CHEMBIOCHEM

Classifying "Kinase Inhibitor-Likeness" by Using Machine-Learning Methods

Hans Briem* and Judith Günther[a]

By using an in-house data set of small-molecule structures, encoded by Ghose–Crippen parameters, several machine learning techniques were applied to distinguish between kinase inhibitors and other molecules with no reported activity on any protein kinase. All four approaches pursued—support-vector machines (SVM), artificial neural networks (ANN), k *nearest neighbor classification with GA-optimized feature selection (GA/kNN), and recursive partitioning (RP)—proved capable of providing a reasonable discrimination. Nevertheless, substantial differences in performance among the methods were observed. For all techniques tested, the use of a consensus vote of the 13 different models derived improved the quality of the predictions in terms of accuracy, precision, recall, and F1 value. Support-vector machines, followed by the GA/kNN combination, outperformed the other techniques when comparing the average of individual models. By using the respective majority votes, the prediction of neural networks yielded the highest F1 value, followed by SVMs.*

Introduction

With the increasing impact of genomics and related technologies on the drug-discovery process, the focus of interest in the pharmaceutical industry has shifted over the past years from individual targets to target families.[1] Considerable synergies can be exploited if the knowledge gathered in individual projects on related target proteins is seen in the larger context of the target family and organized in a way that supports easy knowledge transfer to further members of the target family. This reorganization in the drug-discovery process has basically affected all functions involved.[2,3] This includes medicinal chemistry, which has put substantial effort into the development of elaborate strategies for synthesizing so-called privileged scaffolds[4] for all major target classes of interest. Compound series built with/around such scaffolds are particularly suited for providing inhibitors for the respective target family and are thus preferentially screened whenever new inhibitors for another target family member are to be identified.

One of the major target families that have attracted attention in recent years is protein kinases. The human genome is estimated to contain about 520 protein kinases.[5] Since protein kinases play a major role in crucial cellular events, such as signal transduction processes, they provide feasible targets for a number of different indications, the most important arguably being oncology.[6,7] The vast majority of efforts in drug development have aimed for kinase inhibitors that target the ATP-binding pocket of kinases, which represents a deep and narrow cleft between the C- and N-terminal subdomains.[8] Given that all protein kinases bind an ATP molecule in this pocket, it is not surprising that initial hits often target related kinases as well, and selectivity problems are often encountered in such projects.[9,10] Although essential recognition features, which are highly conserved among protein kinase pockets are often involved in inhibitor binding, the variety of chemical scaffolds that appear suitable for yielding potent kinase inhibitors is still surprisingly wide. One reason for this is that, depending on the individual kinase, only part of the interactions that the ATP molecule forms have to be mimicked by a potent inhibitor. Secondly, the pronounced flexibility of kinase pockets[11,12] facilitates the identification of selective inhibitors, since the (mostly unexpected) opening up of further subpockets provides additional possibilities for strong interactions with the protein. While intermolecular recognition processes clearly take place in three-dimensional space, the expertise that medicinal chemists have gathered over the years has given them a fairly good sense of the features in molecular structures that are important for kinase inhibition. Thus, even the 2D structure of a given molecule allows for some estimate of its usefulness in kinase inhibitor research. Accordingly, we tried to condense some of this empirical knowledge about kinase inhibitors in a computational prediction method. By using supervised-learning techniques, similarly complex classification problems have been successfully addressed with 2D molecular descriptors in the past. Most noteworthy are the pioneering studies of Sadowski and Kubinyi[13] as well as those of Ajay et al.,[14] who trained different neural networks for the classification of drugs and nondrugs ("druglikeness" filtering). More recently, Manallack et al.[15] showed that neural networks along with BCUT-parameters as input descriptors allow for the classification of compounds that are active against biological targets that belong to specific gene families versus a set of randomly selected molecules.

Encouraged by these results, our aim in the study presented here was to come up with fast and easily applicable in silico filter tools that can capture essential features of kinase inhibitor molecules, as opposed to druglike molecules that only hit other target families. At the same time, these tools must be

[a] Dr. H. Briem, Dr. J. Günther
Schering AG, Research Center Europe
CDCC/Computational Chemistry
Muellerstraße 178, 13342 Berlin (Germany)
Fax: (+49) 30-468-94030
E-mail: hans.briem@schering.de

DOI: 10.1002/cbic.200400109

promiscuous enough to cover qualities of inhibitors against *any* protein kinase rather than a particular one. We have applied several machine-learning techniques, including support-vector machines (SVM), artificial neural networks (ANN), *k* nearest neighbor classification with GA-optimized feature selection (GA/*k*NN), and recursive partitioning (RP), to a data set extracted from our in-house library and examined how well they perform in distinguishing kinase inhibitors from non-kinase inhibitors. These tools can be highly valuable whenever a large data set of molecules is to be screened in order to select structures that have a higher likelihood of being kinase inhibitors. Such molecules are often desired/demanded in order to enrich in-house target-specific libraries. Independent of whether these molecules are to be purchased from vendors or to be synthesized by combinatorial chemistry, an empirically derived in silico tool can help to set priorities within the list of accessible molecules.

Computational Methods

Data sets and descriptors

The data set used for training and testing the systems is comprised of both kinase-active and -inactive compounds from the Schering library. A compound is labeled "active" if it shows $IC_{50} \leq 10\,\mu\text{M}$ in at least one of eight in-house kinase assays (three Ser/Thr kinases, five Tyr kinases). This activity threshold was chosen since it represents an internal standard for starting follow-up activities after high-throughput screening.

