\bar{\epsilon}(\vec{x})$ denotes the average error of the individual models $f_k(\vec{x})$, while $\bar{a}(\vec{x})$ is the variance of the individual models with

* Corresponding author phone: +49 681 9325 318; e-mail: {cmerk,lengauer}@mpi-sb.mpg.de.

† Molecular Structure and Design, F. Hoffmann-La Roche Ltd., CH-4070 Basel.

‡ Max-Planck-Institut für Innformatik.

10.1021/ci049850e CCC: $27.50
Published on Web 10/13/2004

respect to the average model $\bar{f}$. Integrating the pointwise ensemble error $e(\vec{x})$ over the input space yields an expression for the generalization error **E** of the ensemble model $\bar{f}$:

$$\mathbf{E} = \bar{\mathbf{E}} - \bar{\mathbf{A}} \tag{1}$$

According to eq 1, the ensemble generalization error **E** is always smaller than the average error $\bar{\mathbf{E}}$ of the individual models since $\bar{\mathbf{A}}$ cannot be negative. We can conclude that an ensemble should consist of several good (low $\bar{E}$) but mutually uncorrelated models (large $\bar{A}$). In practice, we often observed that an ensemble outperforms even the best of its constituting models in terms of generalization error for regression problems.

2.2. Ensemble Methods for Classification. In the case of binary classification the aim is to decide whether a sample belongs to a certain class or not. Methods such as SVM classifiers (see subsection A.2) accomplish this task by constructing a separating hyperplane. The position of a sample with regard to this hyperplane determines the predicted output label. Other methods use a more regression-like approach by approximating a continuous discriminant function, that, when properly scaled to [0, 1], can be interpreted as a confidence that a sample belongs to the class of interest.[9]

When ensembling binary classifiers by averaging the output of several individual models, considerations similar to those in subsection 2.1 hold for the classification error.[10] The output is here a continuous function which is compared to a threshold of 0.5 to discriminate between both classes. For dichotomies this procedure is equivalent to a majority vote.

2.3. Random Subspace Method. For the Random Subspace approach a large number (here 75) of individual models is trained on randomly chosen subsets of all available input variables. The number of variables in each subset is typically set to the square root of the total number of input features. The k-NN model type is well suited for this approach since the leave-one-out (LOO) error on the training data set can be easily obtained. We then discard all models having an LOO error worse than the median LOO error of all models constructed, resulting in an ensemble consisting of 38 models.

A strong advantage of this method is that training N classifiers on a D/N dimensional subset is faster than training one classifier on all D variables for any training algorithm with a time complexity higher than $O(D)$.

2.4. Out-of-Train Technique. The Out-of-Train (OOT) technique is a method for assessing the extra-sample error and can be regarded as a combination of traditional cross-validation (CV) and ensemble averaging. Like in traditional cross-validation, the data set is repeatedly divided into training and test partitions. For one given partitioning, a model is constructed only on samples of the training partition. Test samples are not used for model selection, deriving of stopping criteria or the like. The OOT output for one sample of the data set is the average of the outputs of models for which this sample was not part of the training set (out-of-train) as depicted in Figure 1. The OOT output can be used to compute estimates of the extra-sample error or extra-sample classification rate.

Unlike Breiman's Out-of-Bag (OOB) technique[11] which creates bootstrap replicates of the data set, the OOT technique does not allow samples to occur repeatedly in the training fraction of one model. This could impair some statistical learning algorithms such as the proposed k-NN algorithm (see subsection A.1). Similar to traditional CV, OOT tends to overestimate the generalization error due to the smaller size of each training partition. Unlike CV, it accounts for the ensemble gain (see eq 1) by averaging the outputs of several models. Displaying slightly overestimated error rates should not contradict the conservative approach of this study.

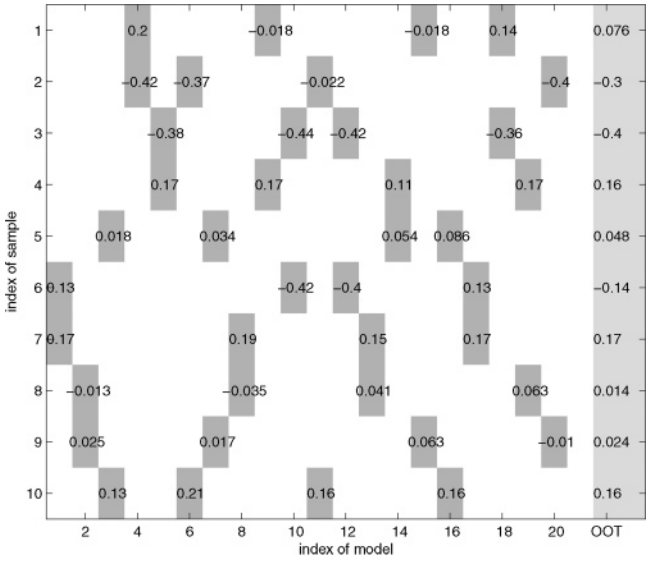

Figure 1. Averaging scheme for OOT calculation for an example data set of 10 samples. On this data set, 20 models were trained. Column j corresponds to model j. For each model, samples used for training are colored white, while samples not used for training are colored gray. For easier reading, only output values for test samples were printed on the respective row and column. To compute the OOT output (grey values in the rightmost column) for the ith sample, the average over the output of all models for which this sample was not in the training fraction is calculated (averaging over all gray fields in a row).

