Chemical Data Mining of the NCI Human Tumor Cell Line Database

Huijun Wang,[†] Jonathan Klinginsmith,[†] Xiao Dong,[†] Adam C. Lee,[‡] Rajarshi Guha,[†] Yuqing Wu,[†]
Gordon M. Crippen,[‡] and David J. Wild*[,†]

Indiana University School of Informatics and Chemical Informatics and Cyberinfrastructure Collaboratory,
901 East Tenth Street, Bloomington, Indiana 47408, and College of Pharmacy, University of Michigan,
428 Church Street, Ann Arbor, Michigan 48109-1065

Received April 20, 2007

The NCI Developmental Therapeutics Program Human Tumor cell line data set is a publicly available database that contains cellular assay screening data for over 40 000 compounds tested in 60 human tumor cell lines. The database also contains microarray assay gene expression data for the cell lines, and so it provides an excellent information resource particularly for testing data mining methods that bridge chemical, biological, and genomic information. In this paper we describe a formal knowledge discovery approach to characterizing and data mining this set and report the results of some of our initial experiments in mining the set from a chemoinformatics perspective.

INTRODUCTION

Since 1990, National Cancer Institute Developmental Therapeutics Program (DTP) has been screening compounds against a panel of 60 human tumor cell line assays. The results are available on the DTP Web site.[1] Approximately 10 000 compounds are screened each year, and at the time of writing, results were available for 44 653 compounds including growth inhibition (GI_{50}), lethal dose (LD_{50}), and total growth inhibition (TGI). The untreated cell lines have also been run through microarray assays, yielding gene expression information.

The tumor cell line data set is interesting in several ways relating to current research in finding biomarkers that cross different kinds of data and in using chemical, biological, and genomic information together. First, it provides a well curated set of tumor-related cellular assay screening results for a large number of compounds (the 60 cell lines include melanomas, leukemias, and cancers of the breast, prostate, lung, colon, ovary, kidney, and central nervous system[2]), which can be considered as a surrogate for high-throughput screening data. Second, the gene expression profiles of untreated cell lines allow some level of integration of genomic information with chemical and biological information. Third, the program is ongoing and so the tumor cell line data set is continually growing, but the cell lines themselves are stable (both in terms of number and comparability of results). Fourth, and most importantly, the data are made freely available through the DTP Web site and are thus available for research and publication.

A substantial amount of research on the tumor cell line data set has been carried out locally at the NCI laboratories including development of the COMPARE algorithm[3,4] which measures similarity between vectors of screening results of compounds using a Pearson correlation coefficient. A searching program based on COMPARE is available online.[5] Zaharevitz et al.[4] cite several examples of the successful application of these approaches in drug discovery projects. The original authors of COMPARE also introduced the use of the *mean graph*[3] that gives a visual bar graph representation of the difference between the screening result for a particular compound and the mean for all compounds, across the 60 cell lines. This representation has been widely used alongside COMPARE.

Other research has used neural networks[6] to classify compounds in the set. In their 2000 paper, Scherf et al.[7] examine correlations between compounds' high-throughput screening results (the activity pattern set) and mRNA expression levels. Recently, Rabow et al.[8] performed a clustering of the tumor cell line data set based on the activity profiles, using a self-organizing map (SOM). Other work at the NCI focused on ellipticine analogs and the potential relationship between the mechanism of action and the 60 cell line activity profiles. The compounds were grouped using hierarchical clustering, and a significant difference in activity profiles was found for groups with different mechanisms of action[9] which led to a follow-up QSAR study.[10]

Researchers at Leadscope Inc. have applied their Leadscope software[11] to relate the information in the tumor cell line data set to structural feature analysis of the DTP compounds, including analysis similar to that done by Scherf[12] and correlations of chemical structural features of cytotoxic agents with gene expression data.[13] Blower et al.[14] also applied a three-stage pipeline to the data set, including filtering for druglikeness, structure alerts, promiscuity and diversity; structural feature based classification using a variant of Recursive Partitioning (requiring separation of actives and inactives) and organization based on hierarchical clustering; and SAR analysis through R-group assembly, macrostructure assembly, and predictive models. The researchers found a close match between classifications and clusters found by Leadscope and manual classifications previously identified at NIH.

* Corresponding author phone: (812)856-1848; e-mail: djwild@indiana.edu.

[†] Indiana University School of Informatics and Chemical Informatics and Cyberinfrastructure Collaboratory.

[‡] University of Michigan.

10.1021/ci700141x CCC: $37.00
Published on Web 10/04/2007

Recently, Richter et al.[15] have evaluated an activity prediction model based on both structural information and genomic information, and at Bristol-Myers Squibb, a version of recursive partitioning derivative was applied.[16] Fang et al.[17] developed a set of Internet-based tools that permit correlations to be found between the activity profiles, gene expression profiles, and compounds using COMPARE as well as Spearman & Kendall correlation coefficients and a p-test to indicate significance of correlation results.

In this work, we have focused on characterizing the compounds present in the data set and applying a variety of methods to discover relationships between the compounds and the biological activity values. We have tried to take a more formal approach to data mining, such as has been applied in other domains where large volumes of information need to be searched for important associations. *Data Mining*, and more generally *Knowledge Discovery in Databases (KDD)*, is an area of computer science that has attracted a significant amount of research, industry, and media attention in the past decade, as the amount and complexity of information in databases has increased. Many KDD techniques, such as cluster analysis and decision trees, are already well established in chemical and bioinformatics, while others, such as data cleaning and pattern verification and discovery, are less widely applied.

PRINCIPLES AND PRACTICES OF KNOWLEDGE DISCOVERY IN DATABASES

KDD is usually defined as the process of identifying *valid*, *novel*, *potentially useful*, and ultimately *understandable* patterns from large collections of data. At an abstract level, it is concerned with the development of methods and techniques for making sense of data. Since its debut in 1989, KDD has become the most rapidly growing field in the database community and was soon adopted in other business and scientific areas, such as marketing, fraud detection, and bioinformatics. In practice, this field covers techniques often applied in chemoinformatics including cluster analysis, machine learning, and visualization techniques. Several KDD models have been proposed in the past decade. For the discussion in this paper, we adopt the 7-step KDD process presented in the most popular data mining textbook by Han and Kamber:[18] data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge presentation.

Knowledge Discovery goals are defined by the intended use of the system. Goals may be *verification goals*, in which the system is limited to verifying users' hypotheses, or *discovery goals*, in which the system is required to autonomously find entirely new patterns. Discovery goals may be *descriptive* (requiring characterization of general properties of the data in the database) and *predictive* (requiring predictions to be made using the data in the database).

Discovery goals are generally achieved through *data mining*. Data mining involves fitting models to, or determining patterns from, observed data. Model fitting may be stochastic or deterministic, although stochastic approaches are the most frequently used.

The first task of data mining is concept description. A concept is a labeling of a collection of data, such as labeling a set of "graduate students", "best-seller books", etc. The goal of concept description is to summarize the data of the class under study in general terms (*data characterization*) and to provide a description comparing two or more collections of data (*data discrimination*). Several methods have been proposed for efficient data summarization and discrimination. For example, a data cube[19] can be used for user controlled data summarization among concept hierarchies; analytical characterization can be used for unsupervised data generalization and characterization. After concept description, classification may be applied. The purpose of data classification is to find a set of models that describes and distinguishes data classes or concepts. Usually, finding such models is not the ultimate goal but rather the first step of using such models to predict the class of objects whose class is unknown or to predict future data trends. Decision trees are one of the most popular methods for data classification and predication.

In addition to classification, unsupervised clustering may be applied. The goal of cluster analysis is to examine data objects without consulting known class labels and is generally used as a way of organizing the database. In cluster analysis, objects are grouped based on *maximizing the intraclass similarity* and *minimizing the interclass similarity*. An excellent overview of clustering in chemoinformatics is given by Downs and Barnard.[20] Popular clustering algorithms used in data mining include partitioning methods such as k-means,[21] k-mediods,[22] and CLARANS[23] algorithm; hierarchical methods such as agglomerative and divisive algorithms, BIRCH[24] algorithm, CURE[25] algorithm, and Chameleon[26] algorithm; density-based methods such as DBSCAN,[27] OPTICS,[28] and DENCLUE;[29] grid-based methods such as STING,[30] WaveCluster,[31] and CLIQUE;[32] and model-based methods such as classification trees and neural networks. It is interesting to note that there is only limited overlap between the methods popularly applied in chemoinformatics and those applied in the data mining community as a whole.

Finally, association analysis may be applied. The goal of association analysis is the discovery of association rules showing attribute value conditions that occur together frequently in a given set of data. The Apriori[33] algorithm family has variants that are suitable for various data types and database models. Combining the association analysis and concept hierarchies, one may generalize the association rules with ISA relationship or various aggregations on different granularities.

Raw data are often not suitable for data mining, due to noise, missing or inconsistent data points, or lack of normalization across data sources. Preprocessing must therefore be applied. The purpose of *data cleaning* is to fill in incomplete data, smooth out noise, and correct inconsistencies. Data may be incomplete when attributes of interest are missing. Approaches for filling missing values include ignoring entries with missing values, filling missing values manually, using a global constant, using the attribute mean to fill in missing values, using the attribute mean for all samples belonging to the same class as the given entry, using the most probable value, and so on. *Noisy data* usually refers to data that contain errors or outlier values that deviate from the expected values. Approaches for noise elimination include the following: binning (smoothing a sorted data value by consulting its neighborhood), clustering (clustering data to detect and eliminate outliers), hybrid methods combining computer and human inspection, and regression (fitting the data to a function). *Inconsistent data* may be the result of errors that happen during data entry or due to the heterogeneous nature of data. The first usually needs to be handled manually. The inconsistency and data redundancy caused by heterogeneous data resources are usually handled in the data integration process.

