Mining Statistically Significant Molecular Substructures for Efficient Molecular Classification

Sayan Ranu* and Ambuj K. Singh*

Department of Computer Science, University of California, Santa Barbara, California

Received January 28, 2009

The increased availability of large repositories of chemical compounds has created new challenges in designing efficient molecular querying and mining systems. Molecular classification is an important problem in drug development where libraries of chemical compounds are screened and molecules with the highest probability of success against a given target are selected. We have developed a technique called *GraphSig* to mine significantly over-represented molecular substructures in a given class of molecules. GraphSig successfully overcomes the scalability bottleneck of mining patterns at a low frequency. Patterns mined by GraphSig display correlation with biological activities and serve as an excellent platform on which to build molecular analysis tools. The potential of GraphSig as a chemical descriptor is explored, and support vector machines are used to classify molecules described by patterns mined using GraphSig. Furthermore, the over-represented patterns are more informative than features generated exhaustively by traditional fingerprints; this has potential in providing scaffolds and lead generation. Extensive experiments are carried out to evaluate the proposed techniques, and empirical results show promising performance in terms of classification quality. An implementation of the algorithm is available free for academic use at http://www.uweb.ucsb.edu/∼sayan/software/GraphSig.tar.

INTRODUCTION

In modern pharmaceutical research, molecular classification plays a key role in lead generation and lead optimization.[1,2] Consequently, molecular classification has been a vibrant research field. Various techniques have been developed including neural network techniques,[3,4] Bayesian models,[5,6] kernel-based methods,[7−9] cell-based or statistical partitioning methods,[10] decision trees or recursive partitioning based methods,[11−13] and subgraph mining-based techniques.[14] Typically, the goal of the mining process is to infer chemical or biological properties of a molecule from its structure, popularly known as the quantitative structure−activity relationship (*QSAR*) approach,[15,16] so that a reduced set of potential hits can be selected from large virtual libraries.[17] The QSAR approach is based on the assumption that the molecular structure is a good indicant of chemical properties. Therefore, a crucial step in analyzing molecular structures is to extract the chemically informative content embedded in the structure.

A number of classification approaches are based on a feature vector representation of the molecules,[5,8,9,12−14] and consequently, extensive research has been performed on analyzing molecular structures and representing them in a virtual space. Toward this goal, chemical descriptors or fingerprints have been widely used to represent molecules in the form of a feature vector, which counts or indicates the presence of certain features in a molecule.[14,18−20] Typically, these features are pharmacophoric features, such as hydrogen bond donors, hydrogen bond acceptors, and hydrophobic centers, or small substructures. The features can therefore be viewed as reference points to describe molecules, and thus, it is of utmost importance to choose them wisely.

Existing vector-based approaches, in the context of mapping molecules to chemical descriptors, can be broadly grouped into two sets: structural keys[18,23] and hashed fingerprints.[19,24] Fingerprints characterize a molecule by generating a fixed width bit vector. Typically, fingerprints are generated by enumerating all cycles and linear paths up to a certain size and hashing each of the structural features into the vector. As a result, fingerprints are able to encode a large number of features in a relatively compact manner. On the other hand, structural keys contain a dictionary of features that are used to screen a molecule and generate its vector representation. The dictionary is known a priori where the features are chosen based on some domain information. A variant of this technique takes a dynamic approach to build the dictionary.[14,21,22] The dictionary is populated by mining features from the given database of molecules and therefore, like fingerprints, is able to automatically adapt to any molecular database.

In recent years, promising classification results have been observed in the second setting, where dynamic features are employed to characterize molecules. One popular approach is to mine frequent substructures from molecules with known chemical and biological properties.[21,22,25] The mined structures are then used to represent molecules in the virtual space in the form of feature vectors or histograms that indicate the presence or absence of the mined features. Typically in this approach, a frequency threshold is supplied and all substructures that occur at a frequency more than the given threshold are used as part of the descriptor. However, frequent substructures may not provide the best characterization of the molecules since frequency may not always be

* E-mail: sayan@cs.ucsb.edu (S.R.); ambuj@cs.ucsb.edu (A.K.S.).

10.1021/ci900035z CCC: $40.75 © 2009 American Chemical Society
Published on Web 10/28/2009

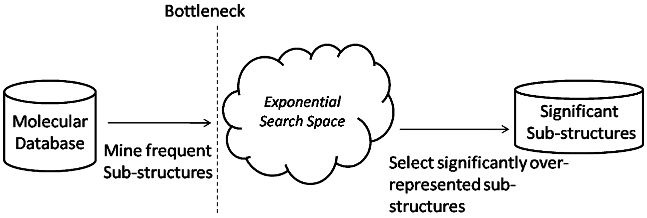

Bottleneck
Molecular Database
Mine frequent Sub-structures
Exponential Search Space
Select significantly over-represented sub-structures
Significant Sub-structures

Figure 1. Straightforward approach to mining significant subgraphs.

related to a discriminative chemical property. For example, consider benzene which is a highly common substructure across all types of chemical compound databases. Even though it is frequent, its usage does not add much discriminative power to the descriptor since benzene is unlikely to indicate any biological or chemical activity. What would be more interesting is to employ those substructures as descriptors that are over-represented in the subset of molecules exhibiting a particular activity when compared to a random set of molecules.

As shown in Figure 1, the straightforward solution would be to filter out frequent substructures that are not statistically significant. This approach, however, may not scale to large data sets primarily due to two reasons. First, since significant subgraphs exist at all frequencies, we need to mine at a very low frequency threshold. For example, a substructure with observed frequency 1% can be significant if the expected frequency is even lower. However, running times of all frequent substructure mining techniques[26−30] grow exponentially with decreasing frequency.[31] As a result, the pipeline in Figure 1 is rendered useless if we are forced to use a low frequency threshold. Further, calculating the frequency of each frequent sub-structure involves performing a subgraph isomorphism test, which is an NP-Complete (where NP stands for nondeterministic polynomial time) problem.

In this paper, we overcome the above-mentioned bottlenecks and present a highly scalable method called *GraphSig* to mine statistically significant substructures from large molecular databases. The application of mining significant substructures lies in multiple areas of molecular database analysis. Given a set of molecules active against a particular disease, GraphSig allows a chemist to ask “What parts of these molecules occur more than one would expect in a database of arbitrary molecules?” Furthermore, the proposed method is highly flexible and, as shown later, can be easily tuned to incorporate domain specific information. To summarize, the paper makes the following contributions to the field of chemoinformatics:

- The presence of significantly over-represented substructures in a molecule indicates its activity, and therefore provides an excellent platform to build classifiers. As shown later, a support vector machine (SVM) achieves extremely promising performance while classifying descriptors formed using the significant patterns. While we demonstrate the potential of significant patterns only through SVM, any other classification technique that works with a vector representation of molecules could be used.
- The significant patterns mined by GraphSig are meaningful on their own. The experimental section finds correlation between the patterns and biological activities. The patterns can potentially provide scaffolds and help in lead optimization during drug design, thereby, having higher information content than descriptors employing exhaustive structural features.

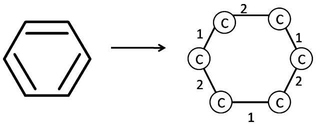


Figure 2. Conversion of benzene to a graph.

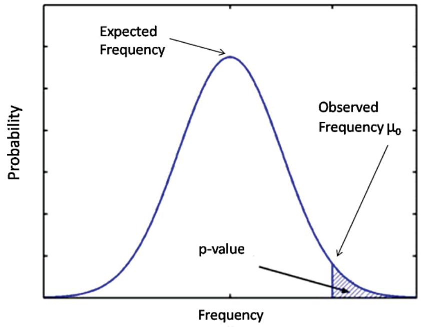


Figure 3. Measuring the *p*-value from the probability distribution function.

METHODS

Overview. Labeled graphs provide a natural structural representation of molecules, where atoms are represented as nodes, edges are formed by the covalent bonds between the atoms, and edge labels correspond to bond order. As prevalent in the topological representation of molecules, H atoms are not defined explicitly. Figure 2 shows the graph representation of benzene.

The graph-based representation of molecules opens up the opportunity to apply powerful graph mining techniques for molecular analysis. Molecular databases can be treated as graph databases, and the problem of finding significant molecular substructures reduces to finding significant subgraphs. In our work, we utilize this aspect of molecular structures and develop techniques based on the graph-based representations.

Since the goal of our work is to mine significantly over-represented substructures, we first need a measure to quantify the significance of any given substructure. Historically, the *p*-value has been widely used to measure significance and is formally defined as follows in the context of substructures:

Definition 1: Statistical Significance. The statistical significance (or p-value) of a substructure g with an observed frequency μ_0 is defined as the probability that it occurs in a random database with a frequency μ, where $\mu \geq \mu_0$.

As shown in Figure 3, the *p*-value of substructure can be calculated by measuring the area to the right of the observed frequency under its *probability distribution function (pdf)*. Clearly, the lower the *p*-value of a substructure, the more significant it is.