3. FEATURE REDUCTION

When analyzing data sets with large numbers of descriptors, it would be helpful to be able to sort out features that are not related to the desired output property:

- If the number of input variables is large compared to the number of observations, the effective number of degrees of freedom may be too large for obtaining reliable estimates of the model's parameters.
- Most Machine Learning Techniques have a larger time complexity than linear in the number of observations and/or number of input variables which prohibits the analysis of data sets with several hundred variables.

A key problem is to appropriately distinguish important features from redundant ones. We therefore applied two approaches to reduce the number of features:

- A conservative approach in which only clearly redundant features were removed.
- A more aggressive approach in which after removing redundant features a forward stagewise selection scheme is used to determine up to 12 *most relevant* variables.

3.1. Feature Reduction by Clustering. We first remove constant and low-entropy variables and then pool variables into clusters in which the absolute value of the pairwise correlation coefficients exceeds 0.98. Finally we discard all

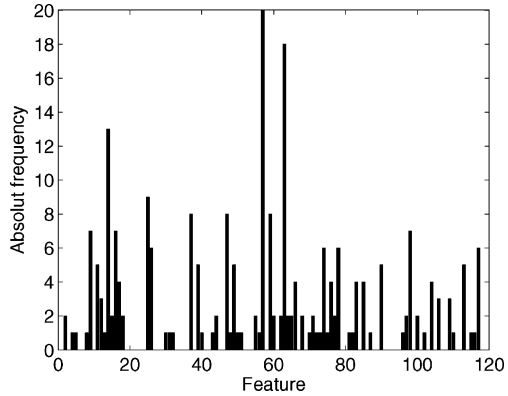

Figure 2. Histogram of the variables selected in 20 cross-validation runs of the feature selection algorithm described in sections 3.1 to 3.2 on the FH GC data set. Note that a variable can be selected at most once in a run. Despite the varying composition of the training data used for every run, the feature selection process exhibits a significant overlap in the selected variables. While most variables occur never or just once, variables 57, 63, and 14 are selected in more than 12 out of 20 runs, variable 57 is present in all of the 20 runs.

variables but a randomly chosen one from each of these clusters.

We did not apply principal component analysis because it leads to linear combinations of variables which would interfere with the goal of a subsequent selection of a small number of features that could be subsequently interpreted by human experts.

3.2. Forward Stagewise Selection. Forward stagewise selection is a greedy-type algorithm that iteratively constructs a subset of relevant variables. It starts with one randomly chosen variable as initial subset and computes the LOO error for all combinations of one of the remaining variables with the variables in the current subset. The variable that improves the LOO error most is added to the current subset. This procedure is repeated until the error does not improve any further or the number of features in the selected subset exceeds a predefined limit. To prevent selecting an irrelevant initial feature, it is removed after a second feature has been selected by the algorithm describe above. Still, this discarded feature could be selected again in a later stage if it actually turns out to be important. We tested the stability of this redundancy removal and feature selection process by computing histograms of the selected variables (see Figure 2). Preferring high-entropy variables would not be beneficial at this point since high entropy does not necessarily include a high correlation with the output variable we want to approximate.

For this selection k-NN models without metric adaption have been used since their sensitivity toward noise in the input variables prevents the selection of irrelevant features. Additionally, they allow the efficient computation of the LOO error.[12]

4. APPLICATION TO ROCHE DATA SETS

4.1. Data Sets. The proposed method has been applied to two data sets:

FH The *Frequent Hitters* data set consists of 902 observations with 1814 descriptors for each sample. The Frequent Hitters set contains 479 molecules coming from the analysis of HTS hit list (experimental part) refined by the vote of eleven medicinal chemists (expert part). The non-Frequent Hitters set consists of a diverse selection of 423 drug molecules. The large number of 1814 descriptors was generated to assess the variable selection process. 1481 of the 1814 descriptors have been computed using the Dragon software,[13] 150 are CATS descriptors,[14] 120 are Ghose and Crippen (GC) descriptors,[15] 63 are additional topological, electronic, count, and structural descriptors.[16] To compare results with a previous artificial neural network (ANN) model,[4] we have extracted a lower-dimensional version of the data set using only GC descriptors which is denoted as FH GC.

