Ligand-Target Prediction Using Winnow and Naive Bayesian Algorithms and the Implications of Overall Performance Statistics

Florian Nigsch,[†] Andreas Bender,[‡,§] Jeremy L. Jenkins,[‡] and John B. O. Mitchell*,[†]

Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, United Kingdom; Lead Discovery Informatics, Center for Proteomic Chemistry, Novartis Institutes for BioMedical Research, 250 Massachusetts Avenue, Cambridge, Massachusetts 02139; and Division of Medicinal Chemistry, Leiden/Amsterdam Center for Drug Research, Leiden University, Einsteinweg 55, 2333 CC, Leiden, The Netherlands

Received March 7, 2008

We compared two algorithms for ligand-target prediction, namely, the Laplacian-modified Bayesian classifier and the Winnow algorithm. A dataset derived from the WOMBAT database, spanning 20 pharmaceutically relevant activity classes with 13 000 compounds, was used for performance assessment in 24 different experiments, each of which was assessed using a 15-fold Monte Carlo cross-validation. Compounds were described by different circular fingerprints, ECFP_4 and MOLPRINT 2D. A detailed analysis of the resulting ≈2.4 million predictions led to very similar measures for overall accuracy for both classifiers, whereas we observed significant differences for individual activity classes. Moreover, we analyzed our data with respect to the numbers of compounds which are exclusively retrieved by either of the algorithms—but never by the other—or by neither of them. This provided detailed information that can never be obtained by considering the overall performance statistics alone.

1. INTRODUCTION

Classification methods have a central role in contemporary drug discovery. They are used in virtual screening experiments to sift through digital collections of molecules and to decide to which class a certain molecule is most likely to belong. The class label is typically determined according to some user-defined threshold (e.g., active if the IC_{50} value is <10 nM). For some uses, only two classes are used (binary classification), as, for example, in a classification into the two classes "active" or "inactive". In a multiclass classification, more than two classes are employed. One example of multiclass classification used in the drug discovery process is the prediction of the most likely protein targets for a molecule. The predicted class is the class that the algorithm assumes to be the most likely one, but other high-ranking classes may also be of interest, because they can be indicative of potential off-target effects. Continuous variables (e.g., solubility, permeability) may be attributed to discrete classes, according to user-defined class boundaries, and then a multiclass model may equally be used for these data. The resulting model could be used to distinguish between compounds with, for example, unacceptable, low, medium, high, or very good solubility values.[1]

The application of all classification methods entails two steps: a training procedure and the classification of test cases. Different sets of molecules are used for these steps, typically called training and test sets, according to their respective purposes. In the training step, the algorithm identifies patterns or rules in the data pertaining to discriminating features of the distinct classes that it will be used to separate. In the subsequent classification step, the knowledge extracted from the training set, and subsequently incorporated into computational rules, is applied to each example of the test set to predict its class label—in the best case, the "true" label.

A notorious limitation in datasets of biological activities of molecules is the fact that not every molecule is measured against all targets, although for some molecules this is indeed the case and they may be labeled as active against more than one target. As a result, the data that are available and used to build models are incomplete. Consequently, during the training step, a molecule listed as active in several classes will be considered separately as a member of each class listed. Also, the predicted class label of a molecule may not be any of the classes that are listed for it. Therefore, this prediction would be a false positive in the predicted class, whereas it would be a false negative in the class for which the molecule is actually listed. Given this lack of information of activity in the predicted class, (i) the prediction may still be correct, but untested, and there are examples of experimental confirmation of such predictions, and (ii) the measures of accuracy of the classification procedure are distorted. Until all possible combinations of (commonly used) proteins and ligands are tested, this will remain a limitation of models such as that which we present here.[2,3]

The two-tiered approach, training—testing, which is described in the previous paragraph, is commonly used with machine learning methods. In a real-world setting, data with known class labels are used for the training set and, in the hope of obtaining useful predictions, this trained classifier

* Author to whom correspondence should be addressed. Phone: +44 (0)1223 762 983. Fax: +44 (0)1223 763 076. E-mail: jbom1@cam.ac.uk.

[†] Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge.

[‡] Lead Discovery Informatics, Center for Proteomic Chemistry, Novartis Institutes for BioMedical Research.

[§] Division of Medicinal Chemistry, Leiden/Amsterdam Center for Drug Research, Leiden University.

10.1021/ci800079x CCC: $40.75 © 2008 American Chemical Society
Published on Web 12/04/2008

is then applied to a set of examples of unknown class labels. If the class labels of interest are the activities of molecules, with respect to a given set of receptors, the objective of the entire process is to find a quantitative relationship between the structure of molecules and the associated activity, i.e., one seeks to establish quantitative structure−activity relationships (QSARs).

Finding the quantitative relationship between individual molecular structures and their biological activities requires, on one hand, a representation of molecules amenable to computation, and, on the other hand, suitable algorithms to effect the aforementioned training and prediction steps. The computational handling and representation of molecular structures has been a long-standing effort in the chemical information and modeling community. As such, there is a wealth of so-called "molecular descriptors" from which to choose, depending on the molecular property for which one wishes to build a model.[4−6] Binary fingerprints (e.g., MACCS from MDL,[7] UNITY from Tripos),[8] and circular fingerprints in particular, are widely accepted and commonly used as molecular descriptors for the purpose of building models to predict bioactivities.[9−11]

The algorithms from which to choose for the processing of the molecular information are less numerous than the molecular descriptors. In order of increasing complexity, possible choices are linear discriminant analysis (LDA),[12] naive Bayesian classifier,[13] Random Forest (RF),[14] binary kernel discrimination (BKD),[15] support vector machine (SVM),[16] and artificial neural network (ANN).[17] This list is nonexhaustive, and the proposed order in complexity is certainly not inflexible. However, there are inherent differences in the design of the methods that invariably render some more complex than others. The relatively simple processing of data in a naive Bayesian classifier is contrasted by the more complicated methodology of other methods. RF is an ensemble of up to hundreds of individual classification trees,[18] SVMs try to find a separating hyperplane after projecting the data into higher-dimensional spaces,[19] and multilayer ANNs adjust up to hundreds of individual parameters in an iterative optimization process.[20] Because there often is no clearly identifiable optimal solution to a problem, each of these methods has its own set of context-dependent advantages and potential drawbacks. Consequently, comparisons of classifiers always must be considered with care. For a given dataset, there will always be a method that performs "best". Therefore, the meaning of "best" is clearly context-dependent. In the context of chemistry, not only is the location of the dataset in chemical space important, but also the partition of this dataset into a training set and a test set.

An important application of classification methods is found in the prediction of protein targets for compounds. One of the first models for the prediction of protein targets was developed by Lagunin et al.[21] More recently, they also applied their methodology to elucidate molecular mechanisms of action to explain certain toxic effects.[22] In 2005, Fliri et al. coined the term "biological activity spectrum".[23] They define the "biological activity spectrum" as the experimental activities of a given chemical across a panel of assays. This was used to cluster biological activities, as well as for the prediction of untested compounds. As opposed to experimentally determined activity spectra, Bender et al. used Bayes scores across a panel of target proteins ("Bayes affinity fingerprints") as descriptors for virtual screening experiments.[24] A similar multiclass Bayesian approach was applied by Nidhi et al. to build a model for protein target prediction based on the WOMBAT database; this model was then used to deconvolute therapeutic target annotations in the MDL Drug Data Report (MDDR).[7,25,26] Models for target prediction, in combination with high-content screening programs, were used to identify the mechanism of action.[27] For a review on the topic of protein target prediction, see the work of Jenkins et al.[28]

A recent paper presented an algorithm known as Winnow in combination with a feature combination technique (primarily) used in text classification, orthogonal sparse bigrams (OSBs). Essentially, a highly accurate spam filter was converted into a tool for virtual screening experiments and successfully applied.[29,30] A comparison with RF showed that Winnow performed equally well or better on a dataset that is commonly used for such comparisons. Moreover, Winnow was determined to be straightforwardly amenable to use with larger datasets.[30]

The main purpose of this paper is to compare the performance of Winnow with an industry-standard combination of classifier and fingerprints on the specific task of the prediction of protein targets for organic molecules. We compared a Laplacian-modified naive Bayesian classifier to Winnow, using SciTegic's ECFP_4 fingerprints as molecular descriptors.[31] Both classifiers were previously shown to perform well in the context of the prediction of biological activity.[11,30] To be able to have conditions that are as comparable as possible, we implemented the Bayesian classifier ourselves in Python. This was motivated by two reasons: (i) we could implement it as an online algorithm, processing the data piece by piece, as does our implementation of the Winnow algorithm; and (ii) this allowed us to incorporate the exact same procedure for the on-the-fly generation of additional features as OSBs.[30] To ensure that our implementation of this algorithm was as similar as possible to a standard application, we compared the results of our Python program and SciTegic's Bayesian classifier. The results obtained on a predefined split of the dataset into a training set and a testing set confirmed that our implementation was basically equivalent to the commercial one, with the exception of some minor differences that are due to post-processing steps that occur in the commercial version (such as pruning of unimportant features, which has not been integrated into our version; see the Supporting Information).

In the case of the present protein−target prediction problem, the reduction of all results into one number only, to compare different sets of parameters and methods, leaves out a wealth of information. Our findings lead us to argue that the assessment of the overall accuracy of a method is not sufficient to provide a complete picture of its performance. Moreover, an analysis of the wealth of data (≈2.4 million predictions) that resulted from the 15-fold Monte Carlo cross-validations of all the 24 experiments that we performed shows that (i) overall performance statistics, even if cross-validated, should not be used exclusively; (ii) there are significant class-specific differences; and (iii) there are individual compounds, as well as groups of compounds, that are either exclusively retrieved by one of the algorithms or by neither.