Data integration and transformation are needed when data from heterogeneous resources are merged and transformed into forms appropriate for mining. In the data integration process, ontology is usually used for schema integration. Additional attention is needed to detect and resolve data value conflicts, such as attributes representing the same concept but using different units. Data transformation techniques include smoothing — removing the noise from data, aggregation, generalization — low level data are replaced by high level concepts, normalization — attribute data are scaled to fall within a small specific range and attribute construction — construct a new attribute to help mining.

Besides precision, performance is another important issue in data mining. The purpose of data selection is to obtain a data representation that is much smaller, yet closely maintains the integrity of the original data. Data reduction is the most common practice used in data selection. Many strategies have been proposed for data reduction: (1) Data cube aggregation,[19] where aggregation operations are applied to the data in the construction of a data cube. (2) Dimension reduction, where irrelevant, weakly relevant, and redundant attributes or dimensions are removed. The most popular dimension reduction algorithms are stepwise forward selection, stepwise backward elimination, hybrid (combination of forward selection and backward selection), and decision tree induction. (3) Data compression, where encoding is used to reduce the data size. Techniques include wavelet transformation, principal components analysis, etc. (4) Numerosity reduction, where the data are replaced or estimated by alternative, smaller data representations such as parametric models, by regression and log-linear models, histograms, clustering, or sampling. (5) Discretization and concept hierarchy generation where raw data values for attributes are replaced by ranges or higher conceptual levels.

A data mining system can generate thousands or even millions of clusters, classes, patterns, and rules. Not all of them are interesting to all users. The measurement of the “interestingness” of a pattern is subjective. Typically, a pattern is considered interesting if it is novel, valid with some degree of certainty, potentially useful, and easy to understand. It is unrealistic to expect a data mining system to generate all interesting patterns or only interesting patterns. This makes the measuring of pattern interestingness an essential component in KDD. A desirable feature of any data mining system is the development of a proper measurement model for a given field or user group and the use of it not only after all patterns are detected but also in the process of data mining as a guide for pruning uninteresting patterns and to speed up the mining process.

The data mining results, whether they are clusters or association rules, need to be presented to users (who usually are in the area of applications and are not database or data mining experts) before they can be deployed. Visualization and knowledge representation techniques are required to present the mining result to users, to improve the understandability. This is especially important for supervised mining tasks, where the user’s involvement is required in the mining process.

DATA CLEANING, INTEGRATION, SELECTION, AND TRANSFORMATION

At the time of writing the tumor cell line data set contained 257 547 compounds in total. Among those compounds, 44 653 compounds have cell line screening data (GI_{50}, LC_{50}, TGI data), and the total number of cell lines is 159, although only 60 of those cell lines have gene expression data. The gene expression data consist of 961 gene expression values for each cell line.[23] For the experiments reported here, we implemented a local version of the database containing the 44 653 compounds, screening results and gene expression values using PostgreSQL along with the gNova CHORD extension to allow chemical searching and generation of fingerprint bits.[34] 166-Bit structural key fingerprints were produced with gNova, based on a SMARTS-based interpretation of the public MACCS key set available from MDL.[35]

Characterization of the Chemical Compounds. There are several well-established methods of characterizing compounds by chemical properties or structural features. We applied two methods to characterize the compounds: first, calculation and profiling of predicted property values compared to two other well-established data sets, and second, a 2D fingerprint based structural feature comparison with compounds in one of the data sets.

In our first experiment, we chose three compound data sets for comparison to the tumor cell line set. The first is the FDA’s Maximum Recommended Therapeutic Dose (MRTD) set containing 1220 current prescription drugs available in SMILES format from the FDA Web site.[36] We chose this set as a representative of current marketed drugs. The second two sets were randomly selected 40 000 compound subsets of PubChem, a freely available chemical database,[37] used as representatives of a diverse set of chemical structures. We calculated properties (Molecular Weight, XLogP, Polar Surface Area, and Numbers of Hydrogen Bond Donors and Acceptors) for all of the structures in the data sets using OpenEye FILTER[38] and then generated property distribution plots for each of the properties for each of the data sets. These profiles can be seen in Figure 1. The most striking result is that the profiles for the tumor cell line set are very similar to those for the MRTD set, indicating that the compounds in the tumor cell line set are very “druglike”. The noticeably different (but consistent) profiles for the two PubChem subsets indicate that the compounds in PubChem are more diverse.

In our second experiment we compared the similarity of the drug compounds in the MRTD with the most similar compounds in the tumor cell line set: the distribution of the Tanimoto similarity values of the 166-bit fingerprints is shown in Figure 2. Overall 29% of the compounds in the MRTD set have a counterpart in the tumor cell line set with similarity greater than 0.8.

Characterization of the Cell Line Screening Growth Inhibition Values. We then went on to examine the distribution of the $-\log GI_{50}$ data points (henceforth referred to as growth inhibition values) across cell lines and compounds. First, it is important to note that there is missing

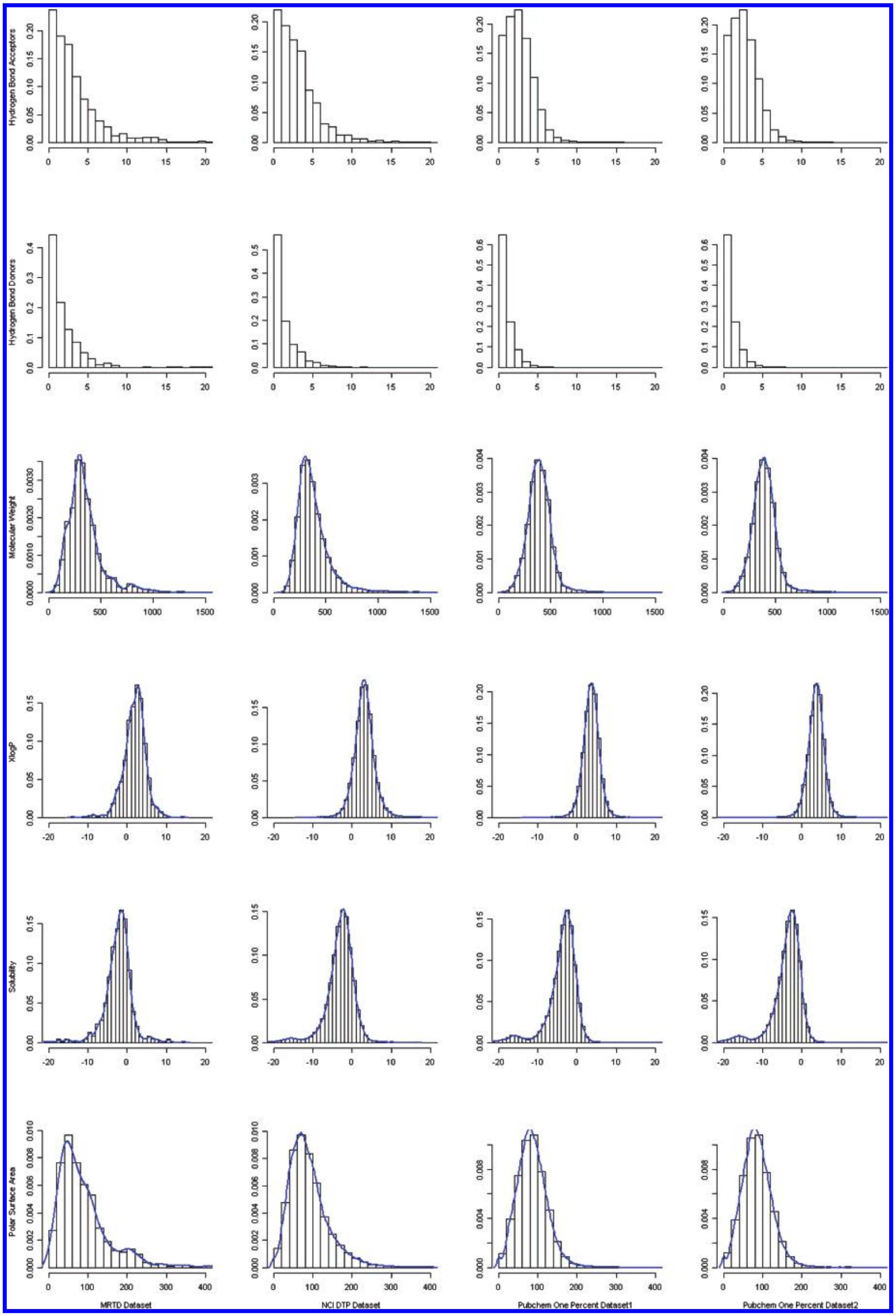

Figure 1. Comparative distribution of various properties for the compounds in the MRTD set (first column), tumor cell line set (second column), and two Pubchem subsets (third and fourth columns).

data: overall 12.1% of the cell line screen data points are missing. Figure 3 shows the percentage of compounds with missing data for each cell line. Only 2696 compounds (6%) have the growth inhibition values for all the 60 cell lines.

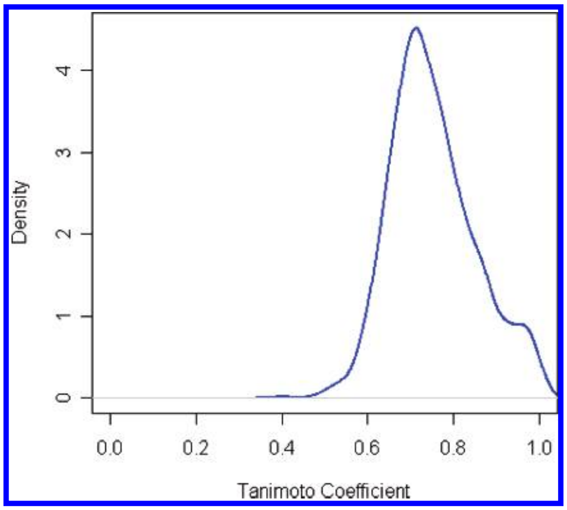

Figure 2. Distribution of Tanimoto similarity values (*x*-axis) between compounds in the MRTD set and the most similar compound for each in the tumor cell line set.