CYP. All data used in the *Cytochrome CYP3A4* data set were collected in a CYP3A4 inhibition screening effort at Roche. Each compound was assigned to one of three classes, according to the determined *IC*50 value. Compounds belonging to class *Medium* with $1\ \mu M \le IC50 \le 50\ \mu M$ were eliminated from the data set. Class *Low* contained 186 compounds with $IC50 < 1\ \mu M$. Class *High* contained 224 compounds with $IC50 > 50\ \mu M$. The final CYP3A4 data set consists of 410 observations with 329 descriptors. The descriptor set consists of 146 2D-descriptors from MOE,[17] the 120 Ghose and Crippen descriptors (GC),[15] and an additional set of 63 topological, electronic, count, and structural descriptors.[16] Again, a lower-dimensional variant of this data set was compiled using only GC descriptors as input variables which is denoted as CYP GC.

4.2. Comparison of Methods. Table 1 and Figure 3 show the cross-validated classification rates and cross-validated Matthew's correlation coefficients (CC) for two data sets and the six different processing schemes described here:

SP. Single SVM classifiers were constructed on all input variables of the respective data set, without removing constant or redundant variables (see Appendix A.2). For each cross-validation fold, the optimal SVM parameters were determined by the OOT technique on the training samples.

EP. An ensemble of 15 SVM classifiers was constructed on all input variables of the respective data set, without removing constant or redundant variables. For each cross-validation fold, the optimal SVM parameters were determined by the OOT technique on the training samples.

RS. A random subspace ensemble of k-NN classifiers was constructed on all input variables. No topological parameters had to be adjusted manually (see Appendix A.1).

RRS. Single SVM classifiers were constructed on the respective data set. Constant and redundant variables have been removed before as described in section 3.1. For each cross-validation fold, the optimal SVM parameters were determined by the OOT technique on the training samples.

RRE. An ensemble of 15 SVM classifiers was constructed on the respective data set. Constant and redundant variables have been removed before as described in section 3.1. For each cross-validation fold, the optimal SVM parameters were determined by the OOT technique on the training samples.

RR+FS. First, constant and redundant variables have been removed from the respective data set. Then a forward stagewise selection has been done to identify up to 12 most relevant features as described in section 3.2. An ensemble of 15 SVM classifiers was constructed on the identified features. For each cross-validation fold, the optimal SVM parameters were determined by the OOT technique on the training samples.

Table 1. Cross-Validated Misclassification Rates (CV MCR), Cross-Validated Matthew's Correlation Coefficients (CV CC), and Rounded Average Number of Descriptors (# vars) Used for Constructing Respective Models on Both Data Sets as Described in Section 4.1[a]

data set	method	# vars	CV MCR	CV CC
FH	SP	1814	0.05 ± 0.02	0.91 ± 0.05
FH	**EP**	1814	0.04 ± 0.02	0.92 ± 0.03
FH	RS	43	0.07 ± 0.03	0.86 ± 0.06
FH	RRS	1124	0.05 ± 0.02	0.9 ± 0.04
FH	RRE	1122	0.05 ± 0.03	0.91 ± 0.07
FH	*RR+FS*	12	0.11 ± 0.03	0.79 ± 0.07
FH	RI	1814	0.06 ± 0.02	0.89 ± 0.05
FH GC	**SP**	120	0.06 ± 0.03	0.88 ± 0.05
FH GC	EP	120	0.06 ± 0.03	0.88 ± 0.06
FH GC	RS	11	0.1 ± 0.03	0.79 ± 0.07
FH GC	RRS	80	0.07 ± 0.04	0.86 ± 0.07
FH GC	RRE	81	0.06 ± 0.03	0.87 ± 0.05
FH GC	*RR+FS*	12	0.11 ± 0.03	0.78 ± 0.07
FH GC	RI	120	0.1 ± 0.03	0.8 ± 0.07
CYP	SP	329	0.06 ± 0.04	0.88 ± 0.09
CYP	**EP**	329	0.07 ± 0.04	0.87 ± 0.07
CYP	RS	19	0.1 ± 0.05	0.81 ± 0.09
CYP	RRS	186	0.07 ± 0.04	0.85 ± 0.08
CYP	RRE	187	0.07 ± 0.04	0.86 ± 0.09
CYP	*RR+FS*	12	0.12 ± 0.05	0.76 ± 0.1
CYP	RI	329	0.08 ± 0.05	0.84 ± 0.11
CYP GC	SP	120	0.09 ± 0.04	0.83 ± 0.08
CYP GC	EP	120	0.08 ± 0.05	0.83 ± 0.1
CYP GC	RS	11	0.11 ± 0.06	0.77 ± 0.13
CYP GC	RRS	61	0.1 ± 0.04	0.8 ± 0.09
CYP GC	RRE	61	0.08 ± 0.04	0.83 ± 0.08
CYP GC	RR+FS	12	0.12 ± 0.06	0.75 ± 0.12
CYP GC	RI	120	0.1 ± 0.05	0.81 ± 0.09

[a] We averaged misclassification rates and cross-validated Matthew's correlation coefficients over the 20-folds of the cross-validation and computed standard deviations for both quantities, given after the ± in the columns CV MCR and CV CC. Results were obtained with six different training methods (see section 4.2). The best performing method for each data set is printed in bold, the worst in italic letters. For easier visual inspection the misclassification rates are also depicted in Figure 3.