Table 1. Dataset of 20 Activity Classes Derived from WOMBAT

activity	counts
alpha1a	753
CA II	1183
CCK1	639
CCK2	741
CDK1 cyclinB	133
CDK2 cyclinA	211
CDK4	43
CDK5	54
D2	1891
delta	1473
gamma secretase	118
kappa	1191
m1	764
m2	670
mGluR5	105
mu	1664
PDE4	549
PDE5	490
Src	143
tubulin	180
TOTAL	12995

2. DATASET, FINGERPRINTS, AND DATA HANDLING

Dataset and Fingerprints. We selected compounds that were active against one or more of 20 targets of pharmaceutical interest from the WOMBAT 2007.1 database.[26] Compounds were identified as "active" if the K_i or IC_{50} values assigned were equal to or below 10 μM against the target. The 20 targets were selected to include both close homologues (e.g., CCK1 and CCK2, CDK4 and CDK5) as well as very different target classes. The complete list of targets, with their respective dataset sizes, is given in Table 1.

For all molecules, we calculated two types of circular fingerprints: the first type is SciTegic's extended connectivity fingerprints (ECFP_4) and the second one is MOLPRINT 2D.[9,11] MOLPRINT 2D fingerprints were calculated using an in-house Python script with MySQL support that we also provide as Supporting Information. We encoded the Sybyl atom types as they are encountered in the MOL 2 files; in the original MOLPRINT 2D fingerprint, the atom types are replaced with numerical codes. For example, where our fingerprint yields C; 1−O−1; 2−C.ar−1; 3−C.ar−2, in the original MOLPRINT 2D fingerprint, the atom types C, O, and C.ar would be replaced with numbers. As opposed to ECFP_4 fingerprints, the MOLPRINT 2D fingerprints are not hashed, but used as strings.

Our in-house circular fingerprints were calculated for two radii, namely, two and three bonds from each central atom. Therefore, the former is closely related to ECFP_4, where the number represents the diameter. These two fingerprints will be designated CFP2 and CFP3 for the remainder of this article. Therefore, every molecule is described by three fingerprints: CFP2, CFP3, and ECFP_4.

Analysis of the Dataset. Of the 20 classes, totalling 12 995 molecules, the classes of the cyclin-dependent kinases (CDK) 4 and 5 are the smallest, with 43 and 54 members, respectively. The largest classes are dopamine subtype 2 receptors, with 1891 members, followed by the μ opioid receptors (MORs), with 1664 molecules. The average number of molecules per class is 650, with a median of 594 molecules per class. The molecular weight range spanned by the molecules ranges from 111 Da to 3946 Da, with an average value of 488 Da. The computed log P (abbreviated as clog P) values range from −16 to 13, with an average value of 3.36. Carbonic anhydrase II inhibitors had the lowest average clog P with a value of 1.23, whereas the γ secretase inhibitors were the most lipophilic molecules, with an average clog P of 4.91. The classes with the fewest "rule of five" (Ro5) violations were cyclin-dependent kinase (CDK) 4 (zero violations), followed by CDK 5 (at least one violation in 1.8%).[34] Most Ro5 violations were observed for the two cholecystokinins, CCK 1 and 2, with at least one violation in 75.5% and 71.0%, respectively.

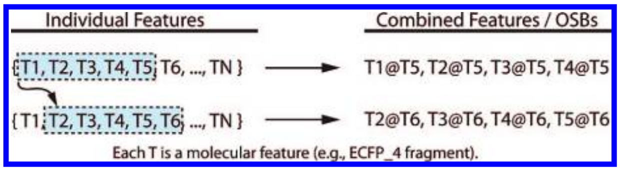
Individual Features
Combined Features / OSBs
{T1, T2, T3, T4, T5, T6, ..., TN }
T1@T5, T2@T5, T3@T5, T4@T5
{T1, T2, T3, T4, T5, T6, ..., TN }
T2@T6, T3@T6, T4@T6, T5@T6
Each T is a molecular feature (e.g., ECFP_4 fragment).

Figure 1. Creation of orthogonal sparse bigrams (OSBs) from a set of individual features {$T1$, $T2$,..., TN}. The first two steps for a windows size of $w = 5$ are shown.

Partitions of the Dataset. To obtain reliable figures for the accuracy of each algorithm in each different setting, we performed a 15-fold Monte Carlo cross-validation (the total data are randomly split into a training set and a test set 15 times) of each single experiment that we performed. The splits of our dataset have all been performed based on a random partition of the compounds into two sets: A and B. For each class, we selected 70% for set A, and the remaining 30% have been kept for set B. These sets A and B were used as either a training set or a test set, allowing us to determine if changes in the size of the training set entail significant changes in classification accuracy. The same 15 splits of our dataset into a training set and a test set have been used for each experiment.

Practical Application. Our implementation of the Winnow algorithm previously described was done in C++, and the Bayesian classifier was implemented in Python. Data analysis and figure plotting was performed in the freely available statistical framework R.[35] All calculations have been conducted on an iMac G5 (1.9 GHz) with 2.5 GB of physical memory.

3. METHODS

Orthogonal Sparse Bigrams (OSBs). To augment the pool of features available for the classification, we create so-called "orthogonal sparse bigrams" (OSBs) from the set of initial features. The set of additional features created consists of a nonexhaustive combination of pairs of original features. The incorporation of these additional features maps the original problem into a feature space of higher dimensionality, where potential synergies between features may be taken into account. Both algorithms create these OSBs on the fly for each training/test example according to the following procedure. A window of size w is moved over the ordered sequence of molecular features and bigrams are created by taking two out of the w features, under the condition that the newest one is always present.[29] For a set T of features $\{t_1,..., t_N\}$ and a window size of $w = 5$, the first set of created bigrams would consist of $\{(t_1, t_5),(t_2, t_5),(t_3, t_5),(t_4, t_5)\}$ (see Figure 1). These OSBs span different axes in feature space and are orthogonal to each other in that way. They are also orthogonal in the sense that no OSB feature can be obtained by combining any other arbitrarily chosen pairs of OSBs, thereby making every single OSB feature nonredundant.

Because of the fact that the creation of OSBs is dependent on the exact order of the sequence of the individual features, it is desirable to control this ordering. To get a deterministic process for the creation of bigrams, we chose the following approach. The list of individual atom-centered atomic neighborhoods is converted into a *set* in the mathematical sense of the term, i.e., every single element occurs exactly once in that set. Subsequently, the features in the set are ordered lexicographically. This entails the clustering together of fragments stemming from the same type of central atom in the case of MOLPRINT 2D, which used the Sybyl atom types.[9] Therefore, all features centered on aliphatic carbons, for example, are regrouped, and the combinations arising are therefore between two features with similar central atoms. Although these atoms may be quite far apart in the actual molecule, they are close in the lexicographical ordering. Thus, to a certain extent, the OSBs created take into account the simultaneous, but topologically distant, presence of features. In addition to aliphatic carbons (C) being grouped together, so are the aromatic carbons (C.ar), hydrogen bond donors (such as sp^3 nitrogens (N.3)), or hydrogen bond acceptors (such as carboxylates (O.co2)). Therefore, the OSBs should be able to encode pharmacophores (such as two polar groups at opposing ends of the molecule) implicitly. The increases observed in classification accuracy with Winnow that result from the use of OSBs are possibly due to such effects, at least to a certain degree.[30] Although the ordering of the hashed ECFP_4 fingerprints may have a different chemical equivalent, the same argument still holds for those, too.

The exhaustive enumeration of all combinations of the m features of a molecule would yield $M_{\text{ex}} = m(m - 1)/2$ unique pairs. In comparison to that, $M_w = (m - w + 1)(w - 1)$ OSBs are obtained when a sliding window of size w is used. This is the product of the $w - 1$ bigrams obtained for each window, and the $m - w + 1$ possible windows of size w that can be formed. Because $M_w/M_{\text{ex}} < 1$, the computational overhead is less if the exhaustive enumeration is avoided, and it was shown that an exhaustive enumeration does not lead to the most-accurate results.[30]

Classification Algorithms. We used two algorithms that are able to perform multiclass classifications: the Bayesian algorithm and the Winnow algorithm. With each algorithm, we built a model that predicts a score for each class that the algorithm is aware of. The predicted class is then the class corresponding to the highest score. As mentioned previously, some compounds may be listed as active in more than one class. To account for this polypharmacology in the models, each unique combination of "compound−activity class" can be part of either the training set or the test set.

Winnow Algorithm. The Winnow algorithm is a linear-threshold learning algorithm that was originally designed to learn Boolean (logical) functions in high-dimensional feature spaces.[31] One of its main characteristics is its multiplicative error-driven learning procedure. In that sense, it is very similar to a Perceptron, which instead uses an additive learning rule to minimize prediction errors. By design, the Winnow algorithm is well-suited to work with datasets that contain many irrelevant attributes. We used the implementation of the Winnow algorithm that was reported in a recent paper. In the following, we will give a brief description of this algorithm; for a more detailed description, we refer the reader to our previous paper.[30] Our implementation of the Winnow algorithm in C++, as used in this paper, is provided as Supporting Information.