The entire pool of training compounds contains 565 molecules classified as "actives" and 7194 as "inactives". In addition, an independent test set of 504 compounds (204 actives and 300 inactives) was put aside for validation purposes. Compounds were randomly assigned to either the training set or the test set.

A scaffold-based clustering of the entire data set that used Bioreason's ClassPharmer program[16] yielded 1112 unique scaffolds (642 clusters and 470 singletons). The active compounds fall into 154 of these clusters while 37 of them are classified as singletons. Each of the "active" clusters also contains at least one inactive compound. Accordingly, the discrimination of kinase-active and -inactive compounds cannot simply be achieved by a classification of the molecular scaffolds contained in the data set.

In addition, the classifiers were tested on a data set of ten kinase inhibitors taken from recent reviews[17,18] (see Table 1). None of these inhibitors exhibits significant 2D similarity to any of the compounds in the training set. For all cases, the Tanimoto-similarity coefficient to their respective nearest neighbors in the training set is <0.7, based on the 166 public MACCS keys.[19]

Due to their proven success in categorizing compounds and their ease of calculation and interpretability, we utilized the fragment-based descriptors developed by Ghose and Crippen.[20] To encode the molecule structures, we employed a SYBYL spl script, which counts the number of occurrences of each of the 120 Ghose–Crippen fragments in a given molecule, thus yielding a vector of 120 integers for each compound in the data set. In order to avoid a bias in descriptor space and numerical problems, we linearly scaled the descriptors to the range $[-1, +1]$.

Classification and model validation

It is well known that some machine-learning methods have difficulties handling unbalanced training sets, in which the number of positive examples is substantially different from the number of negatives (typically much smaller). Therefore, to create balanced subsets from the whole data given, we employed an ensemble-based sampling procedure, similar to the method proposed by Yan et al.[21] The overall architecture of this ensemble approach is depicted in Figure 1.

As the ratio of inactives to actives in our set is about 13:1, we generated thirteen different, individually balanced, training sets. Each member of the training set ensemble is made up of all 565 active compounds as well as the same number of inactives, randomly selected, with replacement from the entire pool of 7194 inactives. Accordingly, on average every inactive compound has the chance to contribute once to the model training.

Models are derived from each training set independently and used for class prediction of the compounds in the external test set. In the Results and Discussion section, we report the predictive power of each individual training set as well as the results from a consensus majority vote of all members of the ensemble. For example, if seven out of the 13 models classified a test compound as active and the other six models voted for inactivity, then the consensus majority vote would classify it as being active.

There are many possible ways to assess the performance of a classifier, with *accuracy*, *precision*, and *recall* probably being the most widely used measures. Their definitions are given in Equations (1)–(3), where tp = number of true positives, tn = number of true negatives, fp = number of false positives, and fn = number of false negatives.

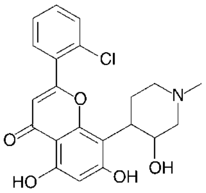
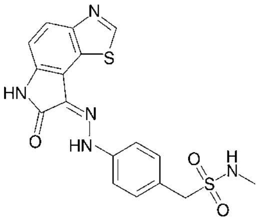
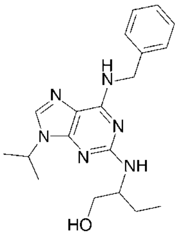
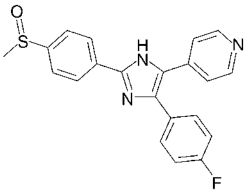
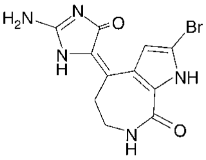
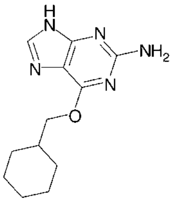
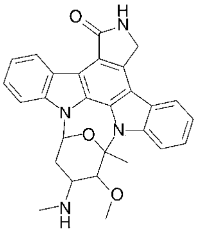
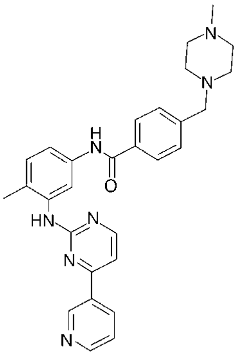
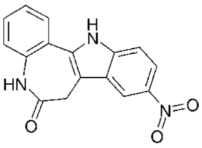
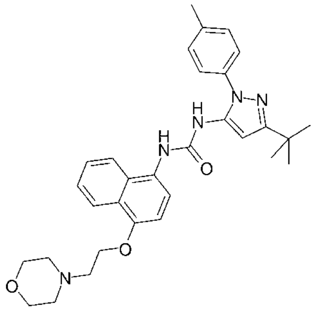
$$\text{accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

$$\text{precision} = \frac{tp}{tp + fp} \tag{2}$$

$$\text{recall} = \frac{tp}{tp + fn} \tag{3}$$

While accuracy is a simple and useful measure for the *overall* classification performance, precision—that is, the ability to predict a particular class correctly—and recall—that is, the ability to pick the true members of a class from a data set—can only be reasonably interpreted in combination with each other. For example, the compounds predicted to be active might in fact all be true actives (precision = 1.0), while at the same time many of the other true actives in the data set might be misclassified as inactives (low recall). On the other hand, a high

Table 1. Correct class predictions on the set of known kinase inhibitors.