The goal of our work is to mine molecular substructures with *p*-values below a user-specified threshold η, and apply the mined structures to generate chemical descriptors for classification. Mathematically, we want to find the answer set A in a molecular database $\mathbb{M}$ such that

$$A = \{g | p\text{-value}(g) \leq \eta, g \subseteq G, G \in \mathbb{M}\} \qquad (1)$$

Incorporating Domain Knowledge on Molecular Graphs. In this section, we introduce a flexible model to incorporate domain information for molecular graphs. We

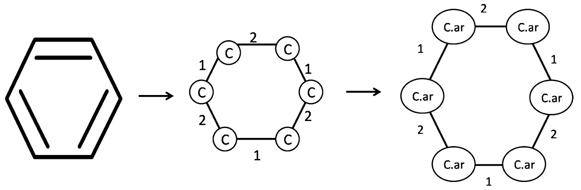


Figure 4. Atom labels after applying the Joelib2 atom-typer.

analyze two different approaches. The first approach is based on enhancing the atom labels in the graph. Instead of just using the atom label, we further divide the atom types based on their physiochemical characteristics. Relabeling of atoms allows us to distinguish between similar atom types in different hybridization states or aromaticities. An example is shown in Figure 4. Variants of this approach have been tried in some previous works as well.[32,33] We use the Joelib2 atom-typer[23] for relabeling the atom types. While the atom-typer is run on all molecular graphs in the database, for illustrative purposes, we use the basic atom types in all examples that follow.

The second approach is based on characterizing the molecules at a coarser level than just atoms. We use functional groups to better reflect the properties of a molecule since they play a crucial role in determining chemical reactivity. Toward this goal, we prepare the list of functional groups in Table 1 and search for their presence in each molecule. If the presence of a functional group is detected in a molecule, the functional group is abstracted out and replaced by a representative single node. This approach of using a reduced graph representation has been applied before and is known to demonstrate good performance in similarity searching.[34−38] A variant of this approach exists in the form of a feature tree, where a molecule is represented as a node-labeled unrooted tree.[39,40]

Owing to the transformation, a node in the reduced molecular graph could represent either an actual atom or a functional group. An example is shown in Figure 5. The list of functional groups can certainly be made more comprehensive or fine-tuned for the targeted data set. However, the goal of our experiment is to study its basic effects on classification quality.

Abstracting functional groups allows us to detect the presence of a functional group in the molecule rather than just the atoms that constitute them. However, we are confronted with the ambiguity on ordering the priorities of the functional groups. Often the structures of the functional groups overlap, and in such cases, only one of them can be abstracted out. To resolve the order dependency, we implemented four different ordering schemes based on the frequency and size of functional groups. Four different orderings were generated by ranking the functional groups based on increasing frequency, decreasing frequency, increasing size, and decreasing size. Experimental results unanimously favored prioritizing functional groups based on the descending order of size, which means that, in case of an overlap, the functional group with the larger size is abstracted.

Representing Molecules as Sets of Histograms. Once domain information is incorporated into molecular graphs, a scalable transformation to convert each molecule into a set of representative histograms is adopted. As discussed in previous sections, scalability is the major issue in computing the *p*-value of a molecular substructure. The bottlenecks are the mining of frequent substructures at low frequencies and the exponential cost of subgraph isomorphism. To remove these bottlenecks, each molecule is converted into a set of histograms, where each histogram represents a substructure in the molecule. Owing to this transformation, the problem of computing *p*-value becomes tractable. We make the assumption that a low *p*-value in the histogram space corresponds to a low *p*-value in substructure space.

To convert each molecule into a set of histograms, we perform random walk with restarts (RWR) on each node of a molecule. The idea is to capture the distribution of *node−node pairs* in the neighborhood of each node in a molecule.

Definition 2: Node−Node Pair. *A node−node pair (NNP) is formed by two nodes sharing a bond; the bond order is also included in the description. For example, in carboxamide, there are three types of NNPs: C-1-C, C-1-N, and C-2-O, while in benzene, the NNP types are C-1-C and C-2-C.*

RWR simulates the trajectory of a random walker that starts from the target node and keeps jumping from one node to a neighbor. Each neighbor has an equal probability of becoming the new station of the walker. At each jump, we track the NNP traversed. At the same time, we do not want the walker to go too far away from the starting node since we just want to capture the neighborhood of the node, not the entire molecule. Thus, a restart probability α is employed to bring the walker back to the starting node. For instance, if we set $\alpha = 0.25$, in the average case, after every four jumps, the walker returns to the starting node.

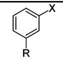
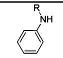
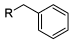
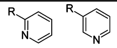
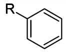
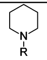
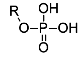
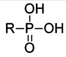
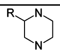
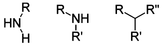
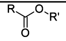
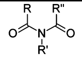
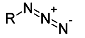
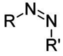
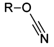
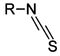
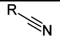
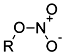
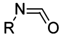
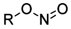
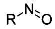
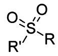
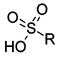
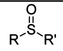
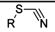
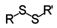
The RWR is iterated on each node until the distribution of NNP-types converges. As a result, RWR produces a continuous distribution for each node where the frequency of a NNP type lies in the range [0, 1]. We term this frequency as *NNP value*, and mathematically, the frequency of a particular NNP type *t* is equal to

$$\text{NNP value of } t = \frac{\text{number of times NNP type } t \text{ is visited}}{\text{number of jumps by the walker}} \tag{2}$$

To make the problem tractable, the NNP values are discretized into 10 bins. For example, a NNP value of 0.07 will be discretized as 1, and a value of 0.34 will be discretized as 3. RWR can therefore be visualized as placing a window at each node in the molecule and capturing a histogram representation of the substructure within it. As a result, a molecule of *m* nodes is represented by *m* histograms. An example is shown in Figure 6, which shows the RWR results on the molecular graph of carboxamide. Four histograms are produced since RWR is performed on each of the four nodes, and each histogram has a dimension of 3, since there are three types of NNPs.

While the approach is simple and effective to capture neighborhood information, the total number of NNP types can be fairly large due to its combinatorial nature, and as a result, the dimension in the histogram can be huge. So we ask the question, do we need to track all types of NNPs? It is common across organic data sets that a high percentage of the atoms are formed by a limited number of distinct atom-types. Moreover, some functional groups such as benzene are also very frequent. To substantiate this claim, we examined the distribution of node types in the reduced graph representation of molecules in the

Table 1. Table of Functional Groups

	Functional Group Name	Functional Group Formula	Structure
1.	Halo-benzene derivatives	RC_6H_4X	
2.	Aniline Derivatives	C_6H_5NR	
3.	Toluene Derivatives	$RCH_2C_6H_5$	
4.	Substituted Pyridines	RC_5H_4N	
5.	Substituted Benzenes	RC_6H_5	
6.	Piperidine Derivatives	$C_5H_{10}NR$	
7.	Phosphates	RPO_4	
8.	Phosphono Derivatives	$RPO(OH)_2$	
9.	Alcohol	ROH	R−OH
10.	Piperazine Derivatives	$C_4H_{10}N_2$	
11.	Amines	RNH_2, R_2NH, R_3N	
12.	Ester	RCO_2R'	
13.	Imide	$RC(=O)NR'(=O)R''$	
14.	Azide	RN_3	
15.	Azo Compounds	RN_2R'	
16.	Cyanates	$ROCN$	
17.	Isothiocyanates	$RNCS$	
18.	Nitriles	RCN	
19.	Nitrates	$RONO_2$	
20.	Isocyanates	$RNCO$	
21.	Nitrite	$RONO$	
22.	Nitroso Compounds	RNO	
23.	Sulfonyls	RSO_2R'	
24.	Sulfonic Acid Derivatives	RSO_3H	
25.	Sulfoxides	$RSOR'$	
26.	Thiols	RSH	
27.	Thiocyanates	$RSCN$	
28.	Disulfides	$RSSR'$	

NCI/NIH AIDS database (http://dtp.nci.nih.gov/). Recall, that these nodes can be either atoms or functional groups. The plot in Figure 7 shows the cumulative frequency of node types (after employing Joelib2 atom-typer) when sorted by their individual frequencies in decreasing order. As can be seen, even though there are 127 node types, 90% of the nodes are formed from the top 25 most-frequent types. The result inspires us to track only those NNP types that are between nodes within the top 25 most-frequent types. Otherwise, we only track the node type of the new node the walker reaches. For example, consider the probable situation where C is within the top 25 frequent node types and F is not. During random walk, if the walker jumps from a C atom to another C atom through a single bonded edge, then the count for C-1-C NNP would be updated. On the other

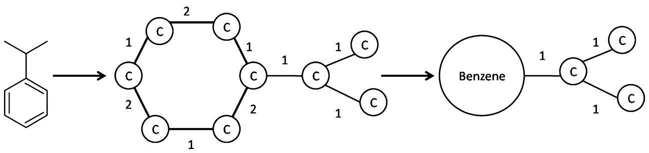
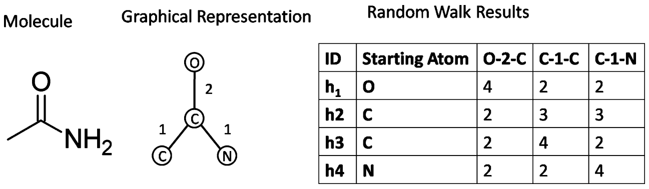


Figure 5. Example of abstracting out the benzene functional group from cumene.



ID	Starting Atom	O-2-C	C-1-C	C-1-N
h_1	O	4	2	2
h2	C	2	3	3
h3	C	2	4	2
h4	N	2	2	4

Figure 6. Random walk results on carboxamide.

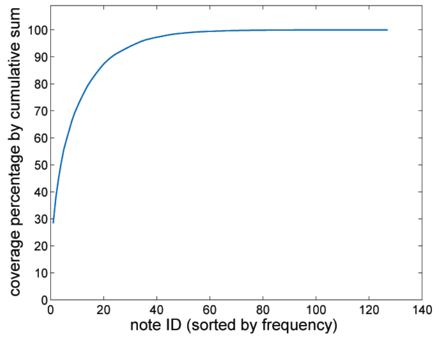


Figure 7. Node-type distribution in the AIDS database.