The Winnow algorithm holds an n-dimensional weight vector $w^c = (w_1^c, w_2^c,..., w_N^c)$ for each class c, with w_i^c the weight of feature i in class c; N is the cardinality (size) of the set F_{tot}, the set of all features in all classes of the training set (i.e., the set of features occurring in the training set). All w_i are initially set to 1. Every single training and test instance x is presented to the algorithm as a set of features $F_x = \{f_i\}_{i=1}^{i=N_x}$, with $N_x \ll N$ (where N_x is the number of features present in that instance) and $F_x \subset F_{\text{tot}}$. Each feature $f_i \in F_x$ is called an "active feature of instance x", and N_x is the number of active features of instance x. The score S_x^c for class c of instance x is then calculated as the sum of the weights of its active features:

$$S_x^c = \sum_{j=1}^{N} \delta(f_j \in F_x) w_j^c$$

with

$$\delta(f_j \in F_x) = \begin{cases} 1 & (\text{if } f_j \in F_x) \\ 0 & (\text{otherwise}) \end{cases}$$

The error-driven learning procedure is effected through a series of trials that are comprised of three steps: (1) algorithm receives instance; (2) prediction of label; and (3) algorithm receives true label ("reinforcement") and adjusts the weights accordingly. The weight vectors are only updated if the predicted label for a given training instance is wrong or (see below) is close to being wrong.[31] A prediction is considered a misclassification if the score is below the threshold for the true class, or above the threshold for the wrong class(es). The threshold T_x for an instance x is the same as the number of active features N_x.

If $S_x^c < T_x$ for the correct class (false negative), then the weights of the active features in w^c are multiplied by a "promotion" factor $1 < p \leq p_{\max}$. On the other hand, if $S_x^c > T_x$ for the wrong class (false positive), the weights of the active features in w^c are multiplied by a "demotion" factor $d_{\min} \leq d < 1$, with $p_{\max} = 1.3$ and $d_{\min} = 0.7$. For a false positive, the weights of the active features are therefore reduced, according to $w_j^{\text{new}} = dw_j$, whereas for a false negative the weights are increased according to $w_j^{\text{new}} = pw_j$. The two multiplicative factors d and p are calculated by a mapping of the distance between score and threshold on sigmoid functions bounded by either $[d_{\min}, 1)$ or $(1, p_{\max}]$.

We also use a threshold exclusion area ("thick threshold"): if the score of a correct prediction is within a certain neighborhood of the threshold, $|S_x^c - T_x| < \epsilon T_x$, it is equally considered a mistake.[32] Consequently, the classifier not only learns upon misclassification, but also if the classification was correct but too close to the threshold. This leads to an increased separation between correct and incorrect predictions and ultimately to a more robust classifier. The "thick threshold" of $\epsilon = 0.15$ is only used with OSBs.

Laplacian-Modified Naive Bayesian Classifier. This classifier relies on Bayes' theorem of conditional probabilities:[33]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Let A be the event for which a compound is active, and B be the event for which a group of n features f_i are all present simultaneously in the compound:

$$P(B) = P(f_1 \cap f_2 \cap \cdots \cap f_n)$$

The assumption of the naïve Bayesian classifier is that all features are independent. Two events are independent if and only if $P(A \cap B) = P(A)P(B)$. It follows that, in the case of an arbitrary number m of independent events A_m, $P(A_1 \cap A_2 \cdots \cap A_m) = P(A_1)P(A_2) \cdots P(A_m)$. The conditional probability of the event of a compound being active, given that it contains a set of n features, is then

$$P(B|A) = P(A)\frac{\prod_i^n P(f_i|A)}{\prod_i^n P(f_i)}$$

Let C_i and T_i be the respective number of occurrences of feature f_i in the actives and the total dataset. The uncorrected conditional probability of a compound being active, given feature f_i, would then be

$$P(A|f_i) = \frac{C_i}{T_i}$$

For small numbers of T_i, i.e., when feature f_i is only scarcely present in the dataset, this estimate would be too large and distant from reality, where, instead, it should approach $P(A)$ to reflect the dataset. For illustration, consider the case of a feature being present four times in the entire dataset, all of them in active compounds: as a result, $P(A|f_i)$ would be 1. The corrected conditional probability $P'(A|f_i)$ introduces additional virtual samples D to alleviate this problem:[25]

$$P'(A|f_i) = \frac{C_i + DP(A)}{T_i + D}$$

If feature f_i is present very rarely, $C_i \rightarrow 0$ and $T_i \rightarrow 0$, and we find the desired result:

$$\lim_{C_i, T_i \to 0} P'(A|f_i) = \lim_{C_i, T_i \to 0} \frac{C_i + DP(A)}{T_i + D} = P(A)$$

In the Laplacian correction, $D = P(A)^{-1}$ and we find

$$P'(A|f_i) = \frac{C_i + 1}{T_i + P(A)^{-1}} = \frac{C_i + 1}{P(A)^{-1}[T_i P(A) + 1]}$$

Division by $P(A)$ yields the relative probability estimate P_{rel}:

$$P_{\text{rel}}(A|f_i) = \frac{P'(A|f_i)}{P(A)} = \frac{C_i + 1}{T_i P(A) + 1}$$

The likelihood P_{active} of a compound being active, given a set of features f_i, is calculated from the relative estimates $P_{\text{rel}}(A|f_i)$:

$$P_{\text{active}} = \prod_i P_{\text{rel}}(A|f_i)$$

To avoid potential numerical problems through the resulting very small numbers, and to have interpretable results, this is typically implemented using logarithms to yield a combined score S:

$$S = \log P_{\text{active}} = \log \prod_i P_{\text{rel}}(A|f_i) = \sum_i \log P_{\text{rel}}(A|f_i)$$

We built such a model for every single class in our dataset.

A Python script for this Laplacian-modified multiclass Bayesian classifier, including OSB and MySQL support, is available as Supporting Information, as is a comparison of our implementation with a commercial (SciTegic) one.

Figures of Merit. For both algorithms, we considered the class label corresponding to the highest score as the predicted class label. Starting from an $n \times n$ confusion matrix $Z = (z_{ij})$ for n classes, it can easily be shown that the sum of false positives $F_p = \sum_n f_p^n$ equals the sum of false negatives $F_n = \sum_n f_n^n$ (column-wise and row-wise marginals minus diagonal elements, respectively). Thus, the Matthews correlation coefficient (see below) is not an unbiased measure for the overall accuracy of multiclass classification. Therefore, we use another measure, known as accuracy, which is defined as the fraction of correct predictions (FCP):

$$\text{FCP} = \frac{\text{tr}(Z)}{N_{\text{tot}}}$$

where tr(Z) is the trace of the confusion matrix and N_{tot} is the number of total predictions made. The FCP parameter accounts for the fraction of all correct predictions (diagonal elements of the confusion matrix), with respect to the total number of predictions made.

To report the accuracy of prediction for a given class, we mainly use the Matthews correlation coefficient (MCC), which is calculated according to

$$\text{MCC} = \frac{t_{\text{p}} t_{\text{n}} - f_{\text{p}} f_{\text{n}}}{\sqrt{(t_{\text{p}} + f_{\text{p}})(t_{\text{p}} + f_{\text{n}})(t_{\text{n}} + f_{\text{p}})(t_{\text{n}} + f_{\text{n}})}}$$

where t_{p} is the number of true positives, t_{n} the number of true negatives, f_{p} the number of false positives, and f_{n} the number of false negatives.

We provide an Excel spreadsheet with detailed results of every experiment that we performed as Supporting Information. This spreadsheet includes, for every single class, for every experiment, the 15-fold Monte Carlo cross-validated figures (and standard deviations over the 15 runs) for true/false positives/negatives, positive/negative recall, positive/negative precision, MCC, and average percentage class correct (APCC).

4. RESULTS AND DISCUSSION

Overall Analysis of Experiments. To establish the performance of the two algorithms in different settings, we varied the following parameters for our experiments: (1) classification algorithm ($n_1 = 2$); (2) molecular fingerprint (CFP2, CFP3, or ECFP_4, $n_2 = 3$); (3) window size, w, for the creation of orthogonal sparse bigrams ($w = 1$ or $w = 3$, $n_3 = 2$); and (4) size of training set (30% or 70%, $n_4 = 2$). Therefore, the number of possible combinations of parameters is $n_1 \times n_2 \times n_3 \times n_4 = 24$. The cross-validated accuracies, in terms of FCP, with confidence intervals of 99% (CI 99%), for all 24 experiments are summarized in Table 2 and Figure 2.

Table 2. Accuracy of the Different Experiments, Using the Fraction of Correct Predictions (FCP) and the Corresponding Standard Deviation (sFCP)[a]

				CFP2		CFP3		ECFP_4	
method	window size, w^b	ϵ^c	%test	FCP	sFCP	FCP	sFCP	FCP	sFCP
				Group A					
Bayes	1		30	0.6575	0.0072	0.6398	0.0056	0.6753	0.0041
Winnow	1	0	30	0.6482	0.0076	0.6417	0.0091	0.6687	0.0091
Bayes	1		70	0.6567	0.0052	0.6550	0.0042	0.6794	0.0044
Winnow	1	0	70	0.6173	0.0118	0.6257	0.0097	0.6469	0.0122
				Group B					
Bayes	3		30	0.6575	0.0072	0.6398	0.0056	0.6753	0.0041
Winnow	3	0.15	30	0.6805	0.0062	0.6624	0.0062	0.6959	0.0057
Bayes	3		70	0.6567	0.0052	0.6550	0.0042	0.6794	0.0044
Winnow	3	0.15	70	0.6724	0.0048	0.6657	0.0052	0.6896	0.0056

[a] Numbers given for the fraction of correct predictions (FCP) are averages over 15 independent runs for each experiment. Groups A and B are as identified in Figure 2, where B includes OSBs. [b] Window size for the creation of OSBs. [c] "Thick threshold" parameter, which only applies to Winnow.

There are two groups (A and B) with different characteristics, as implied by the dashed vertical line in Figure 2. The difference in experimental setup between the two groups A and B is the use of orthogonal sparse bigrams (OSBs) in group B. For group A, the Bayesian classifier is almost always better than Winnow, whereas in group B the contrary is true. The inclusion of OSBs does not change the performance of the Bayesian classifier. In addition to the separation of these two groups, there are three groups, according to the fingerprints: the rank-order of the fingerprints with respect to increasing classification performance is

$$\text{CFP3} < \text{CFP2} < \text{ECFP\_4}.$$

The ECFP_4 fingerprints perform better than either of the other two fingerprints. The unhashed circular fingerprints with a radius of 3 (CFP3) perform worse in all experiments, compared to their counterpart with a radius of 2 (CFP2).