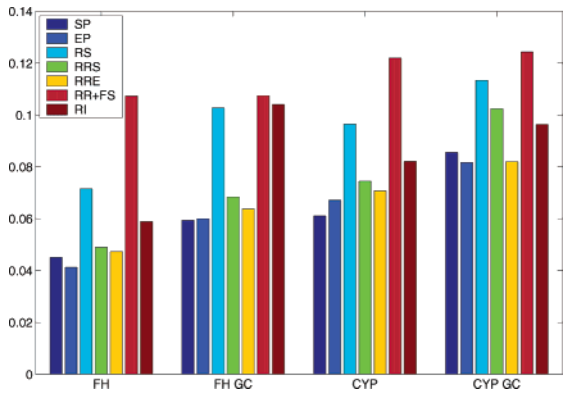

Figure 3. Comparison of cross-validated misclassification rates for all combinations of methods and data sets. Each data set exists in two variants (large descriptor set/120 GC descriptors only). Misclassification rates are averaged over 20 cross-validation folds. Numerical values are also presented in Table 1.

RI. For comparison, we construct single linear models by means of ridge regression.

We used 20-fold cross-validation with 10% test fraction for each partitioning. The test fraction of each fold was used neither for any of the above-mentioned steps nor for deriving scaling parameters or the like.

5. RESULTS AND DISCUSSION

5.1. OOT Performance on the Training Sets. From Table 1 we can infer that constructing classifiers on a subset of aggressively selected features (method RR+FS) is the worst performing training scheme throughout. This result suggests that it is not possible to construct an optimal classifier on a minimal subset of features. Random subspace ensembles (method RS) perform slightly better but fall still into the group of the weakest models in this comparison. They belong to the computationally least demanding methods investigated in this study which could render them an interesting ensemble variant for applications where the computation time is a critical factor. The performance of ridge regression seems to depend strongly on the characteristics of the data sets. Though it always performs better than methods RR+FS, nonlinear models seem to generally outperform this linear method on both data sets.

Single or ensemble SVM classifiers (methods SP and EP) not only are the best-performing on all data sets but also are the most computationally demanding of the methods considered. Actually, the computational demand for SP is not lower than for EP since for each SVM parameter setting an ensemble has to be constructed to assess the OOT error for model selection. Still, this result is astonishing since it contradicts the belief that the generalization ability of statistical learning techniques decreases for high dimensional data sets. The almost vanishing difference between method SP and EP might be caused by a high degree of correlation of SVM classifiers in the ensemble. [The quadratic programming problem arising from the SVM formulation has typically a unique solution.] The ensemble gain of combining highly correlated classifiers tends to be quite small. Nevertheless, Support Vector Machines seem to be a very promising algorithm for a broad range of classification problems, and both methods should be considered when lowest error rates are crucial for a particular application.

Method RRE offers an attractive tradeoff between computational complexity and classification accuracy. On most data sets, the accuracy of this method is only slightly worse than that of methods SP and EP, while the computational demand is significantly smaller due to the prior redundancy reduction. This makes method RRE a preferred choice when one would like to combine the greater flexibility of nonlinear models with a reduced computational effort for the training.

5.2. Frequent Hitters Data Set. We validated the stability of the RRE model based on the 120 GC descriptors for the Frequent Hitters data set by using the latest data set published by Shoichet's group. This data set is composed of 48 aggregators and 63 nonaggregators.[18] Since the training and the validation sets have no molecule in common, we checked the overlap by performing PCA in GC space.

From Figure 4, which shows the score plot generated with SIMCA-P+,[19] we can conclude that the validation set is covered by the training set. Then, we compare the results produced by the SVM model to the ones of the ANN model (see Table 2).The false negatives (FN) are compounds that were found to inhibit proteins through aggregate formation but were not classified as FH. Under the same training conditions and with the same variables and identical observations, the SVM model seems more stable than the ANN model. Indeed, the CCs for the cross-validation of training