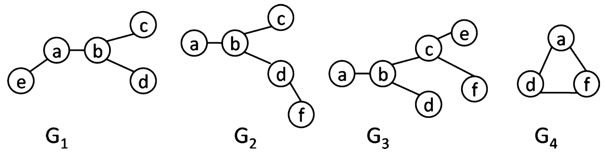


Figure 8. Sample graph database.

hand, if the walker jumps from a C atom to a F atom, the count for node-type F in the histogram would be updated since F is outside of the top 25. With this approach, the histogram consists of a subset of the NNP types, and only those node types that are not within the top 25 most-frequent types. This approach retains a considerable amount of information, while keeping the number of dimensions in the histogram manageable.

RWR inherently takes the proximity of NNP types into account and, therefore, preserves more structural information rather than simply counting the occurrence of NNP types in the neighborhood. For example, an NNP close to the starting node will be visited more often than an NNP further away. As a result, more structural information is preserved than a simple count. The property is reflected in Figure 6. The value for the O-2-C NNP is higher than the other two NNPs for node O since O-2-C is the closest NNP to O and is thus visited more often than the other two.

Probabilistic Framework. As shown in the previous section, RWR captures a histogram based representation of the substructures around each node. Next, we analyze the structural information that can be extracted from their histogram representation. For illustrative purposes, consider G_1-G_4 in Figure 8 as a sample graph database. For simplicity, edges are unlabeled and all NNP types are tracked. Table 2 contains the histograms produced from nodes labeled "a" in each graph at a restart probability of 0.25. As can be seen, only the NNP-types a−b, b−c, and b−d have nonzero values across G_1, G_2, G_3. This result indicates that there could be a subgraph formed by the common nonzero NNP types, which is true in this particular case as shown in Figure 9. On the other hand, no NNP type has a nonzero value across G_1-G_4, since there is no common subgraph among them. The common nonzero NNP types can be computed by taking the *floor* of the histograms.

Table 2. RWR Histograms of Nodes Labeled "a" in G_1-G_4

histogram	a−b	a−d	a−e	a−f	b−c	b−d	c−e	c−f	d−f
G_1	2	0	3	0	1	1	0	0	0
G_2	4	0	0	0	2	1	0	0	1
G_3	3	0	0	0	1	2	1	1	0
G_4	0	3	0	3	0	0	0	0	2

Definition 3: Floor. The floor of a set of histograms { $h_{\underline{1}}$, ..., $h_{\underline{n}}$ } *is a histogram* $h_{\underline{f}}$ *where,* $h_{f_i} = \min(h_{1_i}, ..., h_{n_i})$ *for* $i =$ *1 ... n.* Example: floor([2, 0, 3, 0], [2, 0, 4, 1], [1, 1, 4, 5]) = [1, 0, 3, 0].

Under this setting, if the *p*-value of the histogram produced from taking the floor can be computed to verify its significance, we will be effectively verifying the significance of substructures. We therefore concentrate on developing a probabilistic framework to compute the *p*-value of a histogram.

Recall that to compute the *p*-value of a histogram, we need to calculate its probability density function (pdf) and then calculate the area to the right of the observed frequency under the pdf. The first step in generating the pdf is to compute the probability of occurrence of a histogram. For this task, a probability matrix is computed. The matrix contains the prior probabilities of each feature component (in our case, NNP and node types).

An example is shown in Table 3. The probabilities are generated from Figure 6 under the assumption that carboxamide is the only molecule in the database and all NNP types are tracked. In a typical setting however, the database contains all molecules tested against an assay and the histogram tracks a subset of the NNP and node types. Each row contains the prior probabilities of a feature, and the *x*th column in a feature row represents the probability of finding that feature at least *x* number of times. For example, column 3 in row "C-1-C" contains the probability of *P*(C-1-C ≥ 3) and column 1 in row "O-2-C" contains the probability of *P*(O-2-C ≥ 1). The prior probabilities are computed empirically by examining the RWR histograms. For this particular example in Table 3, C-1-C has a value higher than 3 in two out the four histograms (h_2 and h_3 in Figure 6), and thus, the value is 2/4.

The probability of finding a substructure in a molecule is modeled in the histogram space. We need to calculate the probability of finding a (sub)histogram in another histogram. We therefore first define the notion of subhistograms.

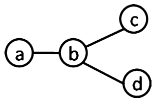


Figure 9. Common subgraph in G_1, G_2, G_3.

Table 3. Prior Probability Matrix

NNPs	1	2	3	4	5
O-2-C	4/4	4/4	1/4	1/4	0
C-1-C	4/4	4/4	2/4	1/4	0
C-1-N	4/4	4/4	1/4	1/4	0

Definition 4: Subhistogram. For two histograms $\underline{x} = [x_1, ..., x_n]$ and $\underline{y} = [y_1, ..., y_n]$, $\underline{x}$ is a subhistogram of $\underline{y}$ if and only if $x_i \le y_i$ for $i = 1, ..., n$. The relation is denoted as $\underline{x} \subseteq \underline{y}$. At the same time, $\underline{y}$ is a superhistogram vector of $\underline{x}$. Example: [2, 0, 3, 0] ⊆ [2, 0, 4, 1] whereas [2, 0, 3, 0] is not ⊆ [1, 1, 4, 5].

The probability of histogram $\underline{x} = [x_1, ..., x_n]$ occurring in a random histogram $\underline{y} = [y_1, ..., y_n]$ can be expressed as a joint probability

$$P(\underline{x}) = P(y_1 \ge x_1, ..., y_n \ge x_n) \tag{3}$$

$$= \prod_{i=1}^{n} P(y_i \ge x_i) \tag{4}$$

Each event in eq 4 is the probability of a feature (NNP or node type) in the random histogram $\underline{y}$ having a value equal or higher than the corresponding value in $\underline{x}$. The computation in eq 4 is facilitated by the prior probability matrix. An example is shown below for histogram h_1 in Figure 6.

Example:

$$\begin{aligned} P([4,2,2]) &= P(\text{O-2-C} \ge 4) \times P(\text{C-1-C} \ge 2) \times P(\text{C-1-N} \ge 2) \\ &= \frac{1}{4} \times \frac{4}{4} \times \frac{4}{4} \\ &= \frac{1}{4} \end{aligned}$$

***p*-Value of $\underline{x}$.** For a given histogram $\underline{x}$, if $P(\underline{x})$ is known, then its frequency in a database of random histograms can be modeled as a binomial distribution. A random histogram can be viewed as a trial and $\underline{x}$ occurring in the histogram a "success". A database consisting of m histograms will involve m trials for $\underline{x}$. The frequency of $\underline{x}$ in the database is the number of successes. Therefore, the probability of $\underline{x}$ having a frequency μ is

$$P(\underline{x};\mu) = \binom{m}{\mu} P(\underline{x})^{\mu} (1 - P(\underline{x}))^{m-\mu} \tag{5}$$

The pdf of $\underline{x}$ can be generated from eq 5 by varying μ in the range $[0, m]$. Therefore, given an observed frequency μ_0 of $\underline{x}$, its p-value can be calculated by measuring the area under the pdf in the range $[\mu_0, m]$, which is

$$p\text{-value}(x, \mu_0) = \sum_{i=\mu_0}^{m} P(\underline{x};i) \tag{6}$$

Equation 6 reduces to the regularized Beta function $I(P(\underline{x}); \mu_0, m)$.[41] When both $mP(\underline{x})$ and $m(1 - P(\underline{x}))$ are large, the binomial distribution can be approximated using a normal distribution.

Mining Significant Histograms. With the conversion of molecules into histograms, and a probabilistic model to evaluate significance of a substructure in the histogram space, we explore how the histograms can be mined to extract the significant regions. We use the significant histogram mining algorithm proposed in GraphRank[42] to mine all significant subhistograms. The algorithm takes the probabilistic model, a p-value threshold, and a frequency threshold as input. It returns all significant subhistograms with p-values below the specified threshold. GraphRank explores the space of all subhistograms in a bottom-up, depth-first manner. The search states can be viewed as a rooted-tree where, the root represents the floor of all histograms in the database. At each of the search states along a path, GraphRank evaluates a superhistogram of the parent of the current state for significance. New states are generated until the entire space of subhistograms is explored. The method has been proven to be correct, complete, and duplicate-free. GraphRank has been applied to rank patterns for significance in chemical and web data.

Each subhistogram mined by GraphRank could potentially represent a significant substructure and could be used for various applications. In this paper, we highlight two of its applications. We develop an efficient classifier based on significant histograms. Second, we also develop a tool that maps the significant histograms to the actual substructures they represent. The significantly over-represented substructures open up the opportunity to multiple areas such as scaffold hopping and molecular diversity analysis.

Molecule Classification. To demonstrate the potential of significant patterns, we attempt to classify molecules by using the discovered patterns as structural keys. Figure 10 utlines our approach. First, the set of significant histograms are mined from each of the classes in the data set. The prior probability matrix for each class is generated by examining the distribution of NNP and node types in all but the selected class. This strategy maximizes the discriminative potential of the mined significant histograms. Next, each molecule is converted to a binary vector that captures the presence or absence of all mined significant patterns. The ith index in a binary vector is 1 if the ith pattern is present in the molecule.