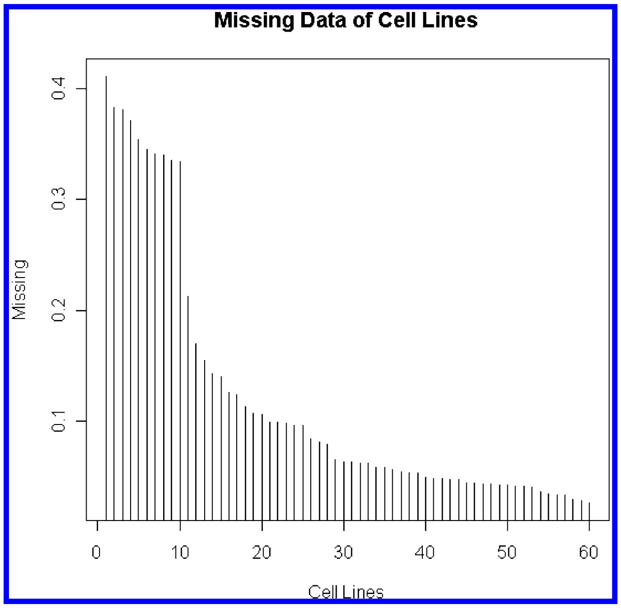

Figure 3. Fraction of the compounds with missing data for each of the 60 cell lines.

Growth inhibition values at or near 4.0 indicate inactivity of compounds (i.e., doses of less than 10^{-4} molar did not inhibit growth). Overall 44.9% of growth inhibition values are equal to 4.0 (see Figure 4 for the distribution across cell lines). When these compounds are removed from the set, a normal distribution can be seen with a peak of values less than 5.0, indicating inactive or extremely weakly active compounds. Based on this data distribution, we decided for our experiments to set the cutoff for determining whether a compound was active or inactive at 5.0: we consider the data which are less than 5 as inactive (set as 0) and the data which are greater or equal to 5 as active (set as 1). Overall, 19.6% compounds are considered active using this cutoff. The percentage of compounds considered "active" using this cutoff for each of the 60 cell lines is shown in Figure 5.

Characterization of the Gene Expression Results. Although this paper does not directly address data mining of the gene expression results, we carried out some initial experiments to characterize the data, for completeness and as a basis for future data mining experiments. The distributions of the microarray gene expression data are shown in Figure 6. The values less than zero represent underexpression from the norm and the values above zero represent overexpression. As shown, the overall distribution and the distribution for individual cell lines are very similar. Based on these distributions, for our work we decided to consider values less than or equal to −1.0 and greater than or equal to 1.0 to indicate under or overexpression, respectively.

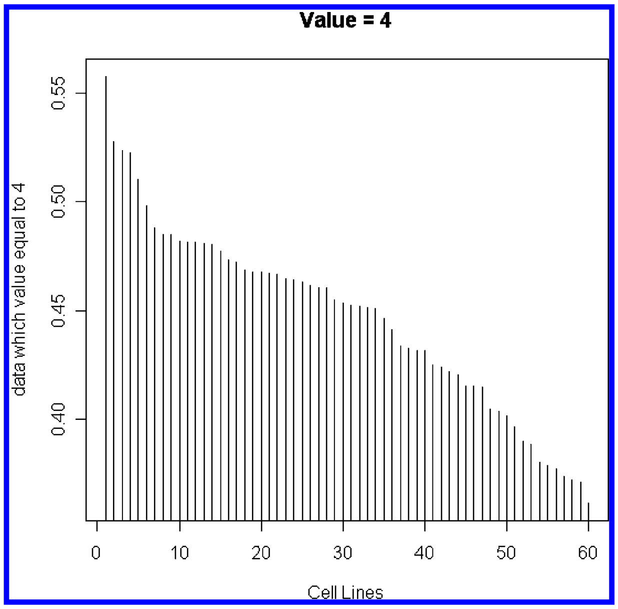

Figure 4. Fraction of compounds with growth inhibition values of 4.0 for each of the 60 cell lines.

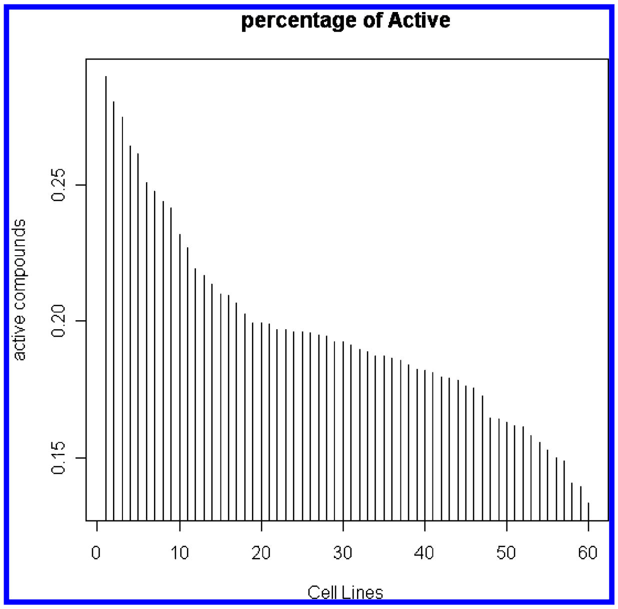

Figure 5. Fraction of compounds showing activity in each of the 60 cell lines.

Predicting Missing Activity Values. In order to test whether it might be possible to estimate the missing data points using computational prediction, we applied a machine learning tool, WEKA,[39] on the 2696 compounds which have values for all 60 cell lines. We did two prediction experiments using various methods: first using only 166 known attributes to predict one attribute (the 166 fingerprint is known and the cell line information is unknown); second a leave-one-out approach, using 255 known attributes to predict one attribute (the 166 fingerprint and 59 cell line growth

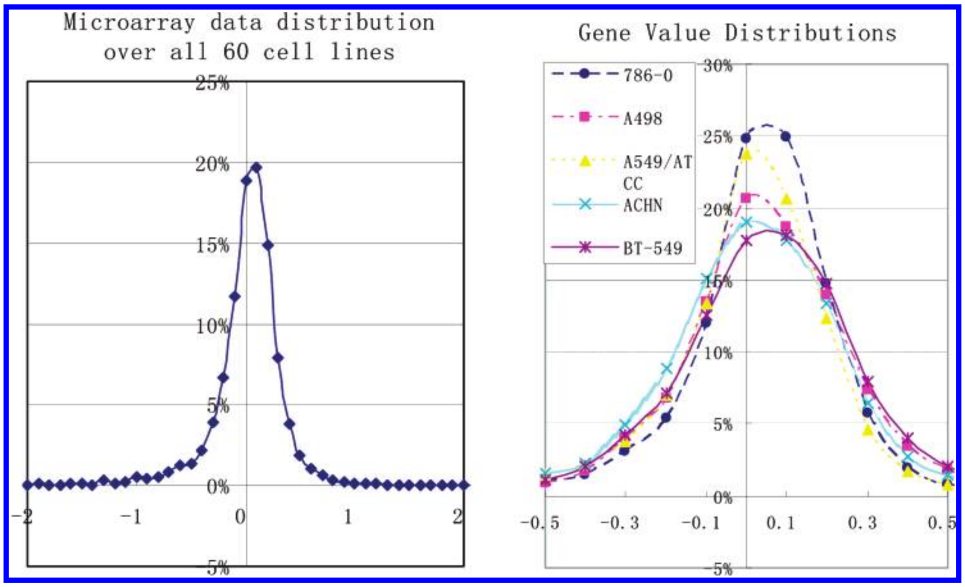

Figure 6. Distribution of the microarray gene expression data across all the 60 cell lines (left) and for five randomly selected cell lines (right).

Table 1. Accuracy of the Prediction Using Only Fingerprint Information

methods	TP_Rate	FP_Rate	Precision	Class
ADTree	0.087	0.047	0.434	0
	0.953	0.913	0.713	1
REPTree	0.192	0.098	0.451	0
	0.902	0.808	0.727	1
Ridor	0.029	0.008	0.59	0
	0.992	0.971	0.709	1
AODE	0.389	0.227	0.418	0
	0.773	0.611	0.751	1
BayesNet	0.436	0.303	0.376	0
	0.697	0.564	0.747	1
VFI	0.545	0.413	0.356	0
	0.587	0.455	0.755	1

Table 2. Accuracy of Prediction Using Fingerprint and Cell Line Information

methods	TP_Rate	FP_Rate	Precision	Class
ADTree	0.813	0.092	0.787	0
	0.908	0.187	0.92	1
REPTree	0.781	0.087	0.789	0
	0.913	0.219	0.909	1
Ridor	0.785	0.091	0.784	0
	0.909	0.215	0.91	1
AODE	0.815	0.102	0.771	0
	0.898	0.185	0.921	1
BayesNet	0.82	0.109	0.759	0
	0.891	0.18	0.922	1
VFI	0.83	0.118	0.746	0
	0.882	0.17	0.925	1

inhibition values as known attributes, one cell line growth inhibition value as unknown). Tables 1 and 2 show the accuracy of the prediction using various methods (ADTree and REPTree, two decision tree methods; RIDOR, a rule-based method; AODE and BayesNet, two Bayesian methods; and VFI, a voting feature interval classifier). The columns show the true and false positive rates, precision, and activity class for each of the methods. Clearly the accuracy is poor when only fingerprint bits are used, but is much improved when other cell line data are included. We may therefore assume that activity in one cell line is related to activity in others. While we would have liked to use this method to predict missing values, we are not confident that the set is complete enough to warrant it: 90% of the compounds miss some cell line data and only 10% of compounds are missing only one cell line data.