The average FCPs of all 24 experiments are within a range of 0.60−0.70; the corresponding standard deviations span the range from 0.0041 (Bayes, ECFP_4) to 0.0122 (Winnow, ECFP_4, $w = 1$). Apart from one exception, the standard deviations are always smaller for the Bayesian classifier, and they get smaller in the case of Winnow when OSBs are included. For the best-performing ECFP_4 fingerprints, all experiments lie within the very narrow interval between 0.6469 and 0.6959. The dependence of accuracy on changes in the size of the training set is conditioned by the use of OSBs: without them, Winnow suffers losses upon reduction of the training set size from 70% to 30% in all cases, whereas the Bayesian classifier is practically immune to these changes. In contrast, when OSBs are used, Winnow is more resilient to changes in the size of the training set in all cases.

When 30% of the data are used for training, the best accuracy (FCP = 0.6896) is achieved by Winnow, using ECFP_4 fingerprints and a window size of $w = 3$. The second-best result (FCP = 0.6794) is observed for the Bayesian classifier using ECFP_4 fingerprints, regardless of OSBs. Using 70% of the data for training, the best accuracy (FCP = 0.6959) is realized through the use of Winnow, ECFP_4 fingerprints, and a window size of $w = 3$.

In summary, the ECFP_4 fingerprints were determined to be superior to the others; the pair of classifiers with the highest accuracies using 70% of the data for testing were Winnow, with a window size of $w = 3$, with its counterpart being a Bayesian classifier without OSBs. Furthermore, the Bayesian classifier is immune to the exclusion of OSBs.

Influence of OSBs. The creation of OSBs increases the total number of available features. Although the Bayesian classifier does not seem to be dependent on their presence or absence, the Winnow algorithm consistently profits from their inclusion (see Figure 3). For the two experiments using ECFP_4 fingerprints and 70% of the data as training, the inclusion of OSBs results in a relative increase of 4% in the overall accuracy (FCP) for Winnow. The use of additional features through the inclusion of OSBs results in increased performance, in terms of MCC of the Winnow algorithm on all 20 classes. For certain classes in the same pair of experiments previously mentioned, the inclusion of OSBs results in considerable increases in accuracy. Examples are observed with CDK 4, CDK 5, and mGluR5: the respective increases in MCC are 44%, 43%, and 18%.

The fact that Winnow generally performs better when provided with a larger number of features was demonstrated earlier, and that was the motivation for its inventor: to separate "relevant from irrelevant attributes".[31] For all classes in all experiments reported here, the performance of Winnow was determined to improve upon the inclusion of OSBs. The Bayesian classifier yields the same results, regardless of the presence or absence of OSBs. This is indicative of the fact that, in the case of the Winnow algorithm, the inclusion of OSBs allows that algorithm to account for nonlinearity in the response function.

Analysis on a Per Class Basis. In terms of positive recall, Bayes outperforms Winnow in 14 of the 20 classes (see Figure 4). The maximum absolute difference in performance is 0.1640 for the CDK5 class (Bayes > Winnow; the relative increase, with respect to Winnow, is 29%), followed by the kappa class, where Bayes is superior by 0.1160 absolute units (the relative increase, with respect to Winnow, is 26%). For the two cholecystokinins 1 and 2 (denoted as CCK1 and CCK2, respectively), we see reversed performances: better results are achieved by Bayes for CCK1 (+14%, with respect to Winnow), whereas for CCK2, Winnow outperforms Bayes (+16%, with respect to Bayes). The Bayesian classifier retrieves 15% more compounds for the CDK1/cyclin B class than Winnow, whereas Winnow retrieves many more ligands of the mu receptor (+40%). Variations for other classes are smaller.

For positive precision, Winnow outperforms Bayes in 15 of the 20 classes. In 12 out of the 14 classes where Bayes has a higher recall of positives, it also has a lower positive precision; these classes are CA II, CCK2, CDK2/cyclin A, CDK4, CDK5, D2, Src, alpha 1a, gamma secretase, kappa, mGluR5, and mu (see Figure 5).

In terms of average MCC, there are seven classes that exhibit an overall difference of >5% for the two classifiers (with respect to the lower one of the two; see Figure 6). These seven classes are CCK2, CDK1/cyclin B, CDK4, CDK5, gamma secretase, kappa, and mu. The Bayesian classifier is better in three of them, notably, CDK1, CDK5, and kappa, with relative increases, with respect to Winnow, of 6%, 10%, and 6%, respectively. For the remaining four

Table 3. Positive Recall (R_p), Positive Precision (P_p) and Matthews Correlation Coefficients (MCC) for the Two Best Classifiers, Using 70% of the Data for Testing[a]

	Bayesian			Winnow			Bayesian−Winnow[b]		
class	R_p	P_p	MCC	R_p	P_p	MCC	R_p	P_p	MCC
alpha 1a	0.8979	0.8245	0.8514	0.8162	0.9203	0.8588	0.0817	−0.0957	−0.0074
CA II	0.9685	0.9978	0.9814	0.9924	0.9785	0.9840	−0.0239	0.0194	−0.0025
CCK1	0.6673	0.5798	0.6008	0.5857	0.6823	0.6121	0.0816	−0.1025	−0.0113
CCK2	0.6609	0.6825	0.6519	0.7691	0.6785	0.7033	−0.1082	0.0040	−0.0514
CDK1/cyclin B	0.6767	0.7164	0.6922	0.5900	0.7309	0.6520	0.0867	−0.0145	0.0401
CDK2/cyclin A	0.7891	0.8680	0.8245	0.7882	0.9031	0.8409	0.0009	−0.0351	−0.0164
CDK4	0.8333	0.6048	0.7076	0.7822	0.7157	0.7448	0.0511	−0.1110	−0.0372
CDK5	0.7333	0.5918	0.6550	0.5694	0.6283	0.5939	0.1640	−0.0365	0.0611
D2	0.9049	0.9403	0.9095	0.9663	0.8612	0.8964	−0.0614	0.0791	0.0131
delta	0.4495	0.4448	0.3751	0.4612	0.4512	0.3819	−0.0116	−0.0064	−0.0068
γ secretase	0.9870	0.7887	0.8804	0.9732	0.9325	0.9520	0.0138	−0.1438	−0.0716
kappa	0.5617	0.4004	0.4106	0.4457	0.4490	0.3877	0.1160	−0.0486	0.0230
m1	0.5868	0.6019	0.5690	0.6247	0.6038	0.5885	−0.0380	−0.0019	−0.0195
m2	0.5390	0.5579	0.5239	0.5048	0.5750	0.5136	0.0343	−0.0171	0.0103
mGluR5	0.9196	0.9491	0.9336	0.9096	0.9557	0.9315	0.0100	−0.0066	0.0020
mu	0.2592	0.3951	0.2417	0.3639	0.4050	0.2945	−0.1048	−0.0099	−0.0528
PDE4	0.9010	0.9688	0.9315	0.8925	0.9321	0.9082	0.0085	0.0367	0.0233
PDE5	0.9724	0.8694	0.9161	0.9673	0.8861	0.9227	0.0051	−0.0167	−0.0065
Src	0.8833	0.8725	0.8758	0.8473	0.9339	0.8877	0.0360	−0.0613	−0.0118
tubulin	0.9248	0.9468	0.9346	0.9175	0.9459	0.9302	0.0073	0.0009	0.0044

[a] Reported numbers are the averages over 15 independent runs for each experiment. [b] The column "Bayesian−Winnow" shows the difference of the corresponding columns shown to the left.

Table 4. Distribution of Compounds Correctly Identified Exclusively by One of the Algorithms or by Neither of Them[a]

Class	B NOT W	W NOT B	NOT (W OR B)	Class Size
alpha 1a	52 (6.9)	6 (0.8)	29 (3.9)	753
CCK1	17 (2.7)	18 (2.8)	83 (13.0)	639
CCK2	1 (0.1)	64 (8.6)	37 (5.0)	741
CDK1/cyclin B	5 (3.8)	2 (1.5)	11 (8.3)	133
CDK2/cyclin A	7 (3.3)	8 (3.8)	22 (10.4)	211
CDK4	2 (4.7)	0 (0.0)	2 (4.7)	43
CDK5	6 (11.1)	0 (0.0)	4 (7.4)	54
delta	28 (1.9)	214 (14.5)	100 (6.8)	1473
kappa	39 (3.3)	44 (3.7)	67 (5.6)	1191
m1	6 (0.8)	21 (2.7)	38 (5.0)	764
m2	25 (3.7)	13 (1.9)	64 (9.6)	670
mGluR5	2 (1.9)	0 (0.0)	4 (3.8)	105
mu	35 (2.1)	409 (24.6)	100 (6.0)	1664
PDE4	10 (1.8)	3 (0.5)	14 (2.6)	549
PDE5	2 (0.4)	1 (0.2)	5 (1.0)	490
Src	4 (2.8)	2 (1.4)	2 (1.4)	143
tubulin	1 (0.6)	2 (1.1)	2 (1.1)	180
CA II	0 (0.0)	13 (1.1)	5 (0.4)	1183
D2	0 (0.0)	38 (2.0)	19 (1.0)	1891
γ secretase	0 (0.0)	1 (0.8)	0 (0.0)	118
total	242 (1.9)	859 (6.6)	608 (4.7)	12995

[a] B = Bayes, W = Winnow. Numbers shown in parentheses are the percentages, with respect to the total class sizes.

classes above the 5% threshold in difference—CCK2, CDK4, γ secretase, and the mu receptor—Winnow has a greater accuracy, by 8%, 5%, 8%, and 22%, respectively. For eight classes, the difference is between 5% and 1% (CCK1, CDK2/cyclin A, D2, delta, m1, m2, PDE4, and Src). For the remaining five classes (alpha 1a, CA II, mGluR5, PDE5, and tubulin), the difference in average MCC is <1%.