Since we simulate the molecules and patterns in the histogram space, we first develop an algorithm (algorithm 1) to construct the binary vector representation of a given molecule. For each node in the reduced molecular graph of

Algorithm 1 Construction of the chemical descriptors $(\mathbb{S}, m)$

```
Require: $\mathbb{S}$ is the set of significant histograms mined from all classes
Require: $m$ is query molecule
Ensure: V is the binary vector representation of m
$\underline{V} \leftarrow$ vector of size $|\mathbb{S}|$ intialized to 0
for each node $a \in m$ do
    $i \leftarrow 0$
    while $i < |\mathbb{S}|$ do
        $\underline{h} \leftarrow \mathbb{S}[i]$
        if $\underline{h} \subseteq RWR\_hist(a)$ then
            $V[i] \leftarrow 1$
        $i \leftarrow i + 1$
return $\underline{V}$
```

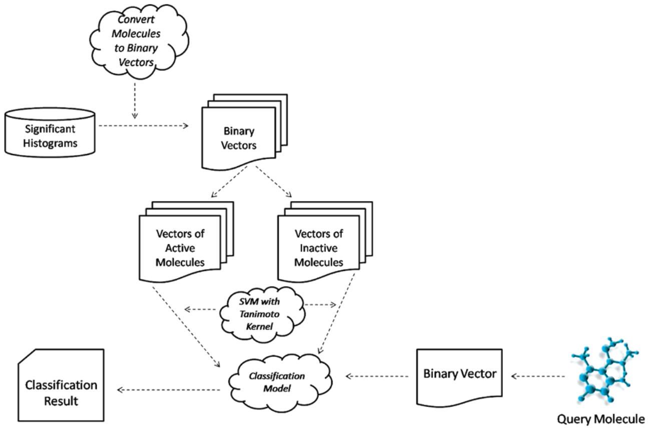


Figure 10. Outline of an approach that employs significant histograms for classification.

the query, containment of a significant histogram is checked (lines 2−5). Essentially, in this step we are comparing the substructure around each node in the query molecule to the mined significant substructures using the histogram representations. Therefore, any significant histogram S_i that is not a subhistogram of a node is discarded, since we are looking for containment of a substructure. Ultimately, the binary feature vector is computed and returned (line 9).

Given the vector representation of a set of molecules, we use support vector machines (SVM)[43] to develop the training model. One key issue that affects the performance of SVM is the choice of kernel employed to measure similarity between vectors. Theoretically, any kernel can be used as long as the similarity matrix computed by the kernel function satisfies the Mercer's conditions. These conditions guarantee that the matrix is symmetric and positive semidefinite.

Definition 5: Positive Semidefinite Matrix. A symmetric matrix M is positive semidefinite if $x'Mx \geq 0$ for any nonzero column vector x, where x′ represents the transpose of x.

We use the Tanimoto coefficient function to define our kernel. The Tanimoto coefficient has been extensively used in the chemoinformatics community and has been shown to be effective in measuring similarity between chemical compounds.[20,44−46] Moreover, the Tanimoto kernel satisfies Mercer's conditions.[47] The Tanimoto kernel function between two vectors X and Y is defined by

$$\kappa_{\text{tm}}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{7}$$

Mapping Significant Histograms to Molecular Substructures. While we demonstrated one application of significant histograms in the last section, it is natural for a chemist to ask "What are the actual substructures these histograms represent?" In this section, we discuss how we answer the question.[31] A schematic outline of the process is provided in Figure 11.

In our approach, the mined significant histograms are employed to locate regions that are likely to embed the corresponding significant substructures. Recall that RWR is performed on each node of the reduced molecular graphs to create a database of histograms. The significant histograms are then mined from the histogram database. For each significant histogram, we first identify the node type it originated from and scan all nodes of the same type in database. For each such node, the RWR histogram produced from it is fetched and compared to the significant histogram. For example, a significant histogram originating from a C node is compared with all RWR histograms of C nodes. In the comparison, it is checked whether the significant histogram is a subhistogram of the RWR histogram. If the property holds, then it guarantees that all NNP and node types that form the significant histogram are contained in the region described by the RWR histogram, and therefore, it is likely that the significant substructure is embedded in the region around the current node. Following this reasoning, interesting regions from chemical compounds are obtained for each significant histogram for further analysis. Mathematically, for a significant histogram S_i, we are compiling a set of nodes where

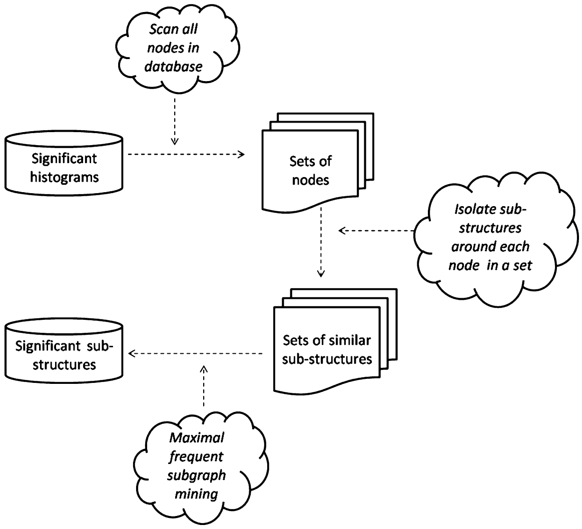

Scan all nodes in database
Significant histograms
Sets of nodes
Isolate sub-structures around each node in a set
Significant sub-structures
Sets of similar sub-structures
Maximal frequent subgraph mining

Figure 11. Outline of the method for obtaining significant substructures from significant histograms.

$$\mathbb{A} = \{a | \text{type}(a) = \text{type}(S_i) \text{ and } S_i \subseteq \text{RWR\_histogram}(a)\} \tag{8}$$

Once the set $\mathbb{A}$ is compiled for a particular significant histogram, the substructure centered on each node in the set within a user-specified radius is extracted. This mechanism produces a set of substructures for each significant histogram. The cutoff radius can be selected based on some prior knowledge about the typical size of substructures that one wants to study. In the worst case, one can select the entire molecule where the node occurs.

Since each set of the substructures in a set are extracted from regions containing the same significant histogram, the set is likely to embed a common substructure. This unique formulation allows us to employ any of the *maximal* frequent subgraph mining tools[26,28,48,49] with a high frequency threshold to identify the common substructure the set contains. The frequency threshold can be something in the range [80%, 100%]. A frequent subgraph is maximal if it is not a subgraph of any other frequent subgraph. In our case, we are looking for a particular significant substructure which is highly frequent in the set. Clearly, this reduces to finding the maximal subgraph as all its subgraphs are going to be frequent as well. As evident in the experimental section later, the largest substructure often forms the core part of a molecule. It is also important to note that the high frequency threshold allows us to mine significant substructures in an

Table 4. Anticancer Screen Data Sets

name	size	number of actives	description
MCF-7	28972	1989	breast
MOLT-4	41810	3391	leukemia
NCI-H23	42164	2235	nonsmall cell lung
OVCAR-8	42386	2255	ovarian
P388	46440	2549	leukemia
PC-3	28679	1692	prostate
SF-295	40350	1936	central nervous system
SN12C	41855	2123	renal
SW-620	42405	2623	colon
UACC-257	41864	1807	melanoma
yeast	83933	10257	yeast anticancer

extremely scalable manner, which is crucial in eliminating the scalability bottleneck. This last step also prunes out false positives where dissimilar substructures are grouped into the same set due to a similar histogram representation.

RESULTS

Classification Performance. We use 11 anticancer screen data sets available at PubChem[50] for comparison. PubChem is a well-maintained compilation of the biological activities of various molecules, containing the bioassay records for anticancer screen data sets against various cancer cell lines. Each data set contains molecules tested against a cancer cell line and its outcome *active* or *inactive*. We selected 11 such molecular data sets from the screen tests. Table 4 contains a brief summary of each of these NCI bioassays.

In the following experiments, classification is performed using the proposed molecular descriptor and compared to four other approaches:

1. Subgraph Mining-Based Molecular Classification. We chose one of the most recent subgraph mining techniques—scalable leap search (LEAP[14])—for comparison. Given an active and inactive data set, LEAP mines discriminative substructures, such that their frequencies in the active and inactive data sets are significantly different. A different method with a similar intuition has been tried before by Auer et al.[51] The discriminative power of a subgraph pattern in LEAP is quantified using a *G*-test score. Following this procedure, subgraphs are mined until all molecules in the training set contain at least one of the mined patterns. Finally, based on the presence or absence of the mined patterns, molecules are converted to a vector.

As can be seen, LEAP employs a more sophisticated approach to mine substructures than frequent subgraph/substructure mining techniques. Moreover, with a frequent substructure mining approach, the number of substructures mined is extremely large unless the frequency threshold is set really high. Therefore, a postprocessing step is required to filter and select the best patterns among the frequent ones.[22]

2. Graph Kernel-Based Method. For the graph kernel based method, we chose the state-of-the-art graph kernel—optimal assignment kernel (OA[7]). OA attempts to make an optimal assignment of atoms from one molecule to the others based on information derived from atom characteristics, neighborhood, and membership to certain structural elements. On the basis of the assignment, a similarity score is calculated between two molecules.

Table 5. Parameter Values for 5-Fold Cross-Validation

parameter	description	value
α	restart probability in random walk	0.1
μ	*p*-value threshold	0.1
δ	convergence threshold for RWR	0.005

3. Structural Keys. For structural keys, we use the Joelib2 computational chemistry library[23] to generate the bit vector representation of a molecule. Joelib generates a 54-dimensional vector where each bit corresponds to a structural key. Structural keys range from certain functional groups to the presence of specific bond types.

4. Fingerprints. For fingerprints, we use Daylight,[19] one of the most popular fingerprinting techniques in the chemoinformatics community. We use the default parameters to generate a 2048 dimensional feature vector where all paths of length 7 in a molecule are searched exhaustively and hashed to create the bit vector.