DATA MINING

Having obtained some broad characterizations of the compounds and cell line screening results in the set, we performed several experiments to find relationships between 2D chemical structure and activities across the 60 cell lines. Our intention in these experiments was to use both statistical and predictive modeling methods to look for associations and relationships between chemical structure features (as encoded by the 166-bit fingerprints) and the actual activities of the compounds in the 60 cell lines. Specifically, we applied a standard statistical ratio technique across all the cell lines, a random forest predictive modeling technique (as might be used in QSAR studies) to each cell line individually, and a novel rule-based SMARTS matching procedure that effectively generates "on-the-fly" structural descriptors related to activities.

Relating Dictionary-Based Structural Keys to Cellular Screening Activities. The activity classifications (active, inactive) and the structural key fingerprint bits described previously were used to determine which structural features were either more prevalent or scarce in active compounds compared with inactives. Two ratios, the active-structural ratio and overall-structural ratio, were created. The active-structural ratio $R_{a,j}$ for a structural feature j is defined as

$$R_{a,j} = \frac{T_{a,j}}{|C_a|}$$

where $T_{a,j}$ is the total number of compounds with the feature j, and C_a is the set of active compounds. The overall-structure ratio R_j is defined as

$$R_j = \frac{T_j}{|C|}$$

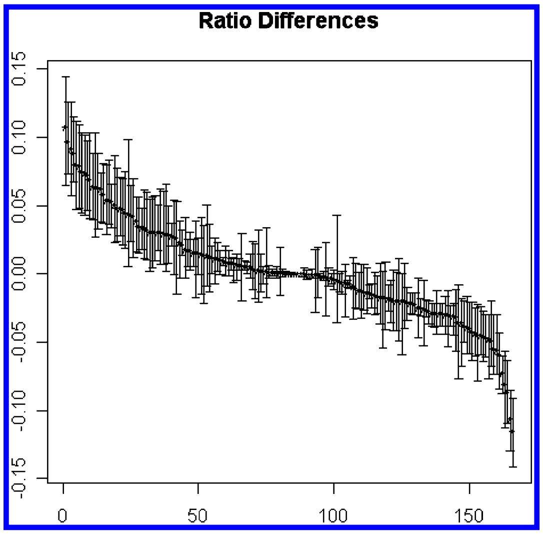

Figure 7. Difference in active structural ratio and overall structural ratio. Each of the 166 structural attributes is represented across the *x*-axis with the 60 cell lines displayed in various points. The central line shows the mean difference in active structural ratio and overall structural ratio.

where T_j is the total number of compounds with a structural feature *j*, and *C* is the complete set of compounds. We may then calculate the difference between these values, which provides a statistical value for how much more prevalent or absent a feature *j* is in the active compounds compared with the feature in all compounds:

$$\mathrm{diff}_j = R_j - R_{a,j}$$

Figure 7 plots the difference between the active ratio and the overall ratio for each of the 166 keys. A positive value indicates the greater percentage of this feature appearing in the active cells. Alternatively, a negative value indicates the lack of the feature in the active compounds compared with all compounds. Each feature was evaluated across all 60 cell lines, and thus each bar on the *x*-axis of the chart (each structural attribute) is based on 60 *y*-values. The effects of the substructure on the compounds' activities are very consistent as shown in the figure. Nearly all 60 cell lines follow the same track. Thus, we can use the average difference of the active ratio and the overall ratio to find the most important substructures in determining the "global" activity and inactivity. We may consider the features associated with global activity to be indicative of promiscuity (i.e., the tendency to bind to anything) and those associated with inactivity to be ones that tend to stop binding to tumor growth related proteins in a variety of situations. We found that the bits 105, 127, 145, 152, and 99 are the most important bits for activity and the bits 117, 110, 92, 77, and 95 are the most significant bits for inactivity. The Daylight SMARTS strings[40] and reasonable interpretations of those significant bits are shown in Table 3. In interpreting these results, it should be noted that approximately 5% of the structural keys differ only in the number of features present in the molecule, and some that almost never occur in biological molecules.

We may deduce from this that compounds with multiple ring systems, particularly involving oxygens and methyl groups, tend to be associated with activity, and close non-amide formations of nitrogens and oxygens as well as sulfur-containing compounds tend to not be active. This is borne out by looking at compounds which are active or inactive in all cell lines: a few examples are given in Figure 8.

Predictive Models of Activity. As shown in the last section, some structure features are highly correlated with activity or inactivity across the cell lines. We next performed experiments to see if it would be possible to build a predictive machine-learning model that can predict individual activity in each of the 60 cell lines. Our previous study with WEKA shows the AD-Tree and Ridor methods work best of the models available in that package. As an example, we initially applied those two methods on various feature subsets using cell line 60 (UO-31). The features are selected based on the rank of active and inactive features across all 60 cell lines and the rank of activity and inactivity features on cell line 60. For example, 20 features contain the top 10 active features and top 10 inactive features.

Table 3. SMARTS and Interpretation for the Bits Associated with Global Activity and Inactivity

SMARTS	bit	interpretation
Significant Features for Activity		
@(@*)@*	105	multiple ring system
@!@[#8].*@*!@[#8]	127	>1 aliphatic oxygen joined to a ring
~1~~*~*~*~*1.*~1~*~*~*~*~*1	145	>1 6-membered rings
[#8]~[#6](~[#6])~[#6]	152	tertiary carbon with 2 carbons and 1 oxygen attached
C=C	99	double-bonded carbons
[CH3].[CH3]	149	>1 methyl group
[CH3].[CH3].[CH3]	141	>2 methyl groups
[CH3]~*~*~[CH2]~*	116	methyl 3 bonds away from a chain carbon
[CH3]~*~[CH2]~*	115	methyl 2 bonds away from a chain carbon
[#7]~[#8]	71	NO
Significant Features for Inactivity		
[#7]~*~[#8]	117	nitrogen one bond away from an oxygen
[#7]~[#6]~[#8]	110	N−C−O
[#8]~[#6](~[#7])~[#6]	92	OC(N)C
[#7]~*~[#7]	77	two nitrogens separated by one bond
[#7]~*~*~[#8]	95	nitrogen two bonds away from an oxygen
[!#6]~*(~[!#6])~[!#6]	106	heteroatom bonded to atom with 2 branched heteroatoms
[#16]	88	sulfur
[#16]~*(~*)~*	81	sulfur off a branched system
[!#6]~[#7]	94	heteroatom bonded to a nitrogen
[#7]~[#6](~[#6])~[#7]	38	NC(C)N

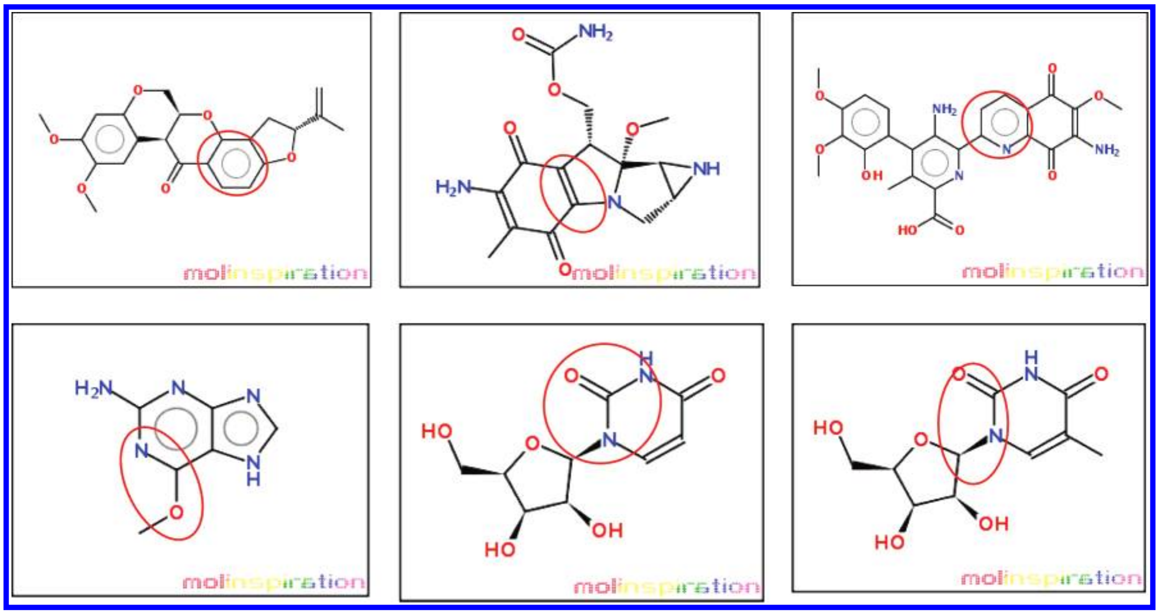

Figure 8. Example compounds which are active in all cell lines (top row) or inactive in all cell lines (bottom row). Depictions were generated by the Molinspiration package, www.molinspiration.com. Features identified in Figure 3 are highlighted.