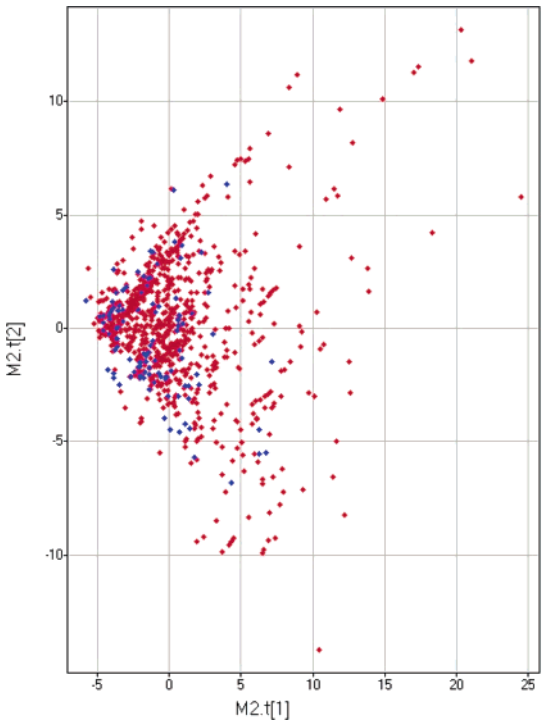

Figure 4. Score plot of the PCA projection on the plane formed by the 2 first PCs of the training (red points) and validation set (blue points) of the Frequent Hitters (FH) model in the GC descriptors space.

Table 2. Comparison of SVM (RRE) Model with a Previously Described ANN Model[4] on a Validation Data Set Consisting of 111 Compounds

	ANN (GC)	SVM (RRE GC)
True Positives (TP)	30	32
False Positives (FP)	10	5
True Negatives (TN)	53	58
False Negatives (FN)	18	16
CC Matthews	0.48	0.63

set are 0.87 and 0.81 for the SVM and ANN, respectively, whereas the CC for the validation set are 0.62 and 0.48. This observation is confirmed by an analysis of the chemical structures of the misclassified compounds. The SVM approach yields a more consistent picture of common substructures in the false positives (FP) and FN sets. Figure 5 depicts 6 representatives of the FN and 5 of FP.

Among the compounds erroneously classified as non-FH (FN), some privileged structures for biogenic amine G-protein coupled receptor (GPCR) are identified, e.g. compounds FN6, and also steroids such as FN1. Several compounds in the FN set, such as econazole (FN4), nicardipine (FN2), and the protein kinase C inhibitor FN5, are actually optimized druglike substances. Because of the similarity between these compounds and others in the training set of druglike compounds, this misclassification makes sense: known drugs should not be classified as frequent hitters even though such compounds may be able to aggregate at high concentrations. The dihydro-indolone substructure present in FN4 (ropinirole), which is a representative of a cluster of 6 molecules, is also present only in non-FH drug compounds in our training set.

For the group of compounds that is experimentally not classified as FH (FP), the results may also be expected from the composition of the training set. Since the GC descriptors

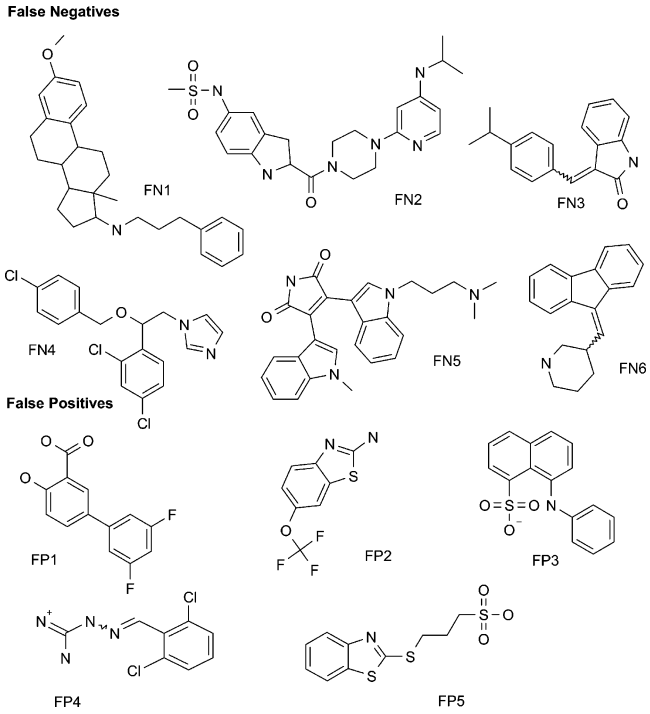

Figure 5. Six representatives of the FN and the five FP. These compounds are misclassified by both ANN and SVM models.

stand for functional groups or small substructures, we have performed substructure searches in the FH training set with parts of the misclassified compounds. For example, the hydroxybenzoic acid moiety of FP1 is present in 25 FH and only in 3 non-FH molecules; this finding is verified for various substructures of other FP. Overall, the SVM models clearly outperform the ANN method both in terms of quantitative measures and consistency in a chemical sense.