To make a uniform comparison, we use the same classification algorithm of SVM (LIBSVM[52]) for each of the above methods. The classification accuracy is evaluated by performing 5-fold cross-validation where the data set is divided into five subsamples. Four of the subsamples are merged to form the training set while the remaining sample is retained as the testing set for validation. Significant patterns are mined from the training set to construct the bit vector representation of molecules in both the training and testing set. Next, vectors in the training set along with their class labels are analyzed to learn the classification model using SVM. Finally, class labels are assigned to the molecules in the testing set. The cross-validation process is repeated five times using each of the subsamples exactly once as the testing set. Moreover, the cross-validation experiment is repeated five times with five randomly sampled data sets to measure consistency in performance.

To construct a balanced data set for evaluation, we sample 750 active molecules and an equal number of inactive molecules. First, the reduced graph representation of the entire set is computed. Since the graph reduction is independent of the cross-validation framework, it is performed only once. Next, 5-fold cross-validation is performed and the same set of parameter values mentioned in Table 5 is used for both the training and testing sets (μ is required only for training). To ensure convergence of RWR, the iteration for a node continues until the distance between two successive histograms is less than δ. As shown in the experimental results later, the OA kernel is unable to scale to large data sets and fails to complete within a reasonable time span. To overcome this problem, we run OA on a balanced data set containing 30% of the molecules in the original set.

The classification quality is measured by computing the area under the receiver operator characteristics (ROC) curve (AUC) for each of the methods. ROC curve is a graphical plot of the true positive rate against false positive rate for a classifier as the discrimination threshold is varied. The area under this curve (AUC) is a measure of the classifier accuracy. The area is bounded within the range of [0, 1], and a perfect model will have an area of 1. Since there are only two classes in each of the cancer data sets, a random classifier is expected to achieve a AUC of 0.5.

Table 6. AUC Comparison between OA, LEAP, Joelib, Daylight, and GraphSig

data set	OA kernel	LEAP	Joelib	Daylight	GraphSig
MCF-7	0.68 ± 0.12	0.76 ± 0.04	0.87 ± 0.02	**0.90 ± 0.01**	**0.90 ± 0.01**
MOLT-4	0.65 ± 0.06	0.72 ± 0.06	0.88 ± 0.02	**0.90 ± 0.02**	0.89 ± 0.02
NCI-H23	0.79 ± 0.08	0.79 ± 0.05	0.89 ± 0.02	**0.93 ± 0.01**	**0.93 ± 0.02**
OVCAR-8	0.67 ± 0.04	0.78 ± 0.02	0.88 ± 0.02	**0.93 ± 0.01**	**0.93 ± 0.02**
P388	0.79 ± 0.07	0.84 ± 0.03	0.89 ± 0.03	**0.93 ± 0.01**	0.91 ± 0.02
PC-3	0.66 ± 0.09	0.76 ± 0.04	0.90 ± 0.02	**0.93 ± 0.01**	**0.93 ± 0.02**
SF-295	0.75 ± 0.11	0.77 ± 0.02	0.90 ± 0.02	0.93 ± 0.01	**0.94 ± 0.02**
SN12C	0.75 ± 0.08	0.80 ± 0.02	0.89 ± 0.02	**0.93 ± 0.01**	**0.93 ± 0.02**
SW-620	0.70 ± 0.02	0.76 ± 0.04	0.88 ± 0.02	**0.93 ± 0.01**	**0.93 ± 0.03**
UACC-257	0.65 ± 0.05	0.75 ± 0.03	0.89 ± 0.03	**0.93 ± 0.01**	**0.93 ± 0.02**
yeast	0.64 ± 0.04	0.71 ± 0.02	0.82 ± 0.03	**0.85 ± 0.02**	0.82 ± 0.03
average	0.702 ± 0.07	0.767 ± 0.03	0.880 ± 0.02	**0.917 ± 0.01**	0.913 ± 0.02

Table 6 shows the AUC for OA, LEAP, Joelib, Daylight, and GraphSig. The best result for each data set is highlighted in bold. Clearly, GraphSig outperforms OA, LEAP, and Joelib. On average, GraphSig achieves an AUC 0.211, 0.146, and 0.033 higher than OA, LEAP, and Joelib, respectively. The average AUC of Daylight and GraphSig are almost identical. As can be seen, they perform equally well on 7 of the 11 data sets. Daylight displays superior performance in three data sets, whereas GraphSig performs better in one.

Table 7. BEDROC Comparison between Daylight and GraphSig

data set	Daylight	GraphSig
MCF-7	0.41	**0.61**
MOLT-4	0.42	**0.45**
NCI-H23	0.44	**0.63**
OVCAR-8	0.40	**0.65**
P388	0.50	**0.55**
PC-3	0.33	**0.62**
SF-295	0.32	**0.63**
SN12C	0.40	**0.62**
SW-620	0.36	**0.60**
UACC-257	0.34	**0.65**
yeast	**0.38**	**0.38**
average	0.39	**0.57**

Table 8. Running Time Comparison between OA, LEAP, Joelib, Daylight, and GraphSig (seconds)

data set	OA kernel	OA kernel(3X)	LEAP	Joelib	Daylight	GraphSig
MCF-7	405	9800	772	79	72	121
MOLT-4	315	10177	309	134	110	253
NCI-H23	280	7800	305	82	80	158
OVCAR-8	220	7980	411	95	86	154
P388	210	7670	108	76	92	124
PC-3	145	5600	234	85	75	149
SF-295	242	9780	488	95	82	171
SN12C	187	8780	1004	78	83	170
SW-620	345	9980	375	109	95	211
UACC-257	176	7560	249	84	90	143
yeast	798	7689	84	83	101	159
average	302.1	8437.8	394.4	90.9	87.8	164.8

Clearly, the performance of Daylight and GraphSig on molecular classification is too close to determine the superior of the two. Therefore, we further evaluate the performance using the Boltzmann-enhanced discrimination of receiver operator characteristics (BEDROC)[53] metric to gain a deeper understanding. BEDROC was developed to answer the shortcomings of ROC which is not sensitive to early recognition of actives. Like AUC, BEDROC is bounded by [0,1] as well. However, BEDROC incorporates the importance of the "early recognition problem" in virtual screening by better rewarding the early ranking of actives. The BEDROC metric is defined as follows:

$$\frac{\sum_{i=1}^{n} \mathrm{e}^{-\alpha r_i/N}}{\frac{n}{N}\left(\frac{1-\mathrm{e}^{-\alpha}}{\mathrm{e}^{\alpha/N}-1}\right)} \frac{R_{\mathrm{a}} \sinh(\alpha/2)}{\cosh(\alpha/2) - \cosh(\alpha/2 - \alpha R_{\mathrm{a}})} + \frac{1}{1-\mathrm{e}^{\alpha(1-R_{\mathrm{a}})}} \quad (9)$$

where, r_i is the rank of the ith active in the ranked list, N is the number of molecules in the testing set, n is the number of actives in the testing set, R_{a} is the ratio of actives in the testing set, and α is a weighing parameter to decide the importance of the early part of the ordered ranked list. For example, a $\alpha = 20$ means that the first 8% of the ranked list contributes 80% of the BEDROC score. A classifier which guesses class labels randomly will achieve a score of

$$B_{\mathrm{r}} = \frac{1}{\alpha} + \frac{1}{1-\mathrm{e}^{\alpha}} \quad \text{if } \alpha R_{\mathrm{a}} \ll 1 \quad (10)$$

As can be seen, there is a condition attached to the metric since BEDROC is sensitive to the "saturation effect". More specifically, the saturation effect can significantly influence the BEDROC score for the same "constant true performance" when the ratio of actives vary in the testing sets.[53] Therefore, to avoid the saturation effect, the condition needs to be enforced. Accordingly, we reduce the ratio of actives to 0.01 and set α to 20.0 to compute the BEDROC score.

Table 7 shows the BEDROC scores for Daylight and GraphSig. As a reference point, a random classifier would achieve a BEDROC score of 0.05. As can be seen, GraphSig performs better than Daylight in 10 out of the 11 cancer data sets. Both Daylight and GraphSig achieve an average score considerably higher than 0.05. GraphSig achieves an average BEDROC of 0.57 compared to an average score of 0.39 for Daylight. This result indicates that GraphSig is a better descriptor than Daylight for molecular classification.

Table 8 shows the running times of OA, LEAP, Joelib, Daylight, and GraphSig. We also show the running time of OA on the entire training set (OA(3X)) to demonstrate that it is not scalable to large data sets. The running times of LEAP, GraphSig, Joelib, and Daylight are measured as the time to compute the vector representation of all molecules in the training set. For OA, the time to compute the kernel

Table 9. Precision/Recall Achieved on MDDR Data Set by Daylight and GraphSig

	Daylight		GraphSig	
class	precision	recall	precision	recall
AC	0.94	**0.98**	**0.97**	0.97
AT	0.98	**0.97**	**0.99**	0.94
PR	**0.94**	**0.92**	**0.94**	**0.92**
RT	**0.94**	0.93	0.92	**0.94**
average	0.95	**0.95**	**0.96**	0.94

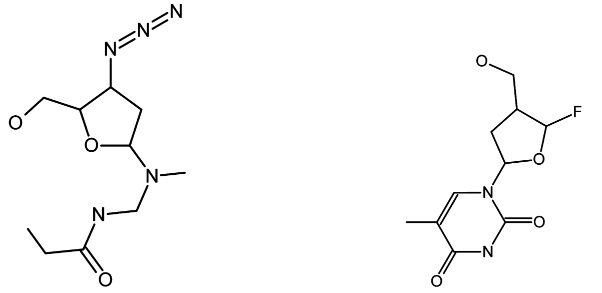
is treated as its running time. All running times are averaged over 5-fold, except for OA(3X). Since OA(3X) takes too long to finish computing over all 5-fold, we report the time for only one-fold. Overall, GraphSig performs 2.4 times faster than LEAP and 51 times faster than OA(3X), while achieving significantly better quality than both of them. Joelib and Daylight are 1.8 and 1.9 times faster than GraphSig because neither of them involves any mining process in computing the chemical descriptors.