Table 4. Accuracy of the Prediction Based on Various Structure Features

	based on the rank cross all cell lines				based on the rank over cell line 60			
	AD-Tree		Ridor		AD-Tree		Ridor	
features	inactive	active	inactive	active	inactive	active	inactive	active
10	0.35	0.71	0.24	0.70	0.22	0.71	0.33	0.71
20	0.48	0.71	0.46	0.71	0.47	0.71	0.52	0.71
40	0.60	0.72	0.54	0.71	**0.61**	**0.72**	0.48	0.71
60	**0.62**	**0.72**	0.56	0.71	0.58	0.72	0.61	0.71
80	0.51	0.71	**0.71**	**0.71**	0.33	0.71	**0.62**	**0.72**
100	0.44	0.72	0.64	0.71	0.46	0.75	0.62	0.72
120	0.41	0.71	0.63	0.71	0.46	0.75	0.62	0.72
140	0.46	0.72	0.61	0.71	0.49	0.74	0.61	0.72
166	0.43	0.71	0.59	0.71	0.43	0.71	0.59	0.71

Table 5. Smarts Bond Types

∼	general bond, any possible bond
-!@	single bond, not part of a ring
=!@	double bond, not part of a ring
#	triple bond
-@!:	single ring bond, not aromatic
=@!:	double ring bond, not aromatic
:	aromatic bond

The results of these experiments are shown in tabular form (Table 4) and graphically (Figure 9). Clearly, not all 166 structural features are useful in determining the cell line activity. Our experiments show that the best prediction accuracy for AD-tree only uses 60 structural features and that the best prediction accuracy for Ridor only uses 80 structural features if the features are chosen based on the rank cross all cell lines. By limiting the number of features, we can increase the prediction accuracy for the inactive group from 43% to 62% for AD-tree and from 51% to 71% for Ridor. The best prediction accuracy for AD-tree only uses 40 structural features, and the best prediction accuracy for Ridor only uses 80 structural features if the features are chosen based on the rank over cell line 60. It also shows that the feature selection helps increase the prediction accuracy. Interestingly, the feature selection based on cell line 60 is slightly worse than the feature selection based on all 60 cell lines.

In addition to these methods, we also considered the random forest.[41] This technique has become popular in the data mining community, and there are a number of examples of its use in the chemical informatics literature.[42−44] The random forest is essentially an ensemble of decision trees and is thus an example of a bagging method.[45] The ensemble character of this method leads to some useful characteristics. Most important for our purposes is the fact that to develop a random forest model, one is not required to perform feature selection a priori. In addition, it can be shown that a random forest model does not overfit. That is, increasing the number of trees in the ensemble does not lead to overfitting, and the only real disadvantage is the increase in memory consumption.

We developed 60 random forest models, one for each cell line, using the randomForest package available in R.[46] We considered the 166-bit fingerprints previously described for the input features. For general usage the default settings for the method lead to good results. The main parameter of interest is the number of trees in the ensemble. As noted above, a higher number of trees does not lead to overfitting. However the default value of 500 trees led to excessive memory consumption when we built all 60 models. We investigated a number of values for this parameter and settled on 250 trees. The models were developed on a machine equipped with a 3.2 GHz dual core Xenon CPU and 2 GB

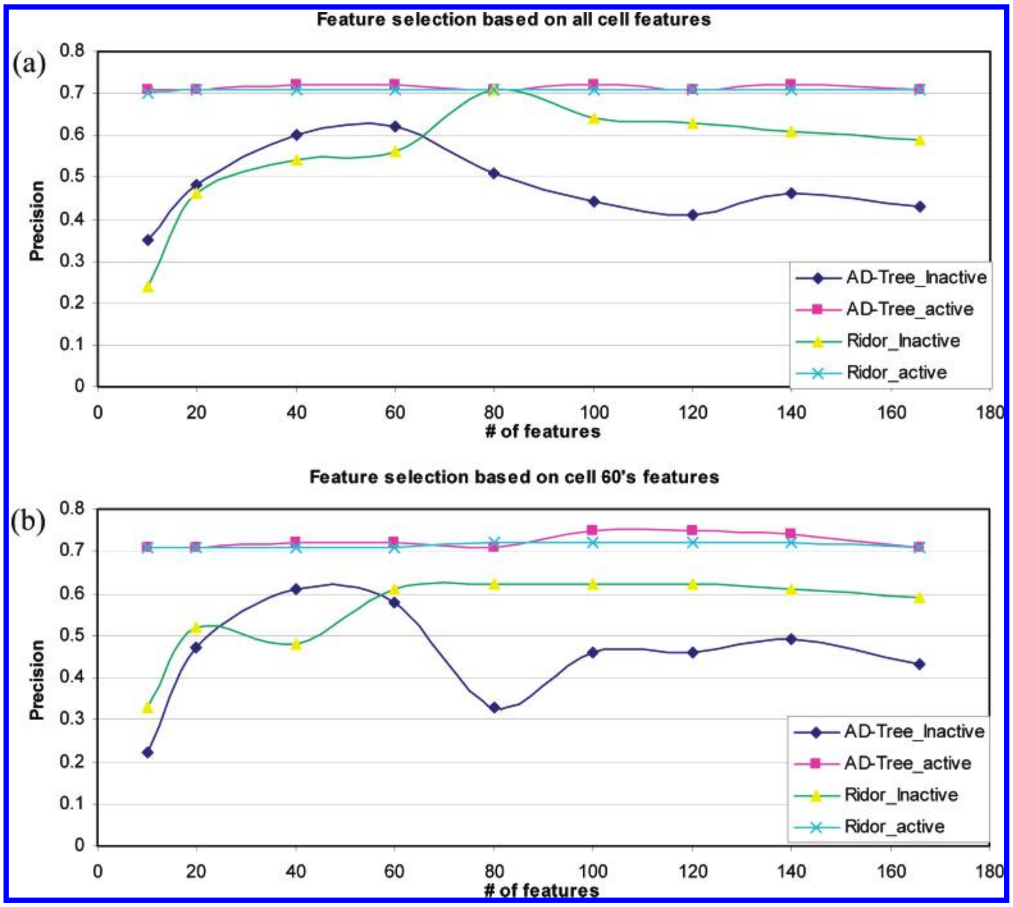

Figure 9. Accuracy of the prediction based on various structural features: (a) structural features are ranked across all cell lines and (b) structure features are ranked over cell line 60.

RAM running Fedora Core 5. On average, the development of a single model took 16.5 min. Since we had a dual core CPU, we processed two cell lines at a time, thus leading to a total run time of 8 h to develop all 60 models. Note that the speed of this process could easily be increased by utilizing one of the many parallel execution packages available for R (such as snow) and a cluster of machines. Alternatives to the random forest could also be considered. Since we are mainly interested in pure predictive ability (as opposed to developing a model of the underlying distribution) one possible approach would be to consider a k-nearest neighbor classification. Though simplistic in nature, this method would be relatively fast, though for larger data sets this may not be such an advantage unless appropriate nearest neighbor detection algorithms were employed. The downside to this and other methods is that some sort of feature selection would need to be performed prior to the prediction step.

As has been noted above, the data sets for each cell line represent an unbalanced classification problem, with the actives being the minor class. As can be seen from Table 4, this leads to very poor predictive performance, since new observations will tend to be classified as inactive, by default. To alleviate this problem in our random forest models, we specified that for each tree in the ensemble the algorithm should consider all the actives as well as a set of randomly selected inactives in the ratio of 1.0:0.6. Thus each tree in the ensemble would not see the highly unbalanced data set but would in fact see a subset that was enriched by the actives. By including a smaller number of inactives, one can effectively force each individual tree to exhibit a high predictive accuracy for the minor (active) class. It is clear that this is simply the reverse of the current situation, where we have very good predictive accuracy for the major (inactives) class. As a result, we experimented with a variety of ratios until we obtained a ratio where the predictive accuracy for the minor and major class were approximately equal. We realize that this approach does lead to a model biased in favor of the actives. We believe that this is justified since our aim is to try and avoid false negatives. Thus by biasing toward the active class, we not only improve the true positive rate but also increase the false positive rate at the expense of the false negative rate. Finally, for each cell line we considered only those observations that had measured values of growth inhibition and split the data sets, such that 70% was placed in a training set and 30% in a test set.

The plots in Figure 10 summarize the predictive accuracy for the 60 models that were developed using the above approach. We consider the predictive accuracy in three ways: Box A represents the range of percentage correct prediction for the test set overall, across the 60 cell lines. For this case we utilized the g-mean measure of accuracy described by Kubat et al.[47] which takes into account the unbalanced nature of the test set. The worst model exhibited a 67% correct accuracy, while the best model exhibited close to 77% correct. Box B represents the percent correct prediction for the actives, across all 60 cell lines. It is clear that the variation in the accuracies for the 60 models is much smaller when the actives are considered in isolation. This is not surprising, since by construction the models are expected to fare better on the actives. Thus we see that the accuracies

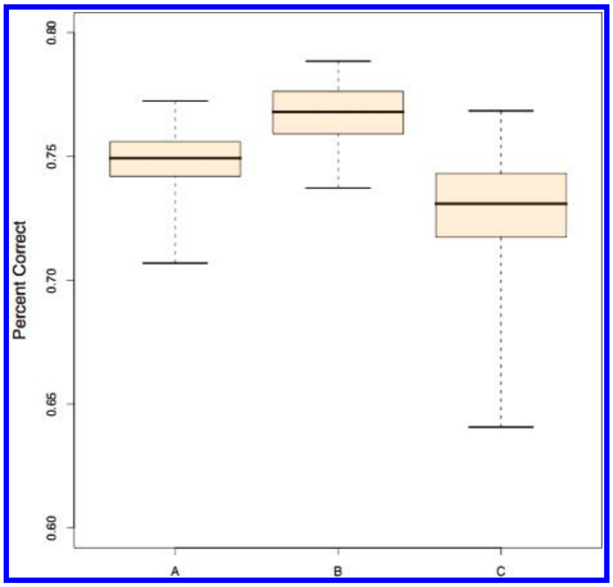

Figure 10. A box and whisker summary of the prediction accuracy for the 60 random forest models developed for the NCI DTP cell lines. Box A is the percent correct accuracy for the overall test set, box B is that for the actives, and Box C is that for the inactives. In each case, the whiskers extend to the extremes of the observed accuracy over the 60 cell lines.

range from 74% to 79% correct. In contrast, Box C represents the percent correct prediction for the inactive class over the 60 cell lines. It is clear that the spread of accuracy is much more than for the actives, and once again this is a result of our model construction. As we noted above, our focus is on identifying actives, thus we accept a slightly poorer performance on the inactive class.