Identification of Difficult Compounds. For the two best classifiers, we determined the number of compounds that were correctly classified at least once by one algorithm, but always incorrectly by the other. We refer to such compounds, together with those that neither algorithm correctly predicts, as "difficult compounds".

Over all 15 runs, we identified 242 compounds that were exclusively correctly predicted by Bayes but not Winnow, and also 859 compounds that were correctly predicted only by Winnow, but not Bayes. Especially for the opioid receptors (delta, kappa, and mu), as well as CCK2, Winnow identifies a considerable number of compounds that are missed by the Bayesian classifier. In total, there are 608 compounds that were not correctly classified by either of the algorithms in any of the runs. Of these, 44% are opioid receptor ligands. The distribution of difficult compounds across the classes to which they belong is summarized in Table 4 and displayed in Figure 7.

The Danger of Averages: Difficult Compounds. An examination of the overall performance statistics, such as the FCP or overall MCC, of both methods implies that they are fairly equivalent "on average". This nonetheless does not imply any equivalence in terms of the individual classes being averaged. Consider, for example, two methods giving rise to the following two populations P_1 and P_2 of four imaginary classes a, b, c, and d: $P_1 = \{a = 1, b = 2, c = 3, d = 4\}$ and $P_2 = \{a = 4, b = 3, c = 2, d = 1\}$. Both averages are 2.5 (= 10/4), although, on a per-class basis, the populations are antisymmetric. If only the averages are considered, one would necessarily conclude that both methods are equivalent. This extreme case is not observed for the two classifiers that we have compared, but it does serve to illustrate that the drawing of conclusions from just one overall statistic may be inappropriate. Therefore, it is advisable not to rely on one method alone.

In all of our experiments, most classes have similar values for all figures of merit that we calculated. For certain pairs of classes, however, we see that Winnow has a higher MCC than Bayes for one of them, with this order being reversed for the other class. This was observed for the recall of compounds of the classes CCK1 (Bayes is better) and CCK2

Table 5. Details about the Difficult Compounds for the CDK1/Cyclin B Class

algorithm	failure[a]	ID[b]	first correct[c]	second correct[c]	most frequent first[d]	d_{12}[e]
Bayes	bnw	6479	1	2	CDK5 (7)	10.76
Bayes	bnw	6971	3	0	CDK2/cyclin A (8)	4.62
Bayes	bnw	7720	1	4	CDK5 (6)	55.18
Bayes	bnw	8143	5	3	CDK2/cyclin A (6)	10.33
Bayes	bnw	5658	3	1	CDK4 (4)	8.82
Bayes	wnb	7210	0	9	CDK5 (11)	66.85
Bayes	wnb	12092	0	4	CDK5 (5)	80.86
Bayes	never	4536	0	1	CDK5 (8)	62.91
Bayes	never	4771	0	10	CDK2/cyclin A (10)	9.79
Bayes	never	7206	0	10	CDK5 (13)	54.98
Bayes	never	7213	0	6	CDK5 (9)	63.23
Bayes	never	7215	0	7	CDK5 (11)	62.68
Bayes	never	12149	0	1	CDK2/cyclin A (7)	3.86
Bayes	never	7987	0	0	kappa (5)	3.73
Bayes	never	8981	0	5	CDK5 (10)	52.18
Bayes	never	12577	0	0	CDK4 (6)	82.18
Bayes	never	7208	0	6	CDK5 (10)	65.50
Bayes	never	12575	0	1	CDK4 (10)	73.48
Winnow	bnw	6479	0	0	D2 (5)	2.48
Winnow	bnw	6971	0	1	CDK2/cyclin A (12)	8.85
Winnow	bnw	7720	0	1	alpha 1a (5)	1.02
Winnow	bnw	8143	0	7	CA II (11)	12.93
Winnow	bnw	5658	0	3	D2 (7)	15.26
Winnow	wnb	7210	1	3	CDK5 (10)	18.81
Winnow	wnb	12092	1	3	CDK5 (4)	16.25
Winnow	never	4536	0	0	CDK5 (7)	11.64
Winnow	never	4771	0	4	CDK2/cyclin A (10)	11.90
Winnow	never	7206	0	3	CDK5 (13)	14.99
Winnow	never	7213	0	2	CDK5 (9)	15.11
Winnow	never	7215	0	2	CDK5 (11)	17.13
Winnow	never	12149	0	0	CDK2/cyclin A (8)	7.57
Winnow	never	7987	0	0	D2 (4)	3.36
Winnow	never	8981	0	1	CDK5 (9)	13.86
Winnow	never	12577	0	0	CDK4 (6)	26.26
Winnow	never	7208	0	4	CDK5 (10)	18.40
Winnow	never	12575	0	0	CDK4 (9)	20.80

[a] Legend: bnw, Bayes NOT Winnow; wnb, Winnow NOT Bayes; and never, NOT (Bayes OR Winnow). [b] ID denotes the identifier of the compound; see Figure 9. [c] The terms "1st correct" and "2nd correct" refer to how often the first- and second-ranked predictions were correct, respectively. [d] The term "most frequent first" shows the class label that was predicted most often, with the number of occurrences given in parentheses. [e] The term "d_{12}" represents the average difference between the two top-ranking scores for those predictions where the top-ranking class is the "most frequent first" class.

Table 6. Details about the Difficult Compounds for the PDE5 Class

algorithm	failure[a]	ID[b]	first correct[c]	second correct[c]	most frequent first[d]	d_{12}[e]
Bayes	bnw	1925	4	0	PDE4 (5)	10.31
Bayes	bnw	12838	3	4	PDE4 (4)	42.41
Bayes	wnb	4495	0	0	CDK4 (6)	5.21
Bayes	never	3043	0	10	PDE4 (10)	172.86
Bayes	never	3855	0	2	tubulin (12)	10.61
Bayes	never	3856	0	0	delta (9)	3.61
Bayes	never	12112	0	6	tubulin (10)	14.50
Bayes	never	3035	0	8	PDE4 (8)	112.84
Winnow	bnw	1925	0	2	Src (9)	6.02
Winnow	bnw	12838	0	3	CA II (4)	15.25
Winnow	wnb	4495	1	3	PDE4 (7)	1.35
Winnow	never	3043	0	3	PDE4 (10)	68.69
Winnow	never	3855	0	2	tubulin (8)	3.75
Winnow	never	3856	0	0	tubulin (9)	3.40
Winnow	never	12112	0	2	tubulin (11)	2.81
Winnow	never	3035	0	3	PDE4 (8)	48.40

[a] Legend: bnw, Bayes NOT Winnow; wnb, Winnow NOT Bayes; and never, NOT (Bayes OR Winnow). [b] ID denotes the identifier of the compound; see Figure 9. [c] The terms "1st correct" and "2nd correct" refer to how often the first- and second-ranked predictions were correct, respectively. [d] The term "most frequent first" shows the class label that was predicted most often, with the number of occurrences given in parentheses. [e] The term "d_{12}" represents the average difference between the two top-ranking scores for those predictions where the top-ranking class is the "most frequent first" class.

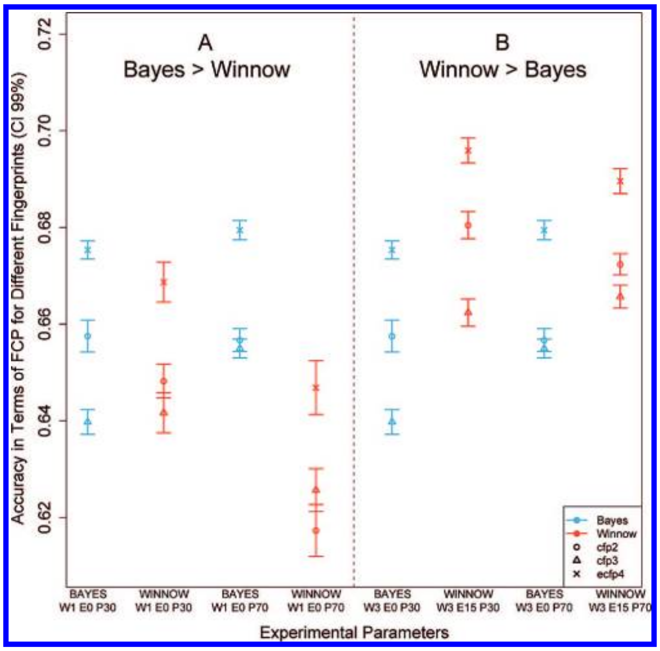

Figure 2. Comparison of all experiments. For experiments with 70% as the test set (experiments including label "P70"), almost 140 000 predictions are consolidated into one point with its accompanying error bars, the equivalent for experiments with the smaller test set of only 30% (experiments including label "P30") is just below 60 000.