To further investigate the adaptability of our method, we attempt to classify a data set with multiple classes. For this purpose, we evaluate GraphSig on the MDDR (MDL Drug Data Report)[54] data set. The MDDR data set is compiled from patent literature and contains 178 418 molecules. Each molecule is tagged with a five-digit activity code (e.g., "31000") and a brief description. Using these tags, 4865 molecules are isolated belonging from four groups: ACE inhibitor (AC), angiotensin II receptor (AT), protease inhibitor (PR), and reverse transcriptase inhibitor (RT). From each of these groups, 500 molecules are selected randomly to prepare a uniform training data set of 2000 molecules. We perform 5-fold cross-validation on this data set to evaluate GraphSig.

For benchmarking purposes, we choose Daylight since clearly it is the closest competitor in terms of performance. Table 9 compares the precision and recall rates for each of the classes. We use precision and recall for measuring the classification performance since ROC curves assume a binary setting for classification. A classifier which guesses randomly will achieve precision and recall rates of 0.25 since there are four different classes.

As can be seen, both the descriptors perform very well when compared to an average precision and recall rate of 0.25. When compared to Daylight, GraphSig achieves at least as good or better precision rates in three classes and the recall rates are better or equally good in two of the classes. Precision and recall rates however are not sensitive to early detection of actives like in BEDROC. Consistent with the AUC metric, the differences in the performance of the two descriptors are small. In the next section however, we highlight a fundamental difference between Daylight and GraphSig that makes GraphSig suitable for studying molecular databases from a different perspective.

Quality Evaluation. While fingerprints pack a large number of structural features into a fixed width bit vector, the features are exhaustive in nature and statistically less interesting. On the other hand, GraphSig employs a more sophisticated pattern selection process. The patterns have higher information content and are capable of doing more than just classification. To emphasize this property, we present a subset of the over-represented substructures retrieved from the molecular data sets to demonstrate the utility of GraphSig.



(a) Sub-structure of AZT

(b) Sub-structure in FDT class of compounds

Figure 12. Some significant substructures in molecules active against AIDS.

The substructures are retrieved using the mechanism discussed earlier. We set the cutoff radius to 12 and use FSG[28] to mine maximal frequent subgraphs with a frequency threshold of 80%. To make the quality assessment more focused, we separate the set of compounds medically active against a disease and run our algorithm on it to retrieve the significant substructures. Besides the cancer and MDDR data sets, we also mine the DTP-AIDS antiviral screen chemical compound data set from NCI/NIH (http://dtp.nci.nih.gov/). The AIDS data set consists of 43 905 classified chemical molecules, and a total of 1.09 million atoms. On average, each molecule contains 25.4 atoms (vertices) and 27.3 bonds (edges). There are 58 distinct atoms in total, although a majority of them are C, O, and N.

Substructures from AIDS Data Set. Figure 12 shows two significant substructures mined by GraphSig in the active set of compounds against AIDS. The retrieved structure in Figure 12a is a substructure of one of the most potent classes of drugs against AIDS, namely azido pyrimidines.[55] The compound 3′-azido-3′-deoxythymidine (AZT; NSC 602670), which is the structurally closest active molecule to the structure mined, has one more carbon and oxygen atoms and three more bonds. AZT is currently the most widely adopted medicine to control the HIV virus.

The structure in Figure 12b is a substructure of a class of medically active compounds called 3′-fluoro-3′-deoxy-thymidine (FDT).[56] FDT is a fluorinated analog of AZT. It is more active against AIDS than AZT; however, it also displays a higher level of toxicity. The substructure retrieved is the core structure in the FDT class of compounds, where it binds with a halogen to form the actual compound.

Both AZT and FDT can be mined using any of the frequent substructure mining techniques as well. However, the structures will remain buried among all the other frequent substructures that will be mined along with them. For example, the frequency of AZT is 12% in the data set of molecules active against AIDS. If all substructures with a frequency of at least 12% are mined, then around 14 000 substructures will be retrieved along with AZT rendering the technique as highly ineffective.

Substructures from Cancer Data Sets. We also analyze our methods in the anticancer screen data sets against leukemia (MOLT-4) and melanoma (UACC-257). Figure 13 is a significant substructure mined from the set of molecules

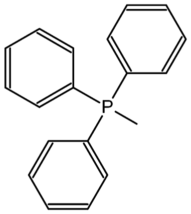
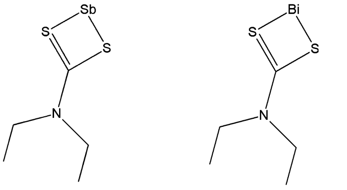


Figure 13. Core substructure in the class of phosphonium salts



(a) Sub-structure made of Sb

(b) Sub-structure made of Bi

Figure 14. Some significant substructures in molecules medically active against leukemia.

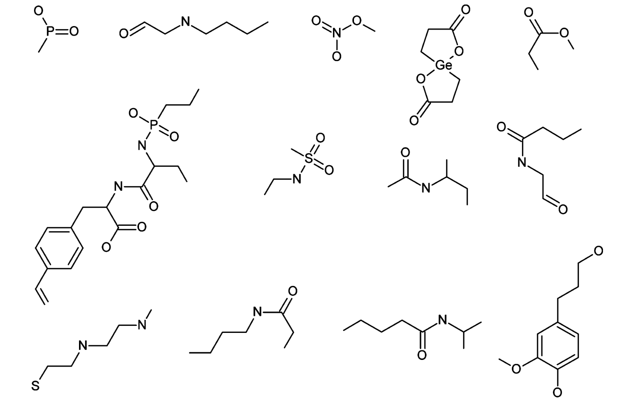


Figure 15. Significant substructures mined from ACE inhibitors in the MDDR database.

active against melanoma. The structure, methyl-triphenylphosphonium, is the core structure of a class of phosphonium salts where, the binding occurs on the single free carbon attached to phosphorus. It displays cytotoxic behavior against a number of cancer cell lines.[57]

Figure 14a and b shows two significant substructures mined from molecules active against leukemia (MOLT-4). It is interesting to note that the only difference in the structures is the presence of antimony (Sb) and bismuth (Bi). Incidentally, both antimony and bismuth are part of the same group of metals in the periodic table. What perhaps is most interesting about the structures is their frequency. The frequencies of both the substructures are below 0.5%, and none of the current frequent subgraph mining techniques can scale to such a low frequency and capture their significance. Moreover, even if the mining process is able to complete within a reasonable time span, millions of frequent but nonsignificant substructures will be included in the answer set nullifying the exclusivity of significance.

Figures 15−18 contain a subset of the significantly over-represented molecular fragments mined from the AC, AT, PR, and RT class of molecules in the MDDR data set. The results are consistent with the literature. For example, in the RT class, our technique is able to identify the functional groups which characterizes the molecules in this group such as 4-azido-5-(hydroxymethyl), pyrimidines, and cyclopropyles. The results establish the utility of our technique in predicting structure−activity relationships.

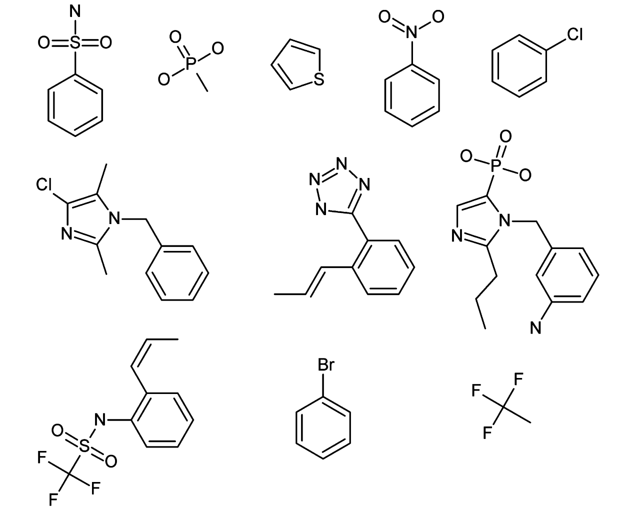


Figure 16. Significant substructures mined from angiotensin II receptors in the MDDR database.

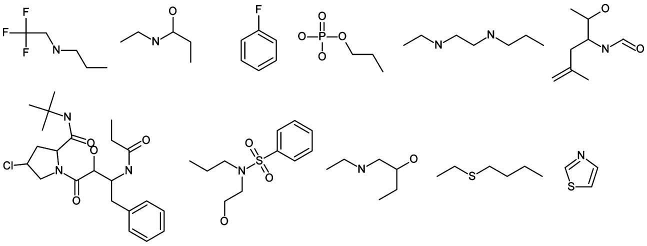


Figure 17. Significant substructures mined from protease inhibitors in the MDDR database.

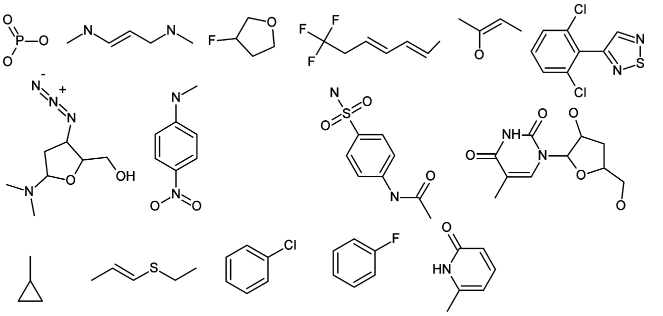


Figure 18. Significant substructures mined from reverse transcriptase inhibitors in the MDDR database.