The models have been deployed in our Web service infrastructure,[48] allowing access to predictions from any client that supports SOAP. As an example we have provided a Web page client that allows one to supply a set of SMILES and obtain the predicted activity class for all 60 cell lines. In addition, the probability associated with each classification is also provided. Thus, values greater than 0.5 indicate an increasingly higher probability of being predicted active and correspondingly for values lower than 0.5. The Web page can be accessed at http://www.chembiogrid.org/cheminfo/ncidtp/dtp.

Relating Freely Generated SMARTS Structures to Cellular Screening Activities. Our previous experiments used a constrained dictionary of 166 SMARTS fragments. We were also interested in applying a free-form approach that has been developed at the University of Michigan in which a larger number of SMARTS-based fragment keys are generated. A brute force method of lengthening and scoring SMARTS strings was applied in order to establish SMARTS strings up to seven atoms long that have a strong tendency to identify active and inactive compounds across the cell lines. For this experiment we used an updated version of the NCI/DTP 60 cancer cell line data set obtained through PubChem. A MOE database was created for the 42 888 compounds that had both structural and growth inhibition data in order to perform iterative scoring based SMARTS structural similarity searches. This method tracks active and inactive hits for a set of SMARTS strings across the entire data set. SMARTS strings are then scored, evaluated, ranked, pruned, and extended for subsequent searches.

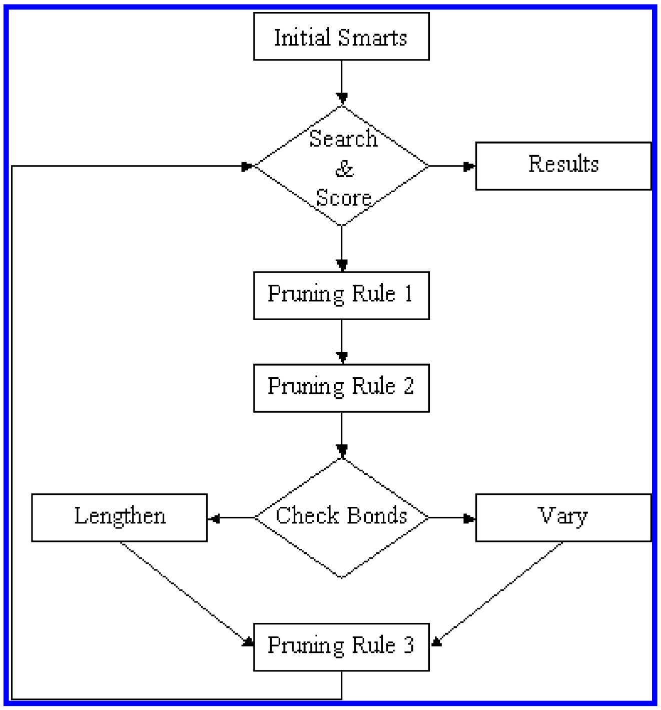

Figure 11. Algorithm workflow.

Scoring is determined by the ratio of active compounds identified by a SMARTS string divided by the number of inactive compounds identified by the same SMARTS string. With this method, scores will range from 0 to ∞. The ratio of active to inactive compounds in the NCI/DTP data set is 7274 to 35 664. If we took a random sampling of the data set we would expect to find one active compound to every five inactive compounds selected. Therefore, the ratio of significance is 1:5 or 0.2. Here we will consider SMARTS strings that demonstrate a tenfold improvement in active or inactive hits as significant. That is, the score of significance for SMARTS strings identifying active compounds is greater than or equal to 2.0 and less than or equal to 0.02 for inactive compounds. Weight can further be given to SMARTS stings, which have a high number of total hits. For example, if SMARTS string A has a score of 5.0 with a total of six hits, five active and one inactive, it is not as significant as SMARTS string B with a score of 5.0 with 240 total hits, 200 active and 40 inactive. In this case SMARTS string A may likely be an artifact of the data set.

Adjusting the scores of significance with the ratio of significance allows one to deal with an unbalanced data set with an even greater skew than the NCI/DTP data set. If the active:inactive ratio of significance were much smaller, for example 1:100 or 0.01, the score of significance for an inactive substance would be taken to be greater than or equal to 0.1. Furthermore, with this strong bias in the data set toward inactives, we would expect that there would be fewer SMARTS strings associated with active substances and more associated with the inactives.

The specific algorithm applied for identifying and lengthening SMARTS strings incorporates three pruning rules at various stages to eliminate redundancies, to improve computational efficiency, and to eliminate artifacts. The workflow of our algorithm is depicted in Figure 11. This procedure was performed on a Dell Precision 380 workstation with 3 gHz CPU with 1 GB RAM. Runtimes for each iteration of

Table 6. Most Common Single Atom SMARTS Strings in the NCI/DTP Data Set

SMARTS strings	element	42 888 compounds	score
[#6]	C	42 845	0.2044
[#8]	O	38 674	0.1965
[#7]	N	34 992	0.1967
[#16]	S	11 969	0.1555
[#17]	Cl	8483	0.2772
[#9]	F	2557	0.2246
[#35]	Br	1832	0.2820
[#15]	P	1305	0.1929
[#53]	I	617	0.2390
[#14]	Si	349	0.2246
[#11]	Na	302	0.0942
[#50]	Sn	198	2.1936
[#78]	Pt	189	0.4427
[#5]	B	136	0.1525

the algorithm were based on the size of the SMARTS string set and ranged from to 2 min to 11 h, for sets on the order of 100 and 20 000, respectively.

The details of the steps performed are as follows:

1. Select Initial SMARTS Strings.

   For the sake of generality, elements 2−105 of the periodic table were selected as single atom SMARTS strings. Hydrogen was not included in this SMARTS string set, as SMILES strings and the molecular connectivity tables provided typically suppress hydrogen atoms.

2. Search & Score

   A substructure search was performed against the NCI/DTP data set using the SMARTS string set. Scores were tabulated, and a bit string hit profile was maintained for each individual SMARTS string across all 42 888 compounds. A bit string hit profile consists of a string of 42 888 1's and 0's, where 1 means that the SMARTS string is found within the compound, and 0 means that the SMARTS string could not be found within the compound.

3. Record incremental SMARTS String Results.

   If SMARTS Strings contain seven atoms and no general bond types, then terminate the algorithm.

4. Apply Pruning Rule 1 to eliminate redundancies.

   Maintain only one SMARTS string child per unique bit string hit profile. The lengthening of SMARTS strings is a tree process leading to the exponential generation of child SMARTS strings. Bit string profiles are used in order to limit branching as they serve to identify all duplicate SMARTS strings as well as SMARTS strings that do not hit any compounds. Pruning will improve the efficiency of subsequent substructural searches.

5. Apply Pruning Rule 2 to improve computational efficiency.

   If the number of SMARTS string children exceeds 24000, then drop all parent SMARTS strings having scores in the range [0.2/X and 0.2*X]. Starting with $X = 1.5$, increase X in increments of 0.1 until the number of SMARTS string children ← 24 000.

6. Check Bonds to select rules for generating child SMARTS strings.

   a. Vary Bond: If the parent SMARTS strings contain general bonds, then generate all possible SMARTS string children by varying the bond type. For SMARTS strings with fewer than five atoms all six specific bond types were used. For SMARTS strings with five atoms or more, the triple bond was disregarded. See Table 5 for a description of the bond types.

   b. Lengthen: If the parent SMARTS strings do not contain any general bonds (∼), then generate all possible SMARTS string children by joining a single atom to all the potential locations on the SMARTS strings with a general bond. For SMARTS strings with fewer than five atoms, the following atoms were appended to the parent SMARTS string: B, C, N, O, Si, P, S, F, Cl, Br, and I. These elements were selected, as they are among the most common in the PubChem compound data set. Table 6 shows the 14 most common single atom SMARTS strings found in the NCI/DTP data set based on the number of compounds identified. Na, Sn, and Pt were not included because our SMARTS strings only consider covalently bound atoms. For SMARTS strings with five or more atoms, C, O, N, P, and S were appended to the parent SMARTS strings. We limit the number of atoms based on the most common nonmetals in order to keep the number of children SMART strings in check. Using common elements allows generation of SMARTS string children that will hit compounds in the data set.

7. Apply Pruning Rule 3 to eliminate artifacts and improve computational efficiency.

Table 7. Description of Results[a]

no. of SMARTS atoms	SMARTS (possible)	SMARTS (used)	SMARTS (hits)	Active (only)	Active (mostly)	Inactive (only)	Inactive (mostly)	Score Range	Data Set Covered
1	105	104	67	1(0)	4(3)	13(3)	0(0)	0.0313−6.25	42 888
2	6930	690	133	4(0)	11(7)	16(1)	1(1)	0.0127−23.0	42 876
3	914760	2094	540	13(1)	26(21)	88(12)	3(3)	0.0132−23.0	42 871
4	1.81E+08	10 248	2470	45(1)	73(49)	481(89)	12(12)	0.0127−31.0	42 862
5	4.78E+10	22 584	18815	52(1)	48(19)	1232(318)	36(36)	0.00873−20.0	42 752
6	5.98E+12	8150	8146	31(1)	66(55)	877(264)	83(83)	0.00532−12.5	31 762
7	8.97E+14	17 155	6470	161(1)	304(204)	1814(359)	121(121)	0.00532−18.0	21 253

[a] SMARTS (possible) is the total number of possible SMARTS strings. SMARTS (used) represents the set of SMARTS strings used in each iterative search after pruning. SMARTS (hits) is the number of SMARTS strings with unique bit string profiles. Active/Inactive (only) represent SMARTS strings identifying compounds that are only active and inactive, respectively. Active/Inactive (cutoff) represent SMARTS scoring >2.0 and <0.02, respectively. Integers within parentheses () indicate the number of significant SMARTS that have a minimum of 10 active or inactive hits. Score Range is minimum − maximum score. Data Set Covered is the number of compounds hit out of 42 888. The significant drop in Data Set Covered for the last two rows resulted from Pruning Rule 2.