5.3. Cytochrome CYP3A4 Data Set. The performance of the RR+FS and SP models on the cytochrome CYP3A4 data set were also analyzed in more detail. The GC descriptors were calculated for a validation set of 90 compounds that belong to a single structural class and originate from a Roche drug discovery project. *IC*50 values have recently been measured for these compounds. 40 compounds belong to class *Low* with $IC50 < 1\ \mu M$ and 50 compounds belong to class *High* with $IC50 > 50\ \mu M$. To see whether the new class of compounds is covered by the RR+FS and SP models, we checked the overlap by performing PCA in GC space (see Figure 6). Although all validation compounds belong to the same structural class, there is considerable overlap with the training set. Only 5 compounds of class *High*, which are located outside the area covered by the training set (circled in Figure 6) were removed from the validation set, resulting in a total of 85 compounds in the validation set. False positives (FP) are compounds that are not inhibitors of CYP3A4 but are classified as such (and vice versa for the FN). The RR+FS and SP models give CCs of 0.31 and 0.39 for the validation set compared to 0.73 and 0.85 for the cross-validation. We compared these results with those of an earlier prediction tool based on the Partial Least Squares (PLS) technique. The PLS model was built on a diverse subset of 100 compounds out of the 410 compounds that were used to build the SVM models. Since the results of this model crucially depend on the classification thresholds, results are given for two different thresholds. At

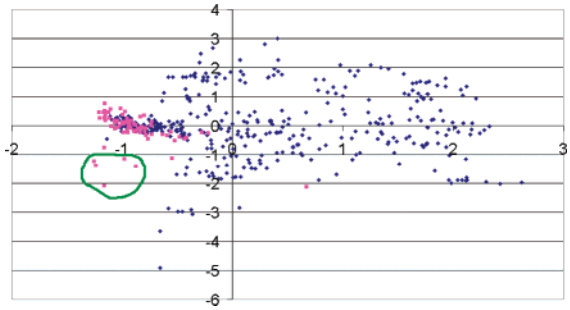

Figure 6. Score plot of the PCA projection on the plane formed by the first two principal components of the training (blue points) and validation set (magenta points) of the Cyp3A4 model in the GC descriptors space. The circled points are removed from the validation set since they do not overlap with the training set.

Table 3. Comparison of Two SVM Models with a Previously Described PLS Model on the Validation Set of 85 Compounds for the Cyp3A4 Problem[a]

	SVM		PLS	
type, respectively threshold	RR+FS	SP	0.4	0.5
True Positives (TP)	35	35	36	36
False Positives (FP)	27	23	18	26
True Negatives (TN)	18	22	27	19
False Negatives (FN)	5	5	4	4
CC Matthews	0.31	0.39	0.52	0.36

[a] For the PLS model two different thresholds, 0.4 and 0.5, were used to distinguish between both classes. Both SVM and PLS models were constructed on the 120 GC descriptors.

a threshold of 0.4 the PLS model seems to be more stable than the SVM models for the validation set, resulting in a CC of 0.52. Using a threshold of 0.5, the performance of the PLS model is decreased to a CC of 0.36. Interestingly, PLS and SVM based models perform similarly good in the correct prediction of true positives (inhibitors); correct prediction of true negatives (noninhibitors) is hardly possible with any prediction method.

6. CONCLUSIONS AND OUTLOOK

We demonstrated the usage of ensemble methods for classification problems arising within the process of drug development. Ensemble methods bear the advantage of being applicable to various types of underlying statistical learning algorithms, from which we chose k-NN classifiers and Support Vector Machines. Though more difficult to interpret, these model types can cope with nonlinear problems where linear methods suffer from high bias due to their limited flexibility.

On all training data sets, ensembles of SVM classifiers constructed with prior removal of redundant features (method RRE) exceed a classification rate of 92% in the cross-validation setting. On the Frequent Hitters classification problem, we were able to improve previous results[4] by reducing the misclassification rate from 10% to 4%−5% (methods SP/EP/RRS/RRE).

Results on the validation sets for the FH and CYP3A4 data sets show an overall improvement, even though not as clear as on the training sets. While the ensemble models show an improved performance in the FH validation, the performance could not be improved for the CYP3A4 validation set relative to previously build PLS models.

Despite careful application of validation procedures on the training sets there is still a significant difference between the cross-validated classification performance and the classification performance seen on the validation sets. In our opinion this is caused to a lesser extent by classical overfitting of the constructed models rather than by the different distribution of the validation samples. This opinion is encouraged by the comparison of the degree of overlap of the validation set with the training set for the FH GC and the CYP3A4 GC data sets. The discrepancy in classification performance between training and validation set is lower for the FH GC data sets which overlap more uniformly (see Figure 4) than for the CYP3A4 GC data sets which overlap less homogeneously (see Figure 6).

Chemical space is so vast that a training set which covers it uniformly would be prohibitively large and therefore difficult to obtain experimentally. To reduce the generalization error outside the region of the chemical space covered by the training set we would propose to further pursue the *Transductive Learning* approach proposed by Vapnik.[20] Furthermore, meaningful and generally valid descriptors are difficult to generate for multiple-mechanism classification problems.