The above results highlight the practical usefulness of our technique in a number of areas. GraphSig successfully retrieved the core structures of molecules that are medically active against cancer or AIDS, thereby establishing a link between significance and medicinal values of the structures. Further, the technique is able to identify functional groups that characterize a given class of molecules in the MDDR database. Our technique also displays potential in providing leads to design new drugs by learning statistical properties of drugs already discovered. Moreover, GraphSig successfully overcomes the scalability bottleneck of mining significant substructures at low frequencies.

Scalability. In this section, we analyze the scalability of our method and compare it to three well-known frequent subgraph mining techniques, FSG,[28] gSpan,[29] and Gaston.[30]

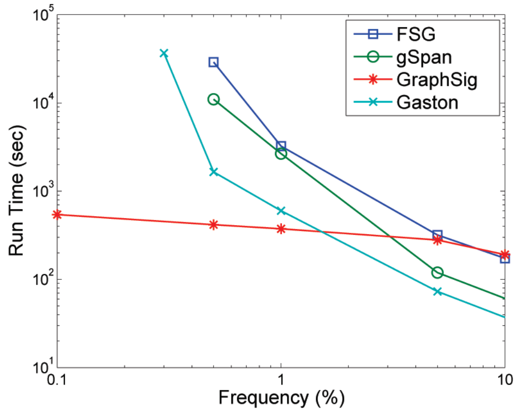


Figure 19. Time vs frequency comparison of the mining techniques.

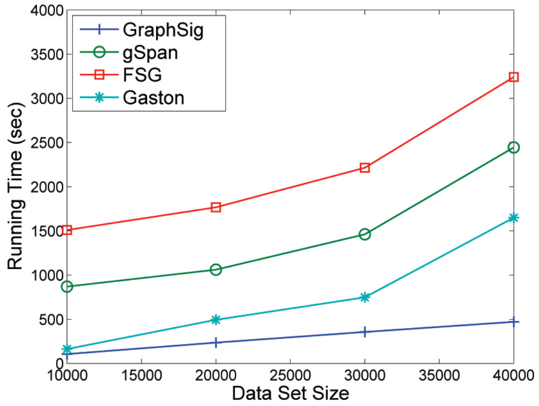


Figure 20. Time vs data set size comparison of the mining techniques.

Scalability is evaluated against frequency and data set size. We use the AIDS data set for benchmarking purposes.

One of the primary motivations in developing GraphSig is to overcome the bottleneck of mining patterns at low frequency thresholds. We therefore examine the running time growth rate of GraphSig against frequency and compare it to FSG, gSpan, and Gaston. As can be seen in Figure 19, the execution times of the three subgraph mining techniques grow exponentially whereas GraphSig grows linearly. The frequency threshold is varied between 0.1% and 10%. The times on the 0.1% frequency threshold for FSG, gSpan, and Gaston are not reported since they fail to complete even after 12 h. The running time of GraphSig is lower-bounded by the time to perform RWR on all nodes in the database. As a result, the frequent subgraph mining techniques perform better than GraphSig at higher frequencies. However, at lower frequencies, the computation cost of performing RWR is minimal compared to the total running time; this results in GraphSig achieving a superior performance.

Figure 20 demonstrates the growth rate of running time against data set size. The data set size is varied between 10 000 and 40 000 by randomly drawing graphs from the AIDS data set. For GraphSig, we use a *p*-value and frequency threshold of 0.1%. Due to the enormous running times of FSG and gSpan at a frequency threshold of 0.1%, we set their frequency threshold to 1%. Gaston is comparatively faster, and to make the comparison as fair as possible, we set the frequency to 0.5% for Gaston. As can be seen, even under this biased setting, GraphSig displays superior performance and grows linearly.

DISCUSSION

In this paper, we studied an important problem of mining significant molecular substructures from large molecular repositories. We developed a novel graph mining technique that converts a molecule into a graph and thereafter mines significant molecular substructures from the graph database. Our proposed approach focused on speeding up the computation process so that it is practically viable to identify the significantly over-represented substructures.

While mining frequent patterns has been an active research area in the past decade,[26,28−30,48,58] current techniques fail since they are unable to scale to low frequencies. At low frequencies, the number of candidate subgraphs grows exponentially due to the inevitable combinatorial explosion. Since it is computationally infeasible to evaluate significance of all candidates, it is crucial that a more informed choice is made to select candidates. The contribution of our work lies on this front where we achieve an enormous speed-up on the computation process by performing frequent subgraph mining only on regions that are likely to contain a significantly over-represented structure.

With our approach, we save time on two accounts. First, due to analysis of significance in the histogram space, we avoid the need to generate a large number of arbitrary molecular databases and compute the frequency distribution of a query molecule. Second, due to grouping of similar substructures into sets, frequent subgraph mining can be performed on each set with a frequency threshold that is able to mine significant patterns scalably.

At the core of our work is the analysis of molecules using a histogram representation. It is natural for one to compare our proposed histogram representation to fingerprints. The problem of evaluating the significance of a fingerprint can be easily solved using the probabilistic model we used. However, the limiting factor of the fingerprint-based approach is its inability to capture local patterns. Fingerprints provide a global representation of the features present in a molecule. This representation does not preserve any information on the relative distances between the features. Because our histogram representations are generated from RWR, the observed features are expected to be within a certain radius where the radius is guided by the restart probability used for RWR. Moreover, with our approach, the mined histograms can be mapped to statistically significant molecular fragments, and this allows chemists to do more than just molecular classification.

Our approach makes the assumption that a low *p*-value in the histogram space corresponds to a low *p*-value in the graph space. Clearly, this is an assumption that needs verification. Unfortunately, it is a very hard problem to derive theoretical bounds on the preciseness of our technique due to the complexity of modeling the distribution of a graph. While we attempt to model the graph distribution through features derived from NNP and node types, they do not always capture the structural dependence. For example, assume the probability of finding P(C-1-Cl >1) and P(C-2-C >3) in an RWR histogram as 0.1 and 0.6, respectively. The probability of finding both in a single histogram when computed using eq 4 as a simple product is 0.06. However, if a Cl atom exists along with a benzene majority of the times, then eq 4 ignores the conditional dependence between them, and therefore, the computed probability will be much lower than the actual probability which is closer to 0.1. As a result, there could be false positives in the answer set. Similarly, a false negative can exist if there is a negative correlation between the co-occurrence of the features. The presence of both false positives and false negatives depends on the conditional dependencies between the features. As long as the correlation between features is low, the exactness of the significant patterns is likely to be high. One natural extension to improve the modeling would be to employ larger features instead of just NNP and node types to capture part of the structural correlations. However, the approach once again results in a combinatorial explosion with too many features.

A different perspective to judge the quality of patterns would be an empirical evaluation. If there is no correlation between the *p*-values of the histogram representations and the actual substructures, the classification accuracy is expected to perform badly. However, as evident in the experimental section, the descriptor based on patterns mined by GraphSig displays good performance. The BEDROC score achieved by GraphSig is higher than Daylight, and the AUC scores are almost identical. Working in the histogram space also brings along the added advantage of not facing the subgraph isomorphism problem and, therefore, further expedites the computations.

Our work is the first technique that is able to mine significant substructures scalably in the presence of a low frequency threshold. While we empirically demonstrated GraphSig's potential as a descriptor in molecular classification, it can be employed to develop other molecular analysis tools as well. Each of the patterns used to define the descriptor holds individual importance on their own and can be used to study a molecular data set. For example, a substructure of AZT is identified as highly significant, underlining the capabilities of GraphSig as a molecular analysis tool. GraphSig opens up a new direction in molecular pattern-based applications by unleashing the potential of significant substructures.

ACKNOWLEDGMENT

We thank X. Yan for sharing the code of LEAP. The work was supported by NSF Grant IIS-0612327 and IIS-0917149. The authors also thank Petko Bogdanov, Kyle Chipman, and Sudipto Das for the very helpful comments during the preparation of the manuscript.