Kinase Inhibitor (Cmpd. No.)	Structure	SVM	ANN	GA/*k*NN	RP	Kinase Inhibitor (Cmpd. No.)	Structure	SVM	ANN	GA/*k*NN	RP
flavopiridole **1**		true	true	true	true	GW9499 **6**		true	true	false	false
roscovitine **2**		true	true	true	true	SB203580 **7**		true	true	true	false
hymenialdisine **3**		true	true	false	false	NU2058 **8**		true	true	true	true
staurosporine **4**		true	true	true	true	Gleevec **9**		true	true	true	true
alsterpaullone **5**		true	false	true	true	BIRB796 **10**		false	false	false	false
						Number of correct predictions:		9/10	8/10	7/10	6/10

recall, that is, a correct prediction for most of the true actives, might be accompanied by a low precision, that is, many false positives. To mitigate this problem, we calculated the harmonic mean of precision and recall, also known as F1 measure or F-score[22] [Eq. (4)]. Ideally, if both precision and recall are high, this measure assumes values close to one.

$$F1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{4}$$

As we are primarily interested in correctly classifying the actives, only the above-mentioned success measures for this class, along with the overall accuracy, are reported.

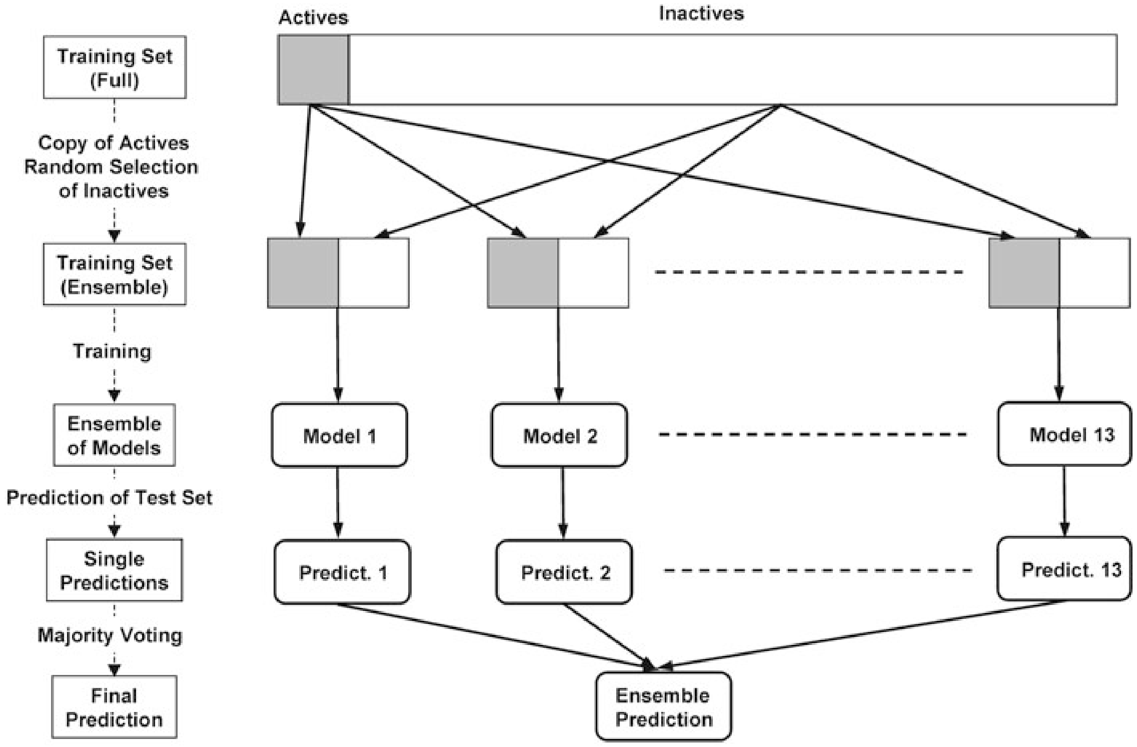


Figure 1. *Architecture and workflow of the ensemble-based sampling and voting procedure.*

Machine-learning methods

A detailed description of the classification algorithms employed in this study would go beyond the scope of this article. We will therefore only briefly outline each method and the parameters used. For more details about the underlying mathematics, we refer to the citations given in each of the following subsections.

Support-vector machine (SVM)

SVMs provide a novel supervised machine-learning technique initially proposed by Vapnik.[23] Since they have shown good classification performance in various scenarios, there is a noticeably growing interest in SVMs. Some recent chemistry-related applications include the prediction of isoelectric points of amino acids,[20,24] aqueous solubility of organic molecules,[25] discrimination of drugs and nondrugs,[26] and compound selection for specific biological assays.[27]

Moreover, Burbidge et al. conducted a comparative study on an SAR data set in which SVM clearly outperformed neural networks of different architecture as well as a decision tree classifier.[28]

The basic concept behind SVMs is to first project the input data vectors, which are composed of the respective descriptors, to a high-dimensional feature space. This mapping is accomplished by applying a so-called "kernel function". The second step of the algorithm involves detection of a hyperplane that optimally separates the individual classes of the training set. Descriptor vectors of the test set are then mapped to the same feature space, and the hyperplane can be used to predict the class membership of these instances.

In this study, we employed LIBSVM, a freely available SVM code, written by Chang and Lin.[29] As recommended by the authors of the program, we applied the radial basis function (RBF) as the kernel. With this setting, two parameters have to be tuned: the penalty parameter, C, and the RBF parameter, χ (see ref. [29b] for a detailed discussion). We used tenfold cross validation to find the optimal values for C and χ for each of the 13 training sets. These parameters are reported in Table 2.

Artificial neural net (ANN)

In this study, a standard feed-forward neural network was applied by using the ANN implementation of the TSAR software package.[30] The network, which undergoes a supervised training by back-propagation of errors, is comprised of 120 input neurons, that is, the Ghose–Crippen descriptors, five hidden neurons, and one output neuron. All layers are completely connected. For each training set, an individual network is trained with default parameters. To avoid overtraining, 30% of the training set (default value of the ANN implementation in TSAR) is randomly chosen and excluded from the training.

k nearest neighbors with genetic algorithm-based variable selection (GA/*k*NN)

*k*NN classifiers provide another supervised-learning method for subdividing a set of data points each of which are characterized by a vector of *x* descriptor values into different classes.