Table 8. Some of the Most Significant SMARTS Strings

order	SMARTS	Total Hits	Score
Active (only)	[#90]	1	
	[#8]-!@[#25]	3	
	[#6]-@!:[#6]-!@[#50]	16	
	[#6]-@!:[#6]-!@[#50]-!@[#6]	13	∞
	[#8]:[#6]-!@[#7]-!@[#7]=!@[#6]	10	
	[#6]:[#6]-@!:[#6](=@!:[#7])-@!:[#6]:[#6]	13	
	[#7]-!@[#6]:[#6]-@!:[#6](=!@[#7])-@!:[#6]:[#6]	12	
Active (mostly)	[#79]	29	6.25
	[#15]-!@[#79]	24	23.0
	[#6]-!@[#15]-!@[#79]	24	23.0
	[#7]-@!:[#6]-@!:[#16]-@!:[#29]	32	31.0
	[#6]-@!:[#6]:[#6]-!@[#6]=!@[#7]	21	20.0
	[#6]:[#6]-@!:[#6]-!@[#8]-!@[#6]-@!:[#8]	81	12.5
	[#6]-!@[#8]-!@[#6]-@!:[#6]:[#6]:[#6]-@!:[#6]	80	15.0
Inactive (mostly)	[#16]-@!:[#8]	80	0.0127
	[#7]:[#16]:[#6]	77	0.0132
	[#6]-@!:[#7]-!@[#7]-@!:[#6]	80	0.0127
	[#8]=!@[#6]-!@[#6]-!@[#6]=!@[#7]	231	0.00873
	[#8]=!@[#6]-!@[#6]-!@[#6]=!@[#7]-!@[#7]	190	0.00529
	[#8]=!@[#6]-!@[#6]-!@[#6](=!@[#7]-!@[#7])-!@[#6]	189	0.00532
Inactive (only)	[#12]	14	
	[#7]-!@[#27]	46	
	[#7]=!@[#6]-!@[#5]	29	
	[#7]=!@[#6]-!@[#7]=!@[#7]	75	0
	[#6]=!@[#6]-@!:[#7]-@!:[#6]-@!:[#7]	147	
	[#7]:[#6](:[#6]-!@[#6]-!@[#6])-!@[#8]	200	
	[#6]:[#7]:[#6](-!@[#6]-!@[#6]):[#6]-!@[#8]	178	

Table 9. Scoring Selective MACCS SMARTS Strings

type	MACCS SMARTS String	no of. Active Hits	no. of Inactive Hits	Score
Active	*@*(@*)@*	5102	19 855	0.2570
Active	*@*!@[#8].*@*!@[#8]	3192	10 256	0.3112
Active	*~1~*~*~*~*~*1.*~1~*~*~*~*~*1	5589	24 358	0.2295
Active	[#8]~[#6](~[#6])~[#6]	4836	19 565	0.2472
Active	C=C	3275	12 144	0.2697
Active	[CH3].[CH3]	3853	15 978	0.2411
Active	[CH3].[CH3].[CH3]	2470	9006	0.2743
Active	[CH3]~*~*~[CH2]~*	2099	7013	0.2993
Active	[CH3]~*~[CH2]~*	1921	5691	0.3376
Active	[#7]~[#8]	804	4296	0.1872
Active	Boolean AND (5 highest scoring Active)	440	1034	0.7407
Active	Boolean AND (All Active)	9	21	0.7500
Active	Boolean OR (All Active)	7246	35 104	0.2064
Inactive	[#7]~*~[#8]	2595	17 823	0.1456
Inactive	[#7]~[#6]~[#8]	2383	16 376	0.1455
Inactive	[#8]~[#6](~[#7])~[#6]	2123	14 314	0.1483
Inactive	[#7]~*~[#7]	2039	13 063	0.1561
Inactive	[#7]~*~*~[#8]	2407	14 617	0.1647
Inactive	[!#6]~*(~[!#6])~[!#6]	2121	13 468	0.1573
Inactive	[#16]	1611	10 358	0.1555
Inactive	[#16]~*(~*)~*	1458	9636	0.1513
Inactive	[!#6]~[#7]	2261	12 817	0.1764
Inactive	[#7]~[#6](~[#6])~[#7]	944	6456	0.1462
Inactive	Boolean AND (5 lowest scoring Active)	81	927	0.08738
Inactive	Boolean AND (All Active)	32	272	0.1176
Inactive	Boolean OR (All Active)	5275	28 573	0.1846

For SMARTS strings with fewer than five atoms, drop all children SMARTS strings with less than 20 total hits. For SMARTS strings having scores with five atoms or more, drop all children SMARTS strings with fewer than 100 total hits.

8. Go to Step 2.

Table 7 describes the overall results generated by our algorithm. It includes the data for SMARTS strings with modifications to all possible positions at which atoms may be added, subject to pruning as noted within the algorithm. Table 8 gives examples of the most selective SMARTS.

We then tested the SMARTS strings from the 166-bit fingerprints with the scoring system from this method. Based on the ratio of significance, the individual SMARTS strings for identifying the active and inactive compounds showed minimal increase and decrease in relative score. We identified all compounds that contained all active motifs and inactive motifs, respectively. When considering collections of low and high scoring motifs in a Boolean AND operation, a 2−4-fold respective increase in selectivity was identified. Furthermore, it was found that when combining more than five MACCS SMARTS strings the score minimally increased or decreased; however, the total number of hits significantly decreased. See Table 9 for details. We then tabulated the Boolean OR incorporating all active and inactive SMARTS strings from the MACCS example. Almost all compounds were selected, and the score of significance for both active and inactive sets was ∼0.2.

We took the Boolean OR for the four sets of SMARTS from this example. As our sets of SMARTS strings were tailored to the NCI60, we expected and confirmed that they outperform the MACCS fingerprints. As one would expect the Active(only) and Inactive(only) sets had scores of ∞ and 0, respectively. The Inactive(mostly) set hit a total of 165 active compounds and 9372 inactive compounds, yielding a score of 0.01761. The Active(mostly) set hit a total of 2999 active compounds and 9949 inactive compounds, respectively, yielding a score of 0.3014. It appears that the Inactive-(mostly) set has been better tailored to identifying inactive compounds due to the low threshold score of 0.02 for each SMARTS string. From this, it can be inferred that there was very little overlap of inactive and active compounds identified. However in the case of the Active(mostly) set, there was obviously considerable overlap. Suppose SMARTS string A identifies two active compounds and one inactive compound, while SMARTS string B identifies the same two active compounds, it identifies a different inactive compound. If we were to use Boolean OR, tabulating a new score when both SMARTS A and B were used together, the new score would be equal to 1.0 as two active compounds are identified by both SMARTS and two inactive compounds are identified, one by SMARTS string A and the other by SMARTS string B. Therefore, due to the low threshold score required for the Active(mostly) SMARTS strings, we cannot group their properties with the Boolean OR and expect significant active hit enrichment, but rather they must be used discretely in order to maintain scores greater than or equal to 2.0. At this juncture, it would be wise to identify the Active(mostly) SMARTS strings with overlapping active and inactive compounds. Further pruning needs to be performed on the SMARTS strings sharing the same set of active compounds in order to obtain the most orthogonal set. This can be accomplished by maintaining only one SMARTS string identifying a specific set of active compounds and dropping all SMARTS strings identifying equal sized or larger sets of different inactive compounds.

Finally, the most significant SMARTS strings can be used to create molecular fingerprints to give a general prediction regarding the activity of compounds yet to be assayed. This method may be further complemented by addressing the activity profiles of compounds identified by multiple selective SMARTS strings. Also one might consider creating profiles for each of the individual 60 cancer cell line assays and weighting the SMARTS strings based on the growth inhibition value, rather than the binary interpretation used in this method with ‘1’ representing an active hit and ‘0’ an inactive hit in order to give a more quantitative growth inhibition-predictions.

CONCLUSIONS AND FUTURE WORK

In this work, we have conducted broad characterizations of the compounds, biological activities, and gene expression values in the NIH DTP Tumor cell line data set. We have shown that compounds active or inactive across the 60 cell lines tend to have structural features in common. We have also demonstrated that a Random Forest model can be used to predict the activity profiles of unknown compounds across the cell lines reasonably well. Finally, we show that a novel SMARTS-based algorithm can be used to give finer resolution structure−activity correlations than a constrained dictionary-based fingerprint.

We are currently in the process of extending our data mining to include the gene expression information, in particular finding features that tend to be associated with activity or inactivity in subgroups of the cell lines which share particular gene expression profiles. We also wish to extend our random forest models to include information from other cell lines in our prediction of individual cell line activities.

ACKNOWLEDGMENT

This work was supported by NIH grant NIH-NHGRI P20 HG003894-02. We would like to thank Dr. Gary Wiggins and Dr. Geoffrey Fox for assistance in facilitating this research.

REFERENCES AND NOTES

(1) Developmental Threapeutics Program Web site. http://dtp.nci.nih.gov (accessed July 23, 2007).

(2) Weinstein, J. N.; Myers, T. G.; O’Connor, P. M.; Friend, S. H.; Fornace, Jr, A. J.; Kohn, K. W.; Fojo, T.; Bates, S. E.; Rubinstein, L. V.; Anderson, N. L.; Buolamwini, J. K.; van Osdol, W. W.; Monks, A. P.; Scudiero, D. A.; Sausville, E. A.; Zaharevitz, D. W.; Bunow, B.; Viswanadhan, V. N.; Johnson, G. S.; Wittes, R. E.; Paull, K. D. An Information-Intensive Approach to the Molecular Pharmacology of Cancer. *Science* **1997**, *275*, 343−349.