We think that ensemble methods offer an interesting way of creating well-performing classifiers in cheminformatics. Additionally, the Out-of-Train technique as an ensemble variant of the cross-validation allows for a combination of training an ensemble of models and validation at the same time. The Random Subspace method significantly accelerates the construction of ensembles of nonlinear models on high-dimensional data sets with as many as 1800 input variables though it sacrifices some classification accuracy.

APPENDIX: DESCRIPTION OF MODEL TYPES

A.1. Nearest-Neighbor Models. A k-nearest-neighbor model takes a kernel-weighted average over the observations y_i in the training set closest [Self-matches of data set points (i.e. each point is considered to be its own nearest neighbor) are prohibited by default since this would strongly bias the error on the training set.] to the query point $\vec{x} \in \mathbf{R}^D$ to produce the outcome

$$f(\vec{x}) = \frac{1}{\sum w_i \vec{x}_i \in N_k(\vec{x})} \sum w_i y_i \qquad (2)$$

where $N_k(\vec{x})$ denotes the k-element neighborhood of $\vec{x}$, given a proper metric. Common choices for the metric are L_1, L_2, and L_∞ norm. To compensate for irrelevant input dimensions, distances are computed using a weighted metric:

$$d(\vec{x},\vec{y}) = \left(\sum_{i=1}^{D} (m_i(x_i - y_i))^L\right)^{\frac{1}{L}} 0 \le m_i \le 1 \qquad (3)$$

The vector $\vec{m}$ of metric coefficients is adjusted by a Genetic Algorithm (GA) that works on a population of vectors of metric coefficients. The GA starts with a population of randomly initialized individuals. The higher the dimension D, the more of the initial metric coefficients are artificially set to zero to favor sparse solutions. A fitness value is assigned to each individual according to its leave-one-out (LOO) training error. The smaller the error, the

higher the fitness value assigned. To create the individuals of the next generation, two individuals of the current generation are selected randomly with a probability proportional to their fitness in order to create two offsprings by straightforward crossover mating. Additionally, random mutations in single coefficients of the offsprings are introduced with a probability 0.2. To prevent losing good solutions, the best solutions of the current generation are copied into the next generation (*N*-elitist approach). The population evolves over a predefined number of generations or until the diversity within the population shrinks below some given threshold.

The smoothing kernel weights w_i are distance dependent $w_i = (1 - (d_i/d_{k+1})^p)^p$, where d_i denotes the distance from the query point $\vec{x}$ to the *i*th nearest neighbor. The parameter p of this smoothing kernel is chosen out of {0.0, 0.5, 1.0, 2.0, 3.0}. The number of nearest neighbors k is adjusted in order to deliver smallest LOO error on the training set. For our investigation we employed the ATRIA implementation of a fast nearest neighbor algorithm[12] that allows the efficient computation of the LOO error.

A.2. SVM Classifiers. Support Vector Classifiers[20,21] belong to the family of *Large Margin Classifiers*. Instead of minimizing a loss function that measures the deviation between model and training outputs, SVMs compute a separating hyperplane that maximizes the margin between both classes. This leads to a quadratic programming problem with a unique solution that can be expressed in terms of training points that lie on or violate the margin, the so-called *Support Vectors*.

The primarily linear approach can be extended to nonlinear problems by means of *kernels* that provide a nonlinear mapping from the input space to a possibly infinite-dimensional *feature space* in which the separating hyperplane is constructed. Popular choices of kernels are as follows:

- Polynomial kernels
- Radial basis function (RBF) kernels with Gaussian basis function

For the numerical experiments we used the libSVM[22] package and applied radial basis kernels, as these yield superior classification performance for most practical applications. Two main tuning parameters remain to be chosen, the width γ of radial basis functions and the penalty parameter C. They can be easily optimized by an exhaustive search over a predefined set of parameter pairs (γ, C), e.g. $D \cdot \gamma \in \{0.1, 1, 5, 10, 50\}$ and $C \in \{1, 10, 100, 1000\}$, where D is the number of input features. For each combination, we compute the Matthew's Correlation Coefficient (CC) with the Out-of-Train technique (see subsection 2.4) and choose the ensemble with highest OOT CC.