REFERENCES AND NOTES

(1) Willett, P. Searching Techniques for Databases of Two- and three-dimensional Chemical Structures. *J. Med. Chem.* **2005**, *48*, 4183–4199.
(2) Bender, A.; Glen, R. C. Molecular Similarity: A Key Technique in Molecular Informatics. *Org. Biomol. Chem.* **2004**, *2*, 3204–3218.
(3) Keserû, G. M.; Molnár, L.; Greiner, I. A Neural Network Based Virtual High Throughput Screening Test for the Prediction of CNS Activity. *Comb. Chem. High Throughput Screening* **2000**, *3*, 535–540.
(4) Bernazzani, L.; Duce, C.; Micheli, A.; Mollica, V.; Sperduti, A.; Starita, A.; Tiné, M. R. Predicting Physical-chemical Properties of Compounds from Molecular Structures by Recursive Neural Networks. *J. Chem. Inf. Model.* **2006**, *46*, 2030–2042.
(5) Watson, P. Naive Bayes Classification Using 2D Pharmacophore Feature Triplet Vectors. *J. Chem. Inf. Model.* **2008**, *48*, 166–178.
(6) Labute, P. Binary QSAR: A New Method for the Determination of Quantitative Structure Activity Relationships. In *Pacific Symposium on Biocomputing*; World Scientific Publishing Company: Singapore, 1999; pp 444−455.
(7) Fröhlich, H.; Wegner, J. K.; Sieker, F.; Zell, A. Optimal Assignment Kernels for Attributed Molecular Graphs. In *Proceedings of the 22nd International Conference on Machine learning*; ACM: New York, 2005; pp 225−232.
(8) Harper, G.; Bradshaw, J.; Gittins, J. C.; Green, D. V. S.; Leach, A. R. Prediction of Biological Activity for High-Throughput Screening Using Binary Kernel Discrimination. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1295–1300.
(9) Muller, K.-R.; Ratsch, G.; Sonnenburg, S.; Mika, S.; Grimm, M.; Heinrich, N. Classifying "Drug-likeness" with Kernel-Based Learning Methods. *J. Chem. Inf. Model.* **2005**, *45*, 249–253.
(10) Hanna Eckert, J. B. Partitioning Methods for the Identification of Active Molecules. *Curr. Med. Chem.* **2003**, *8*, 707–715.
(11) Zmuidinavicius, D.; Didziapetris, R.; Japertas, P.; Avdeef, A.; Petrauskas, A. Classification Structure-activity Relations (C-SAR) in Prediction of Human Intestinal Absorption. *J. Pharm. Sci.* **2003**, *92*, 621–633.
(12) Stockfisch, T. P. Partially Unified Multiple Property Recursive Partitioning (PUMP-RP): A New Method for Predicting and Understanding Drug Selectivity. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1608–1613.
(13) Rusinko, A.; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. Analysis of a Large Structure/biological Activity Data Set Using Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017–1026.
(14) Yan, X.; Cheng, H.; Han, J.; Yu, P. S. Mining Significant Graph Patterns by Scalable Leap Search. In *Proceedings of SIGMOD '08*; ACM: New York, 2008.
(15) Kubinyi, H. Drug Research: Myths, Hype and Reality. *Nat. Rev. Drug. Discovery* **2003**, *2*, 665–668.
(16) Rhyu, K.-B.; Patel, H. C.; Hopfinger, A. J. A 3D-QSAR Study of Anticoccidial Triazines Using Molecular Shape Analysis. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 771–778.
(17) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual Screening: An Overview. *Drug Discovery Today* **1998**, *35*, 160–178.
(18) *MACCS Structural Keys*; Symyx Software: San Ramon, CA, 2005.
(19) *Daylight Theory Manual*; Daylight Chemical Information Systems Inc.: Aliso Viejo, CA, 2008.
(20) Barnard, J.; Downs, G.; Willett, P. Descriptor-Based Similarity Measures for Screening Chemical Databases. In *Virtual Screening for Bioactive Molecules*; Bohm, H.-J., Schneider, G., Eds.; John Wiley & Sons, Inc.: New York, 2000; Vol. 10, pp 59−80.
(21) Deshpande, M.; Kuramochi, M.; Karypis, G. Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*; IEEE Computer Society: Washington, DC, 2003.
(22) Deshpande, M.; Kuramochi, M.; Wale, N.; Karypis, G. Frequent Substructure-Based Approaches for Classifying Chemical Compounds. *IEEE Trans. Knowledge Data Eng.* **2005**, *17*, 1036–1050.
(23) *JOELib-A Java Based Computational Chemistry Package*; Wilhelm-Schickard-Institute for Computer Science: Tübingen, Germany, 2009.
(24) *Unity*; Tripos Inc.: St. Louis, MO, 2008.
(25) Smalter, A.; Huan, L.; Lushington, G.; Jia, Y. GPD: A Graph Pattern Diffusion Kernel for Accurate Graph Classification with Applications in Cheminformatics. In *Proceedings of BIOKDD*; ACM: New York, 2008.
(26) *ClassPharmer Suite*, version 3.5; Bioreason, Inc.: Santa Fe, NM, 2000.
(27) *Chemistry Component, Scitegic Pipeline Pilot*, 6.1.5.0 student ed.; Accelrys, Inc.: San Diego, CA, 2009.
(28) Kuramochi, M.; Karypis, G. Frequent Subgraph Discovery. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*; IEEE Computer Society: Washington, DC, 2001; pp 313−320.
(29) Yan, X.; Han, J. gSpan: Graph-Based Substructure Pattern Mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*; IEEE Computer Society: Washington, DC, 2002.
(30) Nijssen, S.; Kok, J. N. The Gaston tool for Frequent Subgraph Mining. In *Proceedings of the International Workshop on Graph-Based Tools*; Elsevier: Amsterdam, The Netherlands, 2004.
(31) Ranu, S.; Singh, A. K. GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases. In *Proceedings of the 25th International Conference on Data Engineering*; IEEE Computer Society: Washington, DC, 2009; pp 844−855.
(32) Bush, B. L.; Sheridan, R. P. PATTY: A Programmable Atom Type and Language for Automatic Classification of Atoms in Molecular Databases. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 756–762.
(33) Sheridan, R. P.; Miller, M. D. A Method for Visualizing Recurrent Topological Substructures in Sets of Active Molecules. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 915–924.
(34) Gillet, V. J.; 0002, P. W.; Bradshaw, J. Similarity Searching Using Reduced Graphs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 338–345.

(35) Barker, E. J.; Gardiner, E. J.; Gillet, V. J.; Kitts, P.; Morris, J. Further Development of Reduced Graphs for Identifying Bioactive Compounds. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 346–356.

(36) Birchall, K.; Gillet, V. J.; Harper, G.; Pickett, S. D. Training Similarity Measures for Specific Activities: Application to Reduced Graphs. *J. Chem. Inf. Model.* **2006**, *46*, 577–586.

(37) Harper, G.; Bravi, G.; Pickett, S. D.; Hussain, J.; Green, D. V. S. The Reduced Graph Descriptor in Virtual Screening and Data-Driven Clustering of High-Throughput Screening Data. *J. Chem. Inf. Model.* **2004**, *44*, 2145–2156.

(38) Takahashi, Y.; Sukekawa, M.; Sasaki, S. Automatic Identification of Molecular Similarity Using Reduced-graph Representation of Chemical Structure. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 639–643.

(39) Rarey, M.; Dickson, J. S. Feature Trees: A New Molecular Similarity Measurement Based. *J. Comput.-Aided Mol. Des* **1998**, *12*, 471–190.

(40) Fischer, J. R.; Rarey, M. SwiFT: An Index Structure for Reduced Graph Descriptors in Virtual Screening and Clustering. *J. Chem. Inf. Model.* **2007**, *47*, 1341–1353.

(41) Wolfram MathWorld. http://mathworld.wolfram.com/BinomialDistribution.html (accessed April 29, 2009).

(42) He, H.; Singh, A. K. GraphRank: Statistical Modeling and Mining of Significant Subgraphs in the Feature Space. In *Proceedings of the Sixth International Conference on Data Mining*; IEEE Computer Society: Washington, DC, 2006; pp 885−890.

(43) Vapnik, V. N. *Statistical Learning Theory*; Wiley-Interscience: New York, 1998.

(44) Bajorath, J. Integration of Virtual and High-throughput Screening. *Nat. Rev. Drug. Discovery* **2002**, *1*, 882–894.

(45) Whittle, M.; Gillet, V. J.; Willett, P.; Alex, A.; Loesel, J. Enhancing the Effectiveness of Virtual Screening by Fusing Nearest Neighbor Lists: A Comparison of Similarity Coefficients. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1840–1848.

(46) Willett, P. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983–996.

(47) Swamidass, S. J.; Chen, J.; Bruand, J.; Phung, P.; Ralaivola, L.; Baldi, P. Kernels for Small Molecules and the Prediction of Mutagenicity, Toxicity and Anti-cancer Activity. *Bioinformatics* **2005**, *21*, 359–368.

(48) Huan, J.; Wang, W.; Prins, J.; Yang, J. SPIN: Mining Maximal Frequent Subgraphs from Graph Databases. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, 2004; pp 581−586.

(49) Thomas, L. T.; Valluri, S. R.; Karlapalem, K. MARGIN: Maximal Frequent Subgraph Mining. In *Proceedings of the Sixth International Conference on Data Mining*; IEEE Computer Society: Washington, DC, 2006; pp 1097−1101.

(50) The PubChem Project. http://pubchem.ncbi.nlm.nih.gov (accessed April 29, 2009).

(51) Auer, J.; Bajorath, J. Distinguishing between Bioactive and Modeled Compound Conformations through Mining of Emerging Chemical Patterns. *J. Chem. Inf. Model.* **2008**, *48*, 1747–1753.

(52) Chang, C.-C.; Lin, C.-J. LIBSVM: a library for support vector machines, 2009. http://www.csie.ntu.edu.tw/cjlin/libsvm (accessed April 29, 2009).

(53) Truchon, J.-F. F.; Bayly, C. I. Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem. *J. Chem. Inf. Model.* **2007**, *47*, 488–508.

(54) *MDL Drug Data Report*; Symyx Software: San Ramon, CA, 2005.

(55) DTP-AIDS Antiviral Screen Data. http://dtp.nci.nih.gov/docs/aids/aids_data.html (accessed April 29, 2008).

(56) Wilson, I. K.; Chatterjee, S.; Wolf, W. Synthesis of 3′-fluoro-3′-deoxythymidine and Studies of its 18F-radiolabeling, as a Tracer for the NoninvasiveMonitoring of the Biodistribution of Drugs Against AIDS. *J. Fluorine Chem.* **1991**, *55*, 283–289.

(57) Manetta, A.; Gamboa, G.; Nasseri, A.; Podnos, Y. D.; Emma, D.; Dorion, G.; Rawlings, L.; Carpenter, P. M.; Bustamante, A.; Patel, J.; Rideout, D. Novel Phosphonium Salts Display In Vitro and In Vivo Cytotoxic Activity Against Human Ovarian Cancer Cell Lines. *Gynecol. Oncol.* **1996**, *60*, 203–212.

(58) Yan, X.; Han, J. CloseGraph: Mining Closed Frequent Graph Patterns. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, 2003; pp 286−295.

CI900035Z