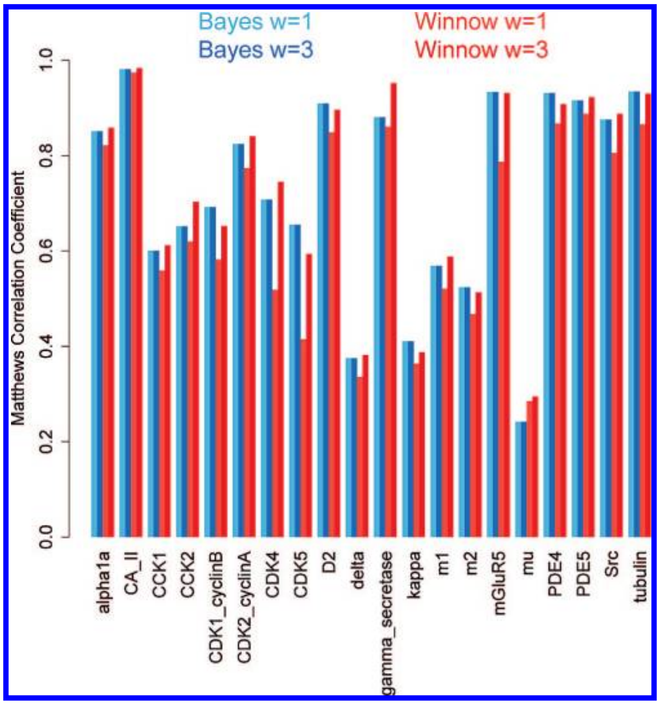

Figure 3. Matthews correlation coefficients using ECFP_4 fingerprints with and without additional orthogonal sparse bigrams (OSBs) obtained using 70% of the data as the test set.

(Winnow is better), and the precision of classes PDE4 (Bayes is better) and PDE5 (Winnow is better).

We have illustrated, with a simple example, that two anticorrelated sequences of class populations may lead to the same overall statistics. If one now additionally considers the fact that the compounds tested in each single run are drawn randomly from an underlying dataset, there are more implications that must be taken into account, as we show in the following paragraphs.

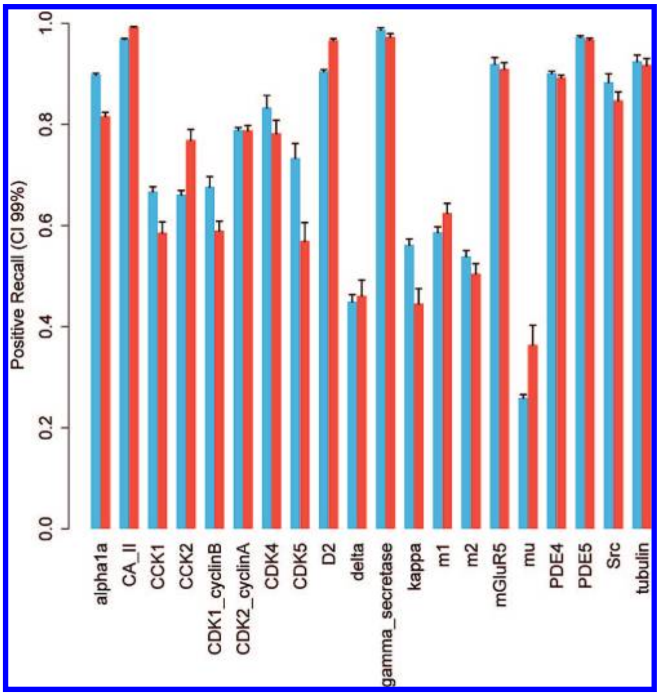

Figure 4. Comparison of positive recall for the two best classifiers. Results shown are averages of 15 independent runs, along with their 99% confidence intervals. (Bayes = blue, Winnow = red.)

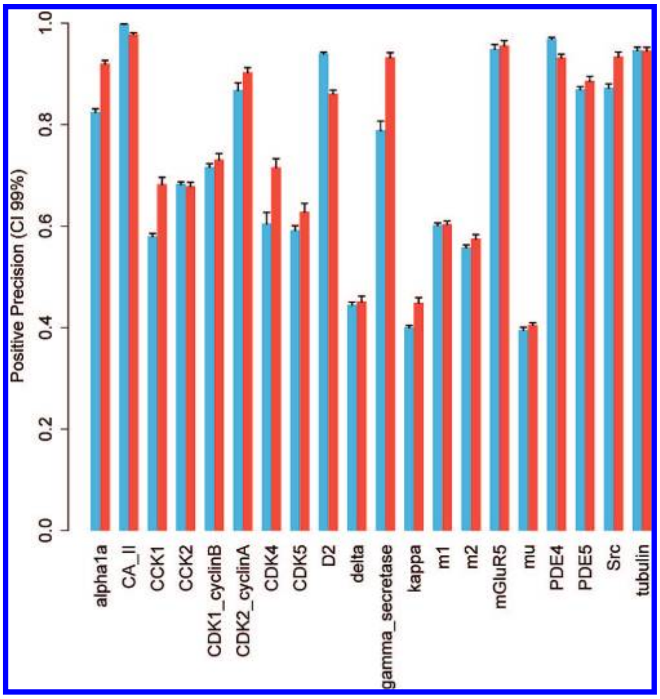

Figure 5. Comparison of positive precision for the two best classifiers. Results shown are averages of 15 independent runs, along with their 99% confidence intervals. (Bayes = blue, Winnow = red.)

We first checked how many unique compounds have occurred in the test set in any of the 15 runs. For both experiments, this was 12 995, i.e., every single compound was used at least once as a test example. Given that every compound is selected with a probability of $P(S) = 0.7$ for testing in each run, the probability of it being sampled at least five times in these 15 runs is already practically equivalent to unity. The average sampling number of all compounds was experimentally determined to be 10.5 (± 0.01). These numbers imply that we can be fairly confident

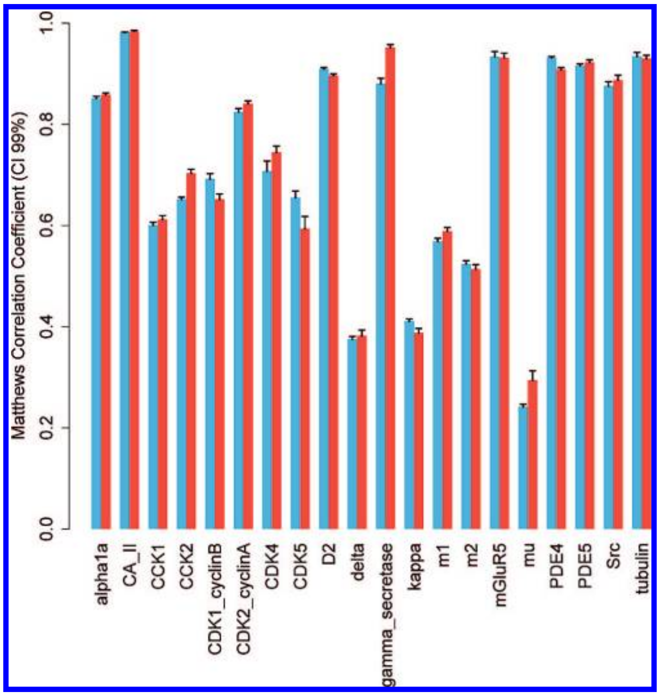

Figure 6. Comparison of Matthews correlation coefficient (MCC) for the two best classifiers. Results shown are averages of 15 independent runs, along with their 99% confidence intervals.

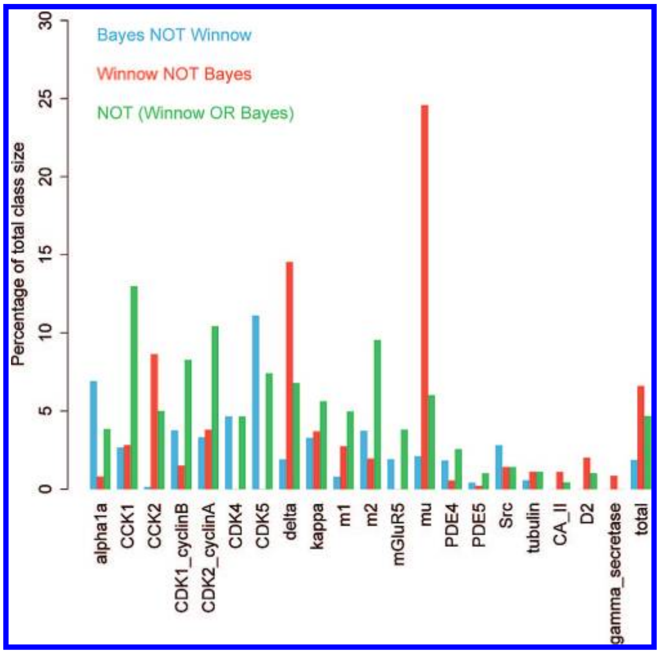

Figure 7. Distribution of "difficult compounds", which are defined to be compounds correctly identified exclusively by either Bayes or Winnow, or by neither of the two algorithms. Results presented are percentages with respect to the total size of the corresponding classes (see Table 4). (Bayes = blue, Winnow = red.)

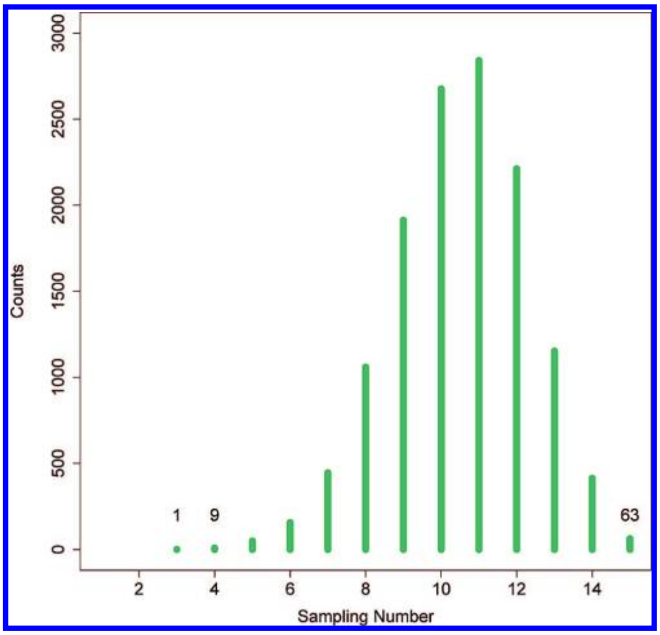

Figure 8. Distribution of the sampling of 9097 compounds in the test set for 15 independent runs. The average sampling number is 11, but significantly lower sampling numbers also are observed.

that inappropriate sampling for either of the methods is not the reason for the observed set differences of compounds that were classified correctly at least once. Our analysis of the selection of compounds for testing confirmed that each compound was used 10.5 times (±0.01) *on average*, as would necessarily be expected. For each of the 15 runs, 70% of all compounds ($0.7N$) are chosen. Thus, the total number of test compound choices made over all runs is ($15 \times 0.7N$) = $10.5N$. The mean number of times a compound is selected for the test set is therefore 10.5, subject to a small error arising from the cardinality of the dataset not being a multiple of 10.