Table 2. Classification results on the test data by using SVM.

Model	*tp*	*fn*	*fp*	*tn*	Accuracy	Precision	Recall	F1	C	χ
1	176	28	30	270	0.89	0.85	0.86	0.86	1.25	−1.25
2	168	36	30	270	0.87	0.85	0.82	0.84	0.75	−1.50
3	175	29	36	264	0.87	0.83	0.86	0.84	2.50	−1.75
4	173	31	28	272	0.88	**0.86**	0.85	0.85	1.75	−2.00
5	**181**	**23**	31	269	0.89	0.85	0.89	0.87	2.00	−0.50
6	173	31	39	261	0.86	0.82	0.85	0.83	0.25	−1.50
7	180	24	31	269	0.89	0.85	0.88	0.87	1.25	−0.50
8	170	34	40	260	0.85	0.81	0.83	0.82	1.75	−2.00
9	168	36	30	270	0.87	0.85	0.82	0.84	0.50	−1.00
10	176	28	**28**	**272**	0.89	**0.86**	0.86	0.86	0.75	-0.50
11	**181**	**23**	30	270	**0.90**	0.86	**0.89**	**0.87**	2.50	−1.25
12	165	39	30	270	0.86	0.85	0.81	0.83	0.50	−1.50
13	171	33	34	266	0.87	0.83	0.84	0.84	0.50	−1.50
average	173.6	30.4	32.1	267.9	0.88	0.84	0.85	0.85		
SD	5.2	5.2	3.9	3.9	0.01	0.02	0.03	0.02		
majority vote	174	30	29	271	0.88	**0.86**	0.85	0.86		

The *k*NN methods predict the classification of an unknown instance based on the majority vote of its *k* nearest neighbors in the given *x*-dimensional feature space. Usually, the Euclidian distances from the given probe to all data points in the knowledge base are calculated. The *k* closest points represent the voters. In order to avoid over-fitting, the dimensionality of the feature space is often reduced and only the most discriminatory features are combined to determine the outcome of the vote. Rather than just scaling all input features numerically to the same range of values, the relative weights of the individual descriptors can be determined by optimizing the accuracy of the *k*NN prediction. Apart from these weights, the value of *k*, which has to be an uneven number to avoid a tie, also exerts an influence on the achieved prediction accuracy. A very elegant way of tackling both of these optimization problems is to link the *k*NN classifier to a genetic algorithm (GA).[31] The GA varies the feature weights and the number of neighbors in an evolution-like procedure, until a near-optimal solution for the training set has been found. Here, we use an adaptation of the GAUCSD software package linked to a *k*NN classifier developed by Raymer.[32] The implemented fitness-function, which monitors the evolutionary progress of the solutions is given in Equation (5). It takes the overall prediction accuracy as well as the balance between classes into account. Moreover, it aims at reaching this goal with as few descriptors as possible by awarding the masking (omission) of features:

$$\begin{aligned}
&\text{Fitness (weight-set, } k \text{ value)}\\
&= 20 \times (\text{incorrect predictions/total predictions})\\
&\quad + 1.0 \times (\text{unmasked features/total features})\\
&\quad + 2.0 \times (\text{incorrect votes/total votes})\\
&\quad + 5.0 \times (\text{difference in error rate among classes}) \qquad (5)
\end{aligned}$$

The training of the GA/*k*NN requires a knowledge base from which the voters are recruited and which is also used in the subsequent testing phase, plus a set of probe data points the classification of which is progressively optimized. Accordingly, the training sets described above are further randomly subdivided into two equally large and balanced sets, each comprising 565 data points. By using the selection of features with their respective weights as well as the *k* value determined during the training phase, the data points of the test sets are subsequently classified according to the vote of the knowledge base, thus providing the unbiased prediction accuracies as reported in Table 3.

Recursive partitioning (RP)

We used the FIRM (formal inference-based recursive modeling) algorithm implemented in TSAR for recursive partitioning.[33] FIRM is a type of decision-tree analysis in which a large data set is progressively split into subgroups based on descriptor values (predictor variables). For each split, a *P* value that indicates the probability of a subgroup being homogeneous with respect to the class membership is calculated. Subsequently, the descriptor/subgroup combination that yields the lowest *P* value is selected to split the data. This procedure is repeated for each newly formed subgroup until no further split can be justified. As a result, a decision tree is formed with the final subgroups being the leaves of the tree. Typically only a subset of input descriptors is used for splitting, and descriptors contributing only little to the discrimination between classes are disregarded. The depth, that is, the number of splits down the tree in which they occur provides an indication of their importance. Thus, in contrast to SVM and ANN but similar to GA/*k*NN, recursive partitioning provides a measure for the relative discrimination potential of the input descriptors.