A.3. Ridge Regression. Ridge regression constructs a linear model $\hat{y} = \boldsymbol{X}\beta + \beta_0$, but instead of minimizing the sum of squared residuals $(y - \boldsymbol{X}\beta - \beta_0)^T(y - \boldsymbol{X}\beta - \beta_0)$, it minimizes the regularized loss function (Tikhonov regularization):

$$\mathrm{RSS}_{\mathrm{pen.}} = (y - \boldsymbol{X}\beta - \beta_0)^T(y - \boldsymbol{X}\beta - \beta_0) + \lambda\beta^T\beta \quad (4)$$

The additional penalty $\lambda\beta^T\beta$ shrinks the regression coefficients $\hat{\beta}$ toward zero, thereby moderately increasing bias while considerably decreasing variance of the constructed models. The penalty parameter $\lambda \geq 0$ controls the amount of shrinkage and can be used to fine-tune the bias-variance tradeoff. For this study, the optimal ridge penalty λ is automatically determined by LOO−CV[9] on each training fold individually. To apply ridge regression to a binary classification problem, training outputs are coded as $y = 0.0, 1.0$ and a threshold of 0.5 is applied to discriminate between both classes when doing predictions. Prior to model construction, input variables are normalized by removing the mean and dividing by the standard deviation for each variable separately.

ACKNOWLEDGMENT

The authors would like to thank the inventor of the libSVM, Chih-Jen Lin, for stimulating comments and discussions. This work is supported by the Deutsche Forschungsgemeinschaft (DFG) grant LE 491/11-1. We would like to thank all colleagues at Roche in Basel and at the MPI in Saarbrücken for their support.

REFERENCES AND NOTES

(1) Perrone, M. P.; Cooper, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In *Neural Networks for Speech and Image Processing*; Mammone, R. J., Ed.; Chapman-Hall: 1993.

(2) Breiman, L. Randomizing Outputs to Increase Prediction Accuracy. *Machine Learning* **2000**, *40*, 229−242.

(3) Freund, Y.; Schapire, R. A short introduction to boosting. *J. Jpn. Soc. Artif. Intel.* **1999**, *15*, 771−780.

(4) Roche, O., et al. Development of a virtual screening method for identification of frequent hitters ïn compound libraries. *J. Med. Chem.* **2002**, *45*, 137−142.

(5) Agrafiotis, D. K.; Cedeno, W.; Lobanov, V. S. On the Use of Neural Network Ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 903−911.

(6) Mattioni, B. E.; Kauffman, G. W.; Jurs, P. C.; Custer, L. L.; Durham, S. K.; Pearl, G. M. Predicting the Genotoxicity of Secondary and Aromatic Amines Using Data Subsetting To Generate a Model Ensemble. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 949−963.

(7) Lucic, B.; Nadramija, D.; Basic, I.; Trinajstic, N. Toward Generating Simpler QSAR Models: Nonlinear Multivariate Regression versus Several Neural Network Ensembles and Some Related Methods. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1094−1102.

(8) Krogh, A.; Vedelsby, J. Neural Network Ensembles, Cross Validation, and Active Learning. In *Advances in Neural Information Processing Systems*; Tesauro, G., Touretzky, D., Leen, T., Eds.; The MIT Press: 1995; Vol. 7.

(9) Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning;* Springer-Verlag: 2001.

(10) Domingos, P. A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In *AAAI/IAAI*; 2000.

(11) Breiman, L. Bagging Predictors. *Machine Learning* **1996**, *24*, 123−140.

(12) Merkwirth, C.; Parlitz, U.; Lauterborn, W. Fast Exact and Approximate Nearest Neighbor Searching for Nonlinear Signal Processing. *Phys. Rev. E* **2000**, *62*, 2089−2097.

(13) DRAGON v1.11, Milano Chemometrics and QSAR group, P.za della Scienza 1, 20126 Milano, http://www.disat.unimib.it/chm/Dragon.htm

(14) Viswanadhan, V.; Ghose, A.; Revankar, G.; Robins, R. Atomic physicochemical parameters for three-dimensional structure directed quantitative structure−activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 163−172.

(15) Ghose, A.; Crippen, G. Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure−activity relationships. 2. Modeling dispersive and hydrophobic interactions. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 21−35.

(16) Zuegge, J.; Fechner, U.; Roche, O.; Parrott, N. J.; Engkvist, O. E. A. A fast virtual screening filter for cytochrome P450 3A4 inhibition liability of compound libraries. *Quant. Struct.-Act. Relat.* **2002**, *21*, 249−256.

(17) MOE 2003.02 − Chemical Computing Group Inc., Montreal, Quebec, Canada.

(18) Seidler, J.; McGovern, S.; Doman, T.; Shoichet, B. Identification and Prediction of Promiscuous Aggregating Inhibitors among Known Drugs. *J. Med. Chem.* **2003**, *46*, 4477−4486.
(19) SIMCA-P+; Umetrics AB (P.O.B. 7960, SE90719): Umeå, Sweden, 2002.
(20) Vapnik, V. *The nature of statistical learning theory;* Statistics for Engineering and Information Science: Springer-Verlag, Berlin, 1999.
(21) Warmuth, M.; Liao, G.; Rätsch, G.; Mathieson, M.; Putta, S. E. A. Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667−673.
(22) Chang, C. C.; Lin, C. LIBSVM − A library for support vector machines (http://www.csie.ntu.edu.tw/∼cjlin/libsvm), 2001.

CI049850E