A probability is an estimate for the relative frequency of a specific outcome given a large pool of possible outcomes. However, a probability of 0.7 of being selected for the test set does not mean that every compound is actually found in exactly 70% of all test sets. The distribution of the sampling of compounds in the test set shows that, indeed, the average is between 10 and 11, but significantly lower sampling numbers occur occasionally (see Figure 8). By symmetry, the same is true for the training set. An obvious source for certain compounds being consistently predicted incorrectly may be that a subgroup of a specific activity class exhibiting certain structural features is predominantly not sampled as part of the training set. As a direct result, the algorithm would not be able to learn the required features to classify test compounds issued from this class correctly. To test this hypothesis, we determined the occurrence in the training set of those compounds always predicted with the wrong class label. As implied by the similarity principle, the inclusion of a compound in the training set should help to classify similar compounds correctly.[36] Therefore, if many of the compounds consistently predicted incorrectly are determined to be less frequently part of the training set, the problem could be solved by changing (eventually deliberately biasing) the sampling procedure. On the other hand, should that not be the case, there must be certain specific structural features that prevent these compounds from being classified correctly.

For the 608 compounds that were always predicted incorrectly, the average sampling number in the test sets was determined to be 10.38, which is only very slightly lower than the expected value of 10.5. For the 242 compounds predicted correctly exclusively by the Bayesian classifier, 26% (63) of them were present more than 11 times in the test sets and 71% (172) had been sampled 9−12 times (i.e., within one standard deviation of ≈1.8 of the average sampling number). In the case of the 859 compounds

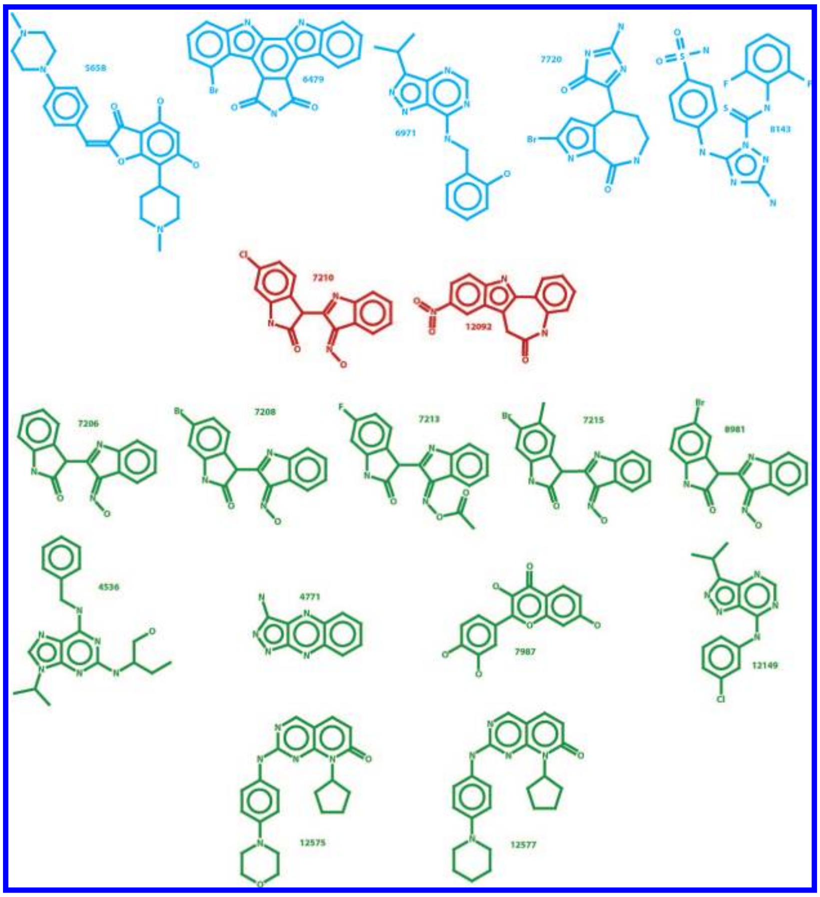

Figure 9. The "difficult" CDK1/cyclin B compounds. Blue compounds represent those compounds that were exclusively predicted correctly by Bayes, red compounds were exclusively predicted correctly by Winnow, and green compounds were never predicted correctly.

exclusively predicted correctly by Winnow, 28% (237) were present more than 11 times in the test sets and 73% (626) were within one standard deviation. This means that the sampling of the compounds that each algorithm retrieves exclusively is comparable. The corresponding numbers for the training sets follow immediately, because of complementarity to the test sets, and we conclude that the compounds that exhibit difficulties are represented equally well in the training sets, relative to any other compound.

The analysis described in the previous paragraph shows that there are no significant differences in the sampling of those compounds which are determined to be particularly difficult to predict for either or both of the two algorithms under scrutiny. Therefore, we conclude that the likely reasons for the observed difficulties are, instead, attributable to structural features that distinguish these compounds in different ways, relative to the algorithms.

We chose two classes with a small number of total cases of exclusive success and joint failure in prediction. For class CDK1/cyclin B, a member of the cyclin-dependent kinases, 5 compounds (3.8%) are correctly classified at least once by the Bayesian method, but never by Winnow; 2 compounds (1.5%) are classified correctly at least once by Winnow, but never using the Bayesian algorithm; and for 11 compounds (8.3%), both algorithms never produce the correct answer (see Table 4). The chemical structures of these compounds are shown in Figure 9. Similarly, for each of the Bayesian and Winnow classifiers, there are two (0.4%) or one (0.2%) phosphodiesterase 5 (PDE5) compounds, respectively, which they exclusively retrieve at least once, and five (1.0%) which are never correctly identified using either method (see Figure 10).

For most of the CDK1/cyclin B compounds, one of the two top-ranking predictions is correct a considerable number of times for both algorithms. The top-ranking predicted class labels may well be wrong, although both algorithms provide classes closely related to the true class as most frequent top-ranking solutions very consistently. There is a strong agreement in the incorrect solution that is most frequently assumed by both algorithms to be the correct one: they disagree only for five compounds (ID Nos. 6479, 7720, 8143, 5658, and 7987; see Table 5). Moreover, this consistency in prediction of the incorrect, although closely related, class may well be the translation of biologically observed selectivity differences. The same comments can be made about the phosphodiesterases; the equivalent set of numbers is summarized in Table 6.

Note that the arguments presented in the last paragraph do not universally favor any algorithm over the other, neither in terms of how often the second-ranked prediction is correct, nor in the difference d_{12} of the scores of the two most likely predictions (see Tables 5 and 6). A large difference between the two top-ranking scores is an indication of the separation between these predictions. Should it be small, the algorithm scores both predictions almost equally, which suggests a lack of discriminatory power for that particular test example. Note that, here, the difference d_{12} may be large, but the predicted label still may be incorrect: in that case, the prediction can

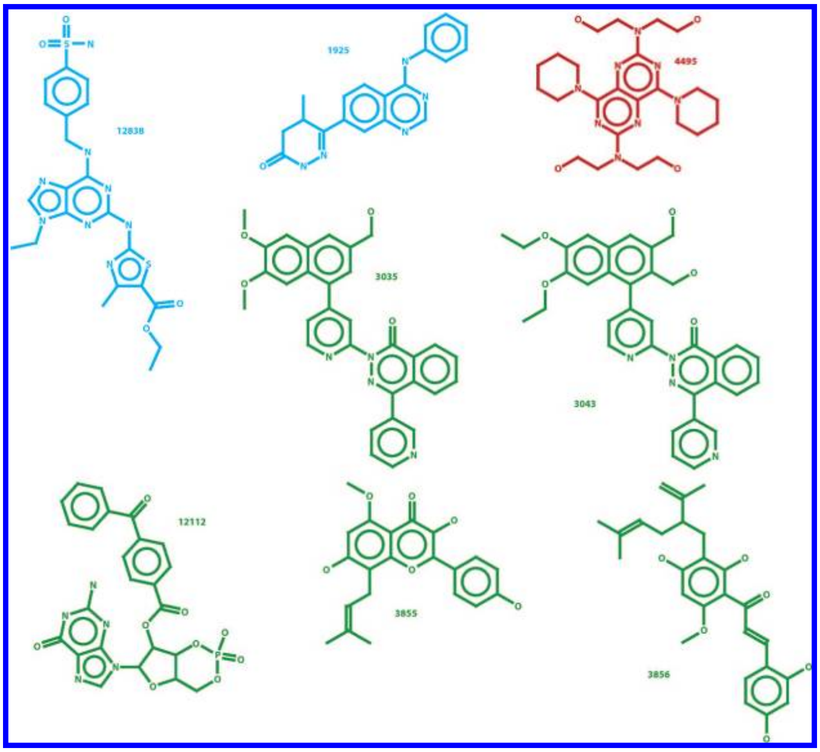

Figure 10. The "difficult" PDE5 compounds. Blue compounds represent those compounds that were exclusively predicted correctly by Bayes, red compounds were exclusively predicted correctly by Winnow, and green compounds were never predicted correctly.

be considered more incorrect than if d_{12} was smaller. A more detailed analysis of the range and average values for d_{12} in the case of different algorithms may yield "confidence estimates" in the future.