Prediction of the test set is straightforward. The splitting rules developed for the training set are applied to the descriptor vector of an unknown compound until it falls into a terminal leaf. The predicted output value is given by the average labels of the training set compounds in this leaf. As this study deals with a two-class problem, with active compounds being assigned a label of 1 and inactives being assigned a label of 0,

Table 3. *Classification results on the test data by using GA/kNN.*

Model	*tp*	*fn*	*fp*	*tn*	Accuracy	Precision	Recall	F1	*k*	No. of descriptors
1	170	34	43	257	0.85	0.80	0.83	0.81	5	41
2	168	36	51	249	0.83	0.77	0.82	0.79	3	39
3	164	40	37	263	0.85	0.82	0.80	0.81	5	33
4	169	35	41	259	0.85	0.81	0.83	0.82	7	36
5	164	40	42	258	0.84	0.80	0.80	0.80	5	38
6	176	28	53	247	0.84	0.77	0.86	0.81	3	44
7	169	35	42	258	0.85	0.80	0.83	0.81	5	38
8	**178**	**26**	61	239	0.83	0.75	**0.87**	0.80	1	39
9	173	31	52	248	0.84	0.77	0.85	0.81	1	44
10	172	32	52	248	0.83	0.77	0.84	0.80	7	36
11	163	41	40	260	0.84	0.80	0.80	0.80	5	34
12	167	37	43	257	0.84	0.80	0.82	0.81	7	40
13	174	30	48	252	0.85	0.78	0.85	0.82	5	46
average	169.8	34.2	46.5	253.5	0.84	0.79	0.83	0.81		
SD	4.7	4.7	6.9	6.9	0.01	0.02	0.02	0.01		
majority vote	172	32	**35**	**265**	**0.87**	**0.83**	0.84	**0.84**		

the possible output of the decision tree is a real number in the range [0,1]. For class prediction, we rounded the output value to the nearest integer.

Results and Discussion

The major goals of this study were first to test to what extent machine-learning methods are capable of learning and predicting the "kinase inhibitor-likeness" of compounds, and second to compare the performance of different classification methods within this scenario.

As outlined in the Computational Methods section, we have generated 13 training sets, each balanced with regard to the number of active kinase inhibitors and inactive compounds. Each set was used to derive a cross-validated model, the predictive power of which was then determined on an independent test set. Moreover, we employed an ensemble-based voting procedure in which the majority of the 13 models decide the class membership of the test compounds.

In the following paragraphs, we discuss the classification results obtained for each of the four learning methods and compare their performance characteristics.

Support-vector machine (SVM)

As can be deduced from Table 2, in general the SVM models yield very high and balanced scores in all success measures. In addition, the intermodel variations, that is, the standard deviations from the average, are quite low compared to the other learning methods (see below). This means that SVM seems to be only marginally sensitive to the particular composition of the training set. Ensemble-based majority voting gives an—albeit small—additional benefit over the average of the individual models. Due to the nature of the SVM algorithm, no information about the relative importance of individual descriptors can be obtained. Taken together, the SVM method provides robust and reliable models irrespective of the success measure and training set.

Artificial neural net (ANN)

Table 4 summarizes the performance of the ANN approach. Interestingly, although the scores of the individual models are significantly lower than those obtained with SVM, the results from the majority voting even slightly outperform SVM. This finding is somewhat hard to rationalize, given the fact that both SVM and ANN must be considered more or less as black-box approaches. It appears that the predictions yielded by the 13 SVM models are more consistent than those of the ANN models. This is also reflected in the higher standard-deviation values of the latter.[34] Accordingly, the majority-voting procedure benefits from this greater variety of "voters".

Table 4. *Classification results on the test data by using ANN.*

Model	*tp*	*fn*	*fp*	*tn*	Accuracy	Precision	Recall	F1
1	171	33	48	252	0.84	0.78	0.84	0.81
2	167	37	64	236	0.80	0.72	0.82	0.7
3	167	37	66	234	0.80	0.72	0.82	0.76
4	168	36	59	241	0.81	0.74	0.82	0.78
5	159	45	53	247	0.80	0.75	0.78	0.76
6	165	39	60	240	0.80	0.73	0.81	0.77
7	176	28	80	220	0.79	0.69	0.86	0.77
8	172	32	77	223	0.78	0.69	0.84	0.76
9	165	39	66	234	0.79	0.71	0.81	0.76
10	175	29	81	219	0.78	0.68	0.86	0.76
11	179	25	75	225	0.80	0.70	0.88	0.78
12	171	33	73	227	0.79	0.70	0.84	0.76
13	176	28	57	243	0.83	0.76	0.86	0.81
average	170.1	33.9	66.1	233.9	0.80	0.72	0.83	0.77
SD	5.6	5.6	10.5	10.5	0.02	0.03	0.03	0.02
majority vote	**180**	**24**	**35**	**265**	**0.88**	**0.84**	**0.88**	**0.86**

k nearest neighbors with genetic algorithm-based variable selection (GA/*k*NN)

Unlike the two previous methods, this machine-learning technique aims at a reduction of the given feature space, even at the cost of a small drop in prediction accuracy. In fact, any single model as depicted in Table 3 uses less than half of the descriptors available, with the number of descriptors included varying between 33 and 46. Although within this range no correlation between the number of descriptors and F1 can be found, the run that takes the largest number of descriptors into account is actually the one yielding the highest F1 value. Surprisingly enough, the GA/*k*NN combination still ranks second best, after SVM, when the average performances of the models are compared. Although the knowledge base is only half the size of the training sets used with the other methods, the generally low standard deviations substantiate the robustness of the prediction. However, when the majority votes are compared, GA/*k*NN falls behind both SVM and ANN by approximately 2%. Evidently, in this case, the ANN can gain predictive power on the *k*NN quite significantly. This is somewhat unexpected, since the majority vote procedure allows for the reincorporation of descriptors that had been omitted in the individual runs. Apparently, this still does not lead to an improvement in terms of F1.