(3) Paull, K. D.; Shoemaker, R. H.; Hodes, L.; Monks, A. P.; Scudiero, D. A.; Rubinstein, L. V.; Plowman, J.; Boyd, M. R. Display and Analysis of Patterns of Differential Activity of Drugs against Human Tumor Cell Lines: Development of a Mean Graph and COMPARE algorithm. *J. Natl. Cancer Inst.* **1989**, *81*, 1088−1092.

(4) Zaharevitz, D. W.; Holbeck, S. L.; Bowerman, C.; Svetlik, P. A. COMPARE: A Web Accessible Tool for Investigating Mechanisms of Cell Growth Inhibition. *J. Mol. Graphics Modell.* **2002**, *20*, 297−303.

(5) DTP Data Search Page. http://dtp.nci.nih.gov/docs/dtp_search.html (accessed July 23, 2007).

(6) Weinstein, J. N.; Kohn, K. W.; Grever, M. R.; Viswanadhan, V. N.; Rubinstein, L. V.; Monks, A. P.; Scudiero, D. A.; Welch, L.; Koutsoukos, A. D.; Chiausa, A. J.; Paull, K. D. Neural Computing in Cancer Drug Development: Predicting Mechanism of Action. *Science* **1992**, *258*, 447−451.

(7) Scherf, U.; Ross, D. T.; Waltham, M.; Smith, L. H.; Lee, J. K.; Tanabe, L.; Kohn, K. W.; Reinhold, W. C.; Myers, T. G.; Andrews, D. T.; Scudiero, D. A.; Eisen, M. B.; Sausville, E. A.; Pommier, Y.; Botstein, D.; Brown, P. O.; Weinstein, J. N. A gene expression database for the molecular pharmacology of cancer. *Nat. Genet.* **2000**, *24*, 236−244.

(8) Rabow, A. A.; Shoemaker, R. H.; Sausville, E. A.; Covell, D. G. Mining the National Cancer Institute’s Tumor-Screening Database: Identification of Compounds with Similar Cellular Activities. *J. Med. Chem.* **2002**, *45*, 818−840.

(9) Shi, L. M.; Myers, T. G.; Fan, Y.; O’Connor, P. M.; Paull, K. D.; Friend, S. H.; Weinstein, J. N. Mining the National Cancer Institute Anticancer Drug Discovery Database: Cluster Analysis of Ellipticine Analogs with p53-Inverse and Central Nervous System-Selective Patterns of Activity. *Mol. Pharmacol.* **1998**, *53*, 241−251.

(10) Shi, L. M.; Fan, Y.; Myers, T. G.; O’Connor, P. M.; Paull, K. D.; Friend, S. H.; Weinstein, J. N. Mining the NCI Anticancer Drug Discovery Databases: Genetic Function Approximation for the QSAR Study of Anticancer Ellipticine Analogues. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 189−199.

(11) Roberts, G.; Myatt, G. J.; Johnson, W. P.; Cross, K. P.; Blower, P. E. Leadscope: Software for Exploring Large Sets of Screening Data. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1302−1314.

(12) Blower, P. E.; Yang, C.; Fligner, M. A.; Verducci, J. S.; Yu, L.; Richman, S.; Weinstein, J. N. Pharmacogenomic Analysis: Correlating

Molecular Substrucure Classes with Microarray Gene Expression Data. *Pharmacogenomics J.* **2002**, *2*, 259−271.

(13) Huang, Y.; Blower, P. E.; Yang, C.; Barbacioru, C.; Dai, Z.; Zhang, Y.; Xiao, J. J.; Chan, K. K.; Sadée, W. Correlating Gene Expression with Chemical Scaffolds of Cytotoxic Agents: Ellipticines as Substrates and Inhibitors of MDR1. *Pharmacogenomics J.* **2005**, *5*, 112−125.

(14) Blower, P. E.; Cross, K. P.; Fligner, M. A.; Myatt, G. J.; Verducci, J. S.; Yang, C. Systematic Analysis of Large Screening Sets in Drug Discovery. *Curr. Drug Discovery Technol.* **2004**, (1), 37−47.

(15) Richter, L.; Rückert, U.; Kramer, S. In *Learning a Predictive Model for Growth Inhibition from the NCI DTP Human Tumor Cell Line Screening Data: Does Gene Expression Make a Difference?* Pac. Symp. Biocomput., 2006; 2006; pp 596−607.

(16) Cho, S. J.; Shen, C. F.; Hermsmeier, M. A. Binary Formal Inference-Based Recursive Modeling Using Multiple Atom and Physicochemical Property Class Pair and Torsion Descriptors as Decision Criteria. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 668−680.

(17) Fang, X.; Shao, L.; Zhang, H.; Wang, S. Web-Based Tools for Mining the NCI Databases for Anticancer Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 249−257.

(18) Han, J.; Kamber, M. *Data Mining: Concepts and Techniques*, 1st ed.; Morgan Kaufmann: 2000.

(19) Gray, J.; Chaudhuri, S.; Bosworth, A.; Layman, A.; Reichart, D.; Venkatrao, M.; Pellow, F.; Pirahesh, H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Min. Knowledge Discovery* **1997**, 29−53.

(20) Downs, G. M.; Barnard, J. M. Clustering Methods and Their Uses in Computational Chemistry. *Rev. Comput. Chem.* **2002**, *18*, 1−40.

(21) MacQueen, J. B. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*; 1967; pp 281−297.

(22) Kaufman, L.; Rousseeuw, P. J. *Findings Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: New York, 1990.

(23) Ng, R. T.; Han, J. In *Efficient and effective clustering methods for spatial data mining;* 1994 International Conference Very Large Data Bases (VLDB'94), Santiago, Chile, 1994; Santiago, Chile, 1994; pp 144−155.

(24) Zhang, T.; Ramakrishnan, R.; Livny, M. In *BIRCH: An efficient data clustering method for very large databases;* 1996 ACM-SIGMOD International Conference Management of Data (SIGMOD '96), Montreal, Canada, 1996; Montreal, Canada, 1996; pp 103−114.

(25) Guha, S.; Rastogi, R.; Shim, K. In *Cure: An efficient clustering algorithm for large databases*; 1998 ACM-SIGMOD International Conference Management of Data, Seattle, WA, 1998; Seattle, WA, 1998; pp 73−84.

(26) Karypis, G.; Han, E.-H.; Kumar, V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *COMPUTER* **1999**, 68−75.

(27) Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. In *A density-based algorithm for discovering clusters in large spatial databases;* 1996 International Conference of Knowledge Discovery and Data Mining (KDD'97), Portland, OR, 1996; Portland, OR, 1996; pp 226−231.

(28) Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; Sander, J. In *OPTICS: Ordering points to identify the clustering structure;* 1999 ACM-SIGMOD International Conference Management of Data (SIGMOD'99), Philadelphia, PA, 1999; Philadelphia, PA, 1999; pp 49−60.

(29) Hoschka, P.; Klosgen, W. A support system for interpreting statistical data. In *Knowledge Discovery in Databases*; AAAI/MIT Press: Cambridge, MA, 1991; pp 325−346.

(30) Wang, W.; Yang, J.; Muntz, R. R. In *STING: A statistical information grid approach to spatial data mining;* 1997 International Conference of Very Large Data Bases (VLDB'97), Athens, Greece, 1997; Athens, Greece, 1997; pp 186−195.

(31) Sheikholeslami, G.; Chatterjee, S.; Zhang, A. In *WaveCluster: A multi-resolution clustering approach for very large spatial databases;* 1998 International Conference of Very Large Data Bases, New York, 1998; New York, 1998; pp 428−439.

(32) Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. In *Automatic subspace clustering of high dimensional data for data mining applications;* 1998 ACM-SIGMOD International Conference Management of Data (SIGMOD'98), Seattle, WA, 1998; Seattle, WA, 1998; pp 94−105.

(33) Agrawal, R.; Imielinski, T.; Swami, A. Mining Association Rules between Sets of Items in Large Databases. *ACM SIGMOD* **1993**, 207−216.

(34) Available from gNova.com.

(35) Elsevier MDL. http://www.mdl.com (accessed July 23, 2007).

(36) FDA MRTD data set. http://www.fda.gov/CDER/Offices/OPS_IO/MRTD.htm (accessed July 23, 2007).

(37) Pubchem. http://pubchem.ncbi.nlm.nih.gov/ (accessed July 23, 2007).

(38) OpenEye. http://www.eyesopen.com (accessed July 23, 2007).

(39) Frank, I. H. W. a. E. *Data Mining: Practical machine learning tools and techniques*; Morgan Kaufmann: San Francisco, CA, 2005.

(40) Daylight SMARTS. http://www.daylight.com (accessed July 23, 2007).

(41) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and regression trees*; CRC Press: Boca Raton, FL, 1984.

(42) Guha, R.; Jurs, P. C. Development of a Linear, Ensemble, and Nonlinear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. *J. Chem. Inf. Model.* **2004**, *44* (6), 2179−2189.

(43) O'Brien, S. E.; deGroot, M. J. Greater than the Sum of its Parts: Combining Models for Useful ADMET Prediction. *J. Med. Chem.* **2005**, *48* (4), 1287−1291.

(44) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, C.; R. P., S.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *42*, 1947−1958.

(45) Breiman, L. Bagging Predictors. *Machine Learning* **1996**, *26*, 123−140.

(46) Team, R. D. C. *A language and environment for statistical computing;* Foundation for Statistical Computing: Vienna, Austria, 2006.

(47) Kubat, M.; Holte, R. C.; Matwin, S. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning* **1998**, *30* (2−3), 195−215.

(48) Dong, X.; Gilbert, K. E.; Guha, R.; Heiland, R.; Kim, J.; Pierce, M.; Fox, G. C.; Wild, D. J. Web Service Infrastructure for Cheminformatics. *J. Chem. Inf. Model.* **2007**, *47* (4), 1303−1307.

CI700141X