5. CONCLUSIONS

The comparison of two methods that can be used for virtual screening—the Bayesian classifier and the Winnow algorithm—showed similar performance when averaged over multiple classes and multiple runs. However, a class-by-class analysis showed important differences in performance of the methods, which would go unnoticed if only figures for overall accuracy were used for comparison.

Of the three different fingerprints that we used, the ECFP_4 fingerprints consistently performed the best, although the difference for the other two fingerprints used was generally <5%. Whereas the inclusion of orthogonal sparse bigrams (OSBs) does not change the performance of the Bayesian classifier, the Winnow algorithm consistently performs better upon their inclusion. Moreover, their inclusion results in a reduction of the standard deviations of the predictions in the case of the Winnow algorithm. Our solution for the ordering of the features prior to the creation of OSBs is perhaps not the ideal one and that may become the subject of future investigations.

In a further examination of our results, we determined individual compounds that either one or both methods applied were repeatedly unable to classify correctly. We found 242 compounds that only the Bayesian classifier was able to classify correctly at least once, whereas the corresponding number for Winnow was determined to be 859. There were also 608 compounds that were never classified correctly.

We anticipate that a further analysis of those difficult compounds and their individual molecular features may well reveal information that can be incorporated into the underlying classification algorithms to improve predictive accuracy. Similarly, an identification of the features of compounds repeatedly misclassified may allow the determination of subsets of a screening library that are likely to be misclassified. A closer examination of such subsets using different methods and complementary information could lead to a reduction of false predictions. Moreover, the identified complementarities of the two methods could be used to develop consensus models: both methods could be used concurrently, giving the prediction of one method more weight in classes where it is proven to outperform the other. This also entails a deeper up-front analysis of the confidence that may be given to individual predictions. As mentioned previously, one possibility to obtain such confidence estimates may be through an in-depth analysis of the separation of the top-ranking predictions.

ACKNOWLEDGMENT

F.N. and J.B.O.M. thank Unilever for funding. A.B. acknowledges a Presidential Postdoctoral Fellowship of the Novartis Institutes for BioMedical Research.

Supporting Information Available: Excel spreadsheets containing all per-class figures of merit for all experiments; comparison of SciTegic's and our in-house implementation of the Bayesian classifier on a determined partition of the data; Python script used for the calculation of circular fingerprints; our implementations of the Winnow algorithm in C++ and of the Naive Bayesian classifier in Python, as used in this paper. This information is available free of charge via the Internet at http://pubs.acs.org/.

REFERENCES AND NOTES

(1) Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *J. Chem. Inf. Model.* **2006**, *46*, 462–470.

(2) Keiser, M. J.; Roth, B. L.; Armbruster, B. N.; Ernsberger, P.; Irwin, J. J.; Shoichet, B. K. Relating protein pharmacology by ligand chemistry. *Nat. Biotechnol.* **2007**, *25*, 197–206.

(3) Sheridan, R. P.; Kearsley, S. K. Why do we need so many chemical similarity search methods? *Drug Discovery Today* **2002**, *7*, 903–911.

(4) Bender, A.; Glen, R. C. Molecular similarity: a key technique in molecular informatics. *Org. Biomol. Chem.* **2004**, *2*, 3204–3218.

(5) Senese, C. L.; Duca, J.; Pan, D.; Hopfinger, A. J.; Tseng, Y. J. 4D-fingerprints, universal QSAR and QSPR descriptors. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1526–1539.

(6) Hughes, L. D.; Palmer, D. S.; Nigsch, F.; Mitchell, J. B. O. Why Are Some Properties More Difficult To Predict than Others? A Study of QSPR Models of Solubility, Melting Point, and Log *P*. *J. Chem. Inf. Model* **2008**, *48*, 220–232.

(7) MDL MACCS Keys; Elsevier MDL: San Ramon, CA, 2008.

(8) Sybyl, version 7; Tripos: St. Louis, MO, 2007.

(9) Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Molecular similarity searching using atom environments, information-based feature selection, and a naive Bayesian classifier. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 170–178.

(10) Glen, R. C.; Bender, A.; Arnby, C. H.; Carlsson, L.; Boyer, S.; Smith, J. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. *IDrugs* **2006**, *9*, 199–204.

(11) Rogers, D.; Brown, R. D.; Hahn, M. Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up. *J. Biomol. Screen.* **2005**, *10*, 682–686.

(12) Arodz, T.; Yuen, D. A.; Dudek, A. Z. Ensemble of linear models for predicting drug properties. *J. Chem. Inf. Model.* **2006**, *46*, 416–423.

(13) Cannon, E. O.; Amini, A.; Bender, A.; Sternberg, M. J. E.; Muggleton, S. H.; Glen, R. C.; Mitchell, J. B. O. Support vector inductive logic programming outperforms the naive Bayes classifier and inductive logic programming for the classification of bioactive chemical compounds. *J. Comput.-Aided. Mol. Des.* **2007**, *21*, 269–280.

(14) Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32.

(15) Wilton, D. J.; Harrison, R. F.; Willett, P.; Delaney, J.; Lawson, K.; Mullier, G. Virtual screening using binary kernel discrimination: analysis of pesticide data. *J. Chem. Inf. Model.* **2006**, *46*, 471–477.

(16) Mahé, P.; Ralaivola, L.; Stoven, V.; Vert, J. P. The pharmacophore kernel for virtual screening with support vector machines. *J. Chem. Inf. Model.* **2006**, *46*, 2003–2014.

(17) Molnar, L.; Keseru, G. M. A neural network based virtual screening of cytochrome P450 3A4 inhibitors. *Bioorg. Med. Chem. Lett.* **2002**, *12*, 419–421.

(18) Breiman, L.; Friedman, J.; Stone, C. J.; Olshen, R. In *Classification and Regression Trees*, 1st Edition; CRC Press LLC: Boca Raton, FL, 1984.

(19) Jorissen, R. N.; Gilson, M. K. Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.* **2005**, *45*, 549–561.

(20) Agrafiotis, D. K.; Cedeno, W. C.; Lobanov, V. S. On the use of neural network ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 903–911.

(21) Lagunin, A.; Stepanchikova, A.; Filimonov, D.; Poroikov, V. PASS: prediction of activity spectra for biologically active substances. *Bioinformatics* **2000**, *16*, 747–748.

(22) Poroikov, V.; Filimonov, D.; Lagunin, A.; Gloriozova, T.; Zakharov, A. PASS: identification of probable targets and mechanisms of toxicity. *SAR QSAR Environ. Res.* **2007**, *18*, 101–110.

(23) Fliri, A. F.; Loging, W. T.; Thadeio, P. F.; Volkmann, R. A. Biological spectra analysis: Linking biological activity profiles to molecular structure. *Proc. Natl. Acad. Sci., U.S.A.* **2005**, *102*, 261–266.

(24) Bender, A.; Jenkins, J. L.; Glick, M.; Deng, Z.; Nettles, J. H.; Davies, J. W. "Bayes affinity fingerprints" improve retrieval rates in virtual screening and define orthogonal bioactivity space: when are multitarget drugs a feasible concept. *J. Chem. Inf. Model.* **2006**, *46*, 2445–2456.

(25) Nidhi; Glick, M.; Davies, J. W.; Jenkins, J. L. Prediction of biological targets for compounds using multiple-category Bayesian models trained on chemogenomics databases. *J. Chem. Inf. Model* **2006**, *46*, 1124–1133.

(26) Olah, M.; Mracec, M.; Ostopovici, L.; Rad, R.; Bora, A.; Hadaruga, N.; Olah, I.; Banda, M.; Simon, Z.; Oprea, T. I. WOMBAT: World of Molecular Bioactivity. In *Cheminformatics in Drug Discovery*; Oprea, T. I., Ed.; Wiley−VCH: New York, 2004; pp 223−239.

(27) Young, D. W.; Bender, A.; Hoyt, J.; McWhinnie, E.; Chirn, G.; Tao, C. Y.; Tallarico, J. A.; Labow, M.; Jenkins, J. L.; Mitchison, T. J.; Feng, Y. Integrating high-content screening and ligand-target prediction to identify mechanism of action. *Nat. Chem. Biol.* **2008**, *4*, 59–68.

(28) Jenkins, J. L.; Bender, A.; Davies, J. W. In silico target fishing: Predicting biological targets from chemical structure. *Drug Discovery Today: Technol.* **2006**, *3*, 413–421.

(29) Siefkes, C.; Assis, F.; Chhabra, S.; Yerazunis, W. S. Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering. In *Proceedings of the European Conference on Principle and Practice of Knowledge Discovery in Databases*, 2004; Springer: Berlin, 2004; pp 410−421.

(30) Nigsch, F.; Mitchell, J. B. O. How To Winnow Actives from Inactives: Introducing Molecular Orthogonal Sparse Bigrams (MOSBs) and Multiclass Winnow. *J. Chem. Inf. Model.* **2008**, *48*, 306–318.

(31) Littlestone, N. Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm. *Machine Learning* **1988**, *2*, 285–318.

(32) Dagan, I.; Karov, Y.; Roth, D. Mistake-driven learning in text-categorization. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics (ACL): Somerset, NJ, 1997; pp 55−63.

(33) Stirzaker, D. In *Probability and Random Variables*; Cambridge University Press: Cambridge, U.K., 1999.

(34) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **2001**, *46*, 3–26.

(35) R Development Core Team *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2008.

(36) Kubinyi, H. Molecular similarity. 2. The structural basis of drug design. *Pharm. Unserer Zeit* **1998**, *27*, 158–172.

CI800079X