Table 5. *Classification results on the test data by using RP.*

Model	*tp*	*fn*	*fp*	*tn*	Accuracy	Precision	Recall	F1
1	146	58	64	236	0.76	0.70	0.72	0.71
2	147	57	42	258	0.80	0.78	0.72	0.75
3	147	57	64	236	0.76	0.70	0.72	0.71
4	143	61	38	262	0.80	0.79	0.70	0.74
5	135	69	31	269	0.80	0.81	0.66	0.73
6	145	59	50	250	0.78	0.74	0.71	0.73
7	150	54	60	240	0.77	0.71	0.74	0.73
8	139	65	40	260	0.79	0.78	0.68	0.73
9	142	62	42	258	0.79	0.77	0.70	0.73
10	**152**	**52**	59	241	0.78	0.72	**0.75**	0.73
11	134	70	60	240	0.74	0.69	0.66	0.67
12	150	54	45	255	0.80	0.77	0.74	0.75
13	**152**	**52**	39	261	0.82	0.80	**0.75**	0.77
average	144.8	59.2	48.8	251.2	0.79	0.75	0.71	0.73
SD	6.0	6.0	11.3	11.3	0.02	0.04	0.03	0.02
majority vote	143	61	**27**	**273**	**0.83**	**0.84**	0.70	**0.77**

Taking a closer look at the descriptors selected in the individual runs and their relative weights, a remarkable variation in the descriptor patterns can be observed. Every single input descriptor has been used in at least one of the 13 models, five have only been used in one single model. Although a fraction of the initial features is sufficient to derive an F1 value well in the order of those obtained with the two methods incorporating all features (SVM: 0.85, ANN: 0.77↔GA/*k*NN: 0.81), there are various possible combinations of descriptors that lead to approximately the same prediction accuracy. At the same time, four descriptors can be identified that are used in every one of the 13 runs and thus appear to play a prominent role. All of them characterize polar atoms attached to or incorporated into planar/aromatic structures. This is well in accordance with the notion that kinase inhibitors in general depict an aromatic system replacing the adenine part of ATP in the binding pocket and forming at least one hydrogen bond to the hinge region of the kinase. Not unexpectedly, the inspection of features that are rarely selected or are assigned very low weights suggests that their selection results from the limited size of the data sets used for training. They should be considered artifacts rather than molecular features conveying kinase binding.

Recursive partitioning (RP)

This method, the results of which are summarized in Table 5, by far performs the poorest of all. This is not surprising given the fact that the predictions are based on the mean "activity" of the training compounds in the final leaves of the decision tree. Analysis of each of the 13 decision trees revealed that the average number of descriptors characterizing a final leaf, that is, the average "depth" of the tree, is only about five. This implies that predictions on average rely on only about 6% (5 out of 120) of the Ghose–Crippen descriptors. On the other hand, the strength of decision trees lies more in the information one gets about the relative importance of the input descriptors rather than on predictive power. Notably, the descriptors recurrently selected by the RP procedure are identical to the ones that the GA/*k*NN identifies as prominent features.

A summary of the performances of the four different classification methods is given in Figure 2. As discussed in the previous sections, the ensemble-based voting procedure in general outperforms averaging over the individual models. The relative gain of the voting procedure over averaging seems to correlate coarsely with the diversity of the individual training models.

Based on the F1 success measure,[35] the following ranking order can be deduced:

a) for average models:

SVM > GA/kNN > ANN > RP

b) for the majority vote of an ensemble of 13 models:

ANN > SVM > GA/*k*NN ≫ RP

The overlap of true predictions of the active compounds in the test set is depicted in Table 6.

Table 6. *Overlap [%] of correctly classified active compounds in the test set*

	SVM	ANN	GA/*k*NN
ANN	92.4		
GA/*k*NN	88.3	89.2	
RP	77.1	74.6	77.0

The overlap percentages are quite high and correlate well with the overall performance rates of the different machine learning methods (Tables 2–5). This implies that there is no

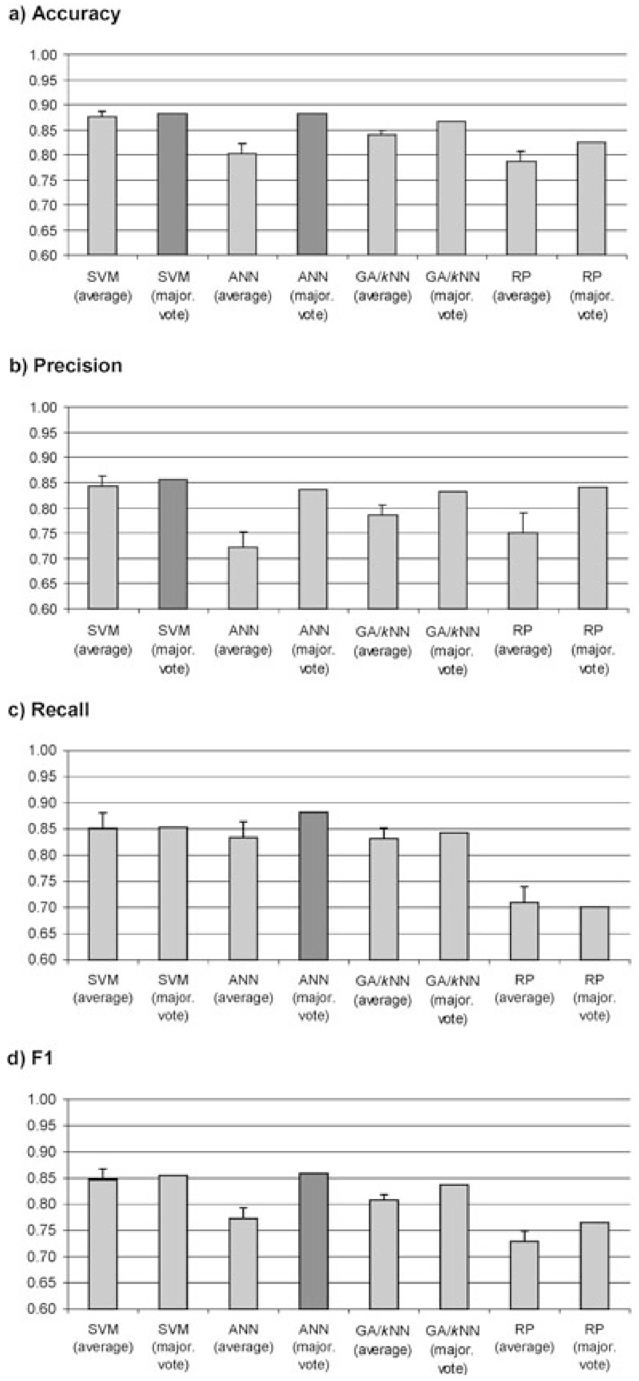


Figure 2. Summary of the results of the machine-learning methods employed on the independent test set. The best-performing model for each success measure is depicted in dark gray.

pronounced inter-relationship between a particular classifier and a certain class of inhibitors. Consequently, all the methods, in combination with Ghose–Crippen descriptors, appear to be suited to generalization, that is, to correctly predict the kinase likeness of different structural classes of inhibitors.

In addition to the validation by statistical measures, we also tested the ability of our models to perform scaffold hopping. The approaches complement each other. Therefore, we compiled a set of known protein kinase inhibitors. As described in the Computational Methods, none of these inhibitors showed significant 2D similarity to any compound in the training set. Thus, a correct classification cannot be attributed to the recognition of the main scaffold alone, but reveals the potential for detecting truly novel inhibitors.

As depicted in Table 1, the rank order of prediction results for this diverse data set resembles that of our test set of in-house compounds, with SVM performing best and RP performing worst—although, with only ten compounds in the data set, no statistical significance can be expected.

Remarkably, Gleevec, which is known to exhibit a distinct binding mode that only partially overlaps with that of most other kinase inhibitors,[36] is correctly predicted by all four classifiers. On the other hand, the p38 MAP kinase inhibitor BIRB796, which shows a binding mode very similar to Gleevec,[37] is misclassified by all methods. This example points to a clear limitation in employing 2D fragment descriptors for classification. As long as the majority of fragments in a test compound have not been part of active compounds in the training set—as seems to be the case for BIRB796—even the most sophisticated classifier must fail. In future work, it might be interesting to feed the machine-learning methods with descriptors derived from 3D docking modes or pharmacophore models.

Conclusion

All four machine learning techniques employed in this study—SVM, ANN, GA/*k*NN, and RP—proved capable of providing a reasonable discrimination between kinase inhibitors and non-inhibitors. Average F1 values above 0.8 could be obtained for both SVM-based models and GA/*k*NN-based models; this suggests that these methods are well applicable for compound selection in practice.

Using the majority vote of the 13 models derived improved the prediction quality for all four methods, but most pronouncedly for the neural networks. In fact, the ANN majority vote outperformed all other predictions in terms of recall (0.88) and F1 (0.86). The majority vote of the SVM models yielded the highest precision (0.86) and as good an accuracy (0.88) as the ANN ensemble. Although the different data sets and molecular descriptors used limit the comparability of results, these figures are in the same range as the prediction accuracies of 79% that Manallack et al.[15] obtained for the discrimination of kinase inhibitors and nonkinase inhibitors.

If information on the underlying discriminatory features is not desired or not required, and the prediction machine is only to be used as a black box, either ANN or SVM can be used to derive highly predictive models. The difference in performance between the two methods is minor if an ensemble vote of individual models is used. While the improvement in performance is relatively small for SVM, it is large for ANN. Thus, a single SVM-based model may well be used. When choosing ANN, however, the derivation of an ensemble of models for majority voting is advisable.

If a good prediction along with some information on the discriminatory features is desired, the GA/*k*NN combination appears the method of choice. This approach also offers advantages if the size of the data set and the descriptor space suggest a reduction of feature space to avoid overtraining.

Due to the enormous reduction of feature space in RP, it is not surprising that RP cannot quite compete with the other three methods in the scenario given here. While this method should not preferentially be used to screen large compound sets for feasible kinase inhibitors, it provides a very valuable and extremely fast approach toward the identification of discriminatory features in a given data set. Moreover, it gives a rough first estimate as to what order of prediction accuracy might be obtained with more sophisticated methods. In addition, biases in a given data set that enable high prediction accuracy but only seemingly resolve the given classification problem can quickly be identified with this method.

Our results are in close agreement with other comparative studies (Byvatov et al.,[26] Burbidge et al.[28]), featuring SVM as a fast and reliable machine-learning method, which is at least comparable in performance to neural networks and other classification approaches.

Concerning the choice of molecular descriptors, we believe that the balance between a general and a detailed level of abstraction provided by the set of Ghose–Crippen fragments makes them well-suited for the classification of small molecules with respect to their potential for inhibiting particular target family classes.

Acknowledgements

We are grateful to Dr. Chih-Chen Lin (National Taiwan University) for his generous help in getting us started on LIBSVM and to Dr. Leslie Kuhn (Michigan State University) for the fruitful cooperation in the field of GA/kNN algorithms. Moreover, we would like to thank Dr. Sabine Schlyer (Berlex Biosciences Inc.) and Sonja Zechner (Schering AG) for carefully reading the manuscript.

Keywords: computer chemistry · drug design · inhibitors · kinases · machine learning

[1] G. Wess, M. Urmann, B. Sickenberger, *Angew. Chem.* **2001**, *113*, 3443–3453; *Angew. Chem. Int. Ed.* **2001**, *40*, 3341–3350.

[2] K.-H. Bleicher, H.-J. Böhm, K. Müller, A. I. Alanine, *Nat. Rev. Drug Discovery* **2003**, *2*, 369–378.

[3] P. M. Dean, E. D. Zanders, D. S. Bailey, *Trends Biotechnol.* **2001**, *19*, 288–292.

[4] G. Müller, *Drug Discovery Today* **2003**, *8*, 681–691.

[5] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, S. Sudarsanam, *Science* **2002**, *298*, 1912–1934.

[6] P. Cohen, *Nat. Rev. Drug Discovery* **2002**, *1*, 309–315.

[7] J. Drevs, M. Medinger, C. Schmidt-Gersbach, R. Weber, C. Unger, *Curr. Drug Targets* **2003**, *4*, 113–121.

[8] S. Cheek, H. Zhang, N. V. Grishin, *J. Mol. Biol.* **2002**, *320*, 855–881.

[9] M. Cherry, D. H. Williams, *Curr. Med. Chem.* **2004**, *11*, 663–673.

[10] T. Naumann, H. Matter, *J. Med. Chem.* **2002**, *45*, 2366–2378.

[11] M. Huse, J. Kuriyan, *Cell* **2002**, *109*, 275–282.

[12] R. A. Engh, D. Bossemeyer, *Pharmacol. Ther.* **2002**, *93*, 99–111.

[13] J. Sadowski, H. Kubinyi, *J. Med. Chem.* **1998**, *41*, 3325–3329.

[14] A. Ajay, W. P. Walters, M. A. Murcko, *J. Med. Chem.* **1998**, *41*, 3314–3324.

[15] D. T. Manallack, W. R. Pitt, E. Gancia, J. G. Montana, D. J. Livingstone, M. G. Ford, D. C. Whitley, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1256–1262.

[16] ClassPharmer, Version 3.2, Bioreason Inc., Santa Fe, NM, USA.

[17] M. E. M. Noble, J. A. Endicott, L. N. Johnson, *Science* **2004**, *303*, 1800–1805.

[18] I. R. Hardcastle, B. T. Golding, R. J. Griffin, *Ann. Fac. Agrar. Ann. Rev. Pharmacol. Toxicol.* **2002**, *42*, 325–348.

[19] J. L. Durant, B. A. Leland, D. R. Henry, J. G. Nourse, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1273–1280.

[20] a) A. K. Ghose, G. M. Crippen, *J. Comput. Chem.* **1986**, *7*, 565–577; b) A. K. Ghose, G. M. Crippen, *J. Chem. Inf. Comput. Sci.* **1987**, 27, 21–35; c) A. K. Ghose, A. Pritchett, G. M. Crippen, *J. Comput. Chem.* **1988**, *9*, 80–90.

[21] R. Yan, Y. Lui, R. Jin, A. Hauptmann, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong (China), April 6–10, **2003** (http://www-2.cs.cmu.edu/~yanrong/Publication/ICASSP03.pdf).

[22] a) C. J. van Rijsbergen, *Information Retrieval*, 2nd ed, Butterworths, London, **1979**; b) D. D. Lewis, W. A. Gale, In: *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* **1994**, 3–12.

[23] a) C. Cortes, V. Vapnik, *Machine Learning* **1995**, *20*, 273–297; b) V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, **1995**.

[24] H. X. Liu, R. S. Zhang, X. J. Yao, M. C. Liu, Z. D. Hu, B. T. Fan, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 161–167.

[25] P. Lind, T. Maltseva, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1855–1859.

[26] E. Byvatov, U. Fechner, J. Sadowski, G. Schneider, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1882–1889.

[27] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, C. Lemmen, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667–673.

[28] R. Burbidge, M. Trotter, B. Buxton, S. Holden, *Comput. Chem.* **2001**, *26*, 5–14.

[29] a) C.-C. Chang, C.-J. Lin, LIBSVM: *A library for support vector machines*, **2001**. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm; b) http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[30] TSAR, Version 3.3, Accelrys Inc., San Diego, CA, USA.

[31] R. D. Judson, *Rev. Comput. Chem.* **1997**, *10*, 1–73.

[32] a) M. L. Raymer, Dissertation, Michigan State University, 2000; b) M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, L. A. Kuhn, *J. Mol. Biol.* **1997**, *265*, 445–464.

[33] D. M. Hawkins, S. S. Young, A. Rusinko, *Quant. Struct.-Act. Relat.* **1997**, *16*, 296–302.

[34] It should be noted that in contrast to SVM training, we have not tuned any parameters in ANN training but rather used default values of the ANN implementation in TSAR. Thus, we cannot rule out the possibility that such a tuning could have resulted in even better and more consistent models.

[35] All classification methods employed in this study train on optimal accuracy rather than on F1. As we believe that the latter measure is more useful in many real-life applications, we suggest to authors of machine learning programs to consider a more flexible metric selection for future releases. In fact, training on F1 rather than accuracy might have lead to a further improvement of our classification results.

[36] B. Nagar, W. G. Bornmann, P. Pellicena, T. Schindler, D. R. Veach, W. T. Miller, B. Clarkson, J. Kuriyan, *Cancer Res.* **2002**, *62*, 4236–4243.

[37] C. Pargellis, L. Tong, L. Churchill, P. F. Cirillo, T. Gilmore, A. G. Graham, P. M. Grob, E. R. Hickey, N. Moss, S. Pav, J. Regan, *Nat. Struct. Biol.* **2002**, *9*, 268–272.

Received: April 19, 2004
Published online on February 7, 2005