Acyclic Subgraph based Descriptor Spaces for Chemical Compound Retrieval and Classification*

Nikil Wale and George Karypis

Department of Computer Science & Engineering,
Digital Technology Center, Army HPC Research Center
University of Minnesota, Minneapolis, MN 55455

{nwale, karypis}@cs.umn.edu.

Technical Report: #06–008

Abstract

In recent years the development of computational techniques that build models to correctly assign chemical compounds to various classes or to retrieve potential drug-like compounds has been an active area of research. These techniques are used extensively at various phases during the drug development process. Many of the best-performing techniques for these tasks, utilize a descriptor-based representation of the compound that captures various aspects of the underlying molecular graph's topology. In this paper we introduce and describe algorithms for efficiently generating a new set of descriptors that are derived from all connected acyclic fragments present in the molecular graphs. In addition, we introduce an extension to existing vector-based kernel functions to take into account the length of the fragments present in the descriptors. We experimentally evaluate the performance of the new descriptors in the context of SVM-based classification and ranked-retrieval on 28 classification and retrieval problems derived from 17 datasets. Our experiments show that for both the classification and retrieval tasks, these new descriptors consistently and statistically outperform previously developed schemes based on the widely used fingerprint- and Maccs keys-based descriptors, as well as recently introduced descriptors obtained by mining and analyzing the structure of the molecular graphs.

1 Introduction

Discovery, design and development of new drugs is an expensive and challenging process. Any new drug should not only produce the desired response to the disease but should do so with minimal side effects. One of the key steps in the drug design process is the identification of the chemical compounds (*hit* compounds or just *hits*) that display the desired and reproducible behavior against the specific biomolecular target [22]. This represents a significant hurdle in the early stages of drug discovery. Therefore, computational techniques that build models to correctly assign chemical compounds to various classes or retrieve compounds of desired class from a database have become popular in the pharmaceutical industry.

Over the last twenty years extensive research has been carried out to identify representations of molecular graphs that can build good classification models or retrieve actives from a database in an effective way. Towards this goal, a number of different approaches have been developed that represent each compound by a set of descriptors that are based on frequency, physiochemical properties as well as topological and geometric substructures (fragments) [1, 3, 6, 8, 13, 28–30, 36]. Historically, the best performing and most widely used descriptors have been based on fingerprints, which represent each molecular graph by a fixed length bit-vector derived by enumerating all bounded length cycles and paths in the graph (e.g., Daylight [29]), and on sets of fragments that have been identified a priori by domain experts (e.g., Maccs keys [30]). However, in recent years, research in the data mining community has generated new classes of descriptors based on frequently occurring substructures [8] and selected cycles & trees [13] that have been shown to achieve promising results.

In this paper, we build on the experience gained from this earlier work and introduce a new set of fragment-based descriptors that are designed to better capture the underlying structure of molecular graphs. These descriptors are derived from all connected acyclic fragments (AF) present in the graphs and their length (number of bonds) is constrained not to exceed a user-supplied parameter. We present an efficient algorithm for finding these descriptors and study their effectiveness for the tasks of building classification models and of retrieving active compounds from a chemical compound library. Within the context of these tasks we also study the effectiveness of different descriptor-based similarity measures for both deriving kernel functions for SVM-based classifica-

*This work was supported in part by NSF EIA-9986042, ACI-0133464, IIS-0431135, and NIH RLM008713A; the Digital Technology Center at the University of Minnesota; and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under Cooperative Agreement number DAAD19-01-2-0014. The content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

tion and for ranked-retrieval.

To assess the effectiveness of the new class of descriptors we perform a comprehensive experimental study using 28 different classification and retrieval problems derived from 17 datasets containing up to 78,995 compounds. Our study compares the performance achieved by the acyclic fragments to that achieved by previously developed schemes (fingerprints [14], Maccs keys [30], frequent sub-structures [8], Cycles & Trees [13]) as well as two subsets of AF, one containing the fragments that form paths (PF) and the other containing the fragments that form trees (TF).

Our experiments show that for both the classification and the retrieval tasks, the AF descriptors consistently and statistically outperform all previously developed schemes. Moreover, a kernel function introduced in this paper that takes into account the length of the fragments present in the set of descriptors lead to better overall results, especially when used with the AF descriptors.

The rest of the paper is organized as follows. Section 2 provides some background on the molecular graph representation of chemical compounds. Section 3 describes the previously developed descriptors used in our experimental evaluation. Section 4 provides a detailed description of the descriptors introduced in this paper. Section 5 provides a detailed description of the various kernel functions used. Section 6 contains experimental evaluation of the different descriptors and also provides some trends and analysis from the experiments. Section 7 provides concluding remarks on this work.

2 Representation of Chemical Compounds

In this paper we represent each compound by its corresponding molecular graph [19]. The vertices of these graphs correspond to the various atoms (e.g., carbon, nitrogen, oxygen, etc.), and the edges correspond to the bonds between the atoms (e.g., single, double, etc.). Each of the vertices and edges has a label associated with it. The labels on the vertices correspond to the type of atoms and the labels on the edges correspond to the type of bonds. The vertex labels (atom typing) and edge labels (bond typing) used in this paper for all the input chemical graphs and descriptors generated from them (except fingerprints and Maccs keys) is the default typing used by Babel [23]. We apply two commonly used structure normalization transformations [22]. First, we label all bonds in aromatic rings as *aromatic* (i.e., a different edge-label), and second, we remove the hydrogen atoms that are connected to carbon atoms (i.e., hydrogen-suppressed chemical graphs). To generate fingerprints and Maccs keys we use the Smiles [29] representation as an input.

3 Overview of Existing Fragment-Based Descriptor Spaces

In this section, we briefly describe some of the most popular as well as recently introduced approaches to extract fragment-based descriptors from molecular graphs.

3.1 Fingerprints

Fingerprints [29] are used to encode structural characteristics of a chemical compound into a fixed bit vector and are used extensively for various tasks in chemical informatics. These fingerprints are typically generated by enumerating all cycles and linear paths up to a given number of bonds and hashing each of these cycles and paths into a fixed bit-string. The specific bit-string that is generated depends on the number of bonds, the number of bits that are set, the hashing function, and the length of the bit-string. A desirable property of the fingerprint-based descriptors is that they encode a very large number of sub-structures into a compact representation. We will refer to these descriptors as *fp-n* where n is the number of bits that are used.

3.2 Maccs Keys (MK)

Molecular Design Limited (MDL) created the key based fingerprints (Maccs Keys) [30] based on pattern matching of a chemical compound structure to a pre-defined set of structural fragments that have been identified by domain experts [9]. Each such structural fragment becomes a key and occupies a fixed position in the descriptor space. Therefore, this approach relies on pre-defined rules to encapsulate the molecular descriptions a-priori and does not learn them from the chemical dataset.

This descriptor space is notably different from fingerprint based descriptor space. Unlike fingerprints, no *folding* (hashing) is performed on the sub-structures. The advantage of such an approach over fingerprints is that sub-structures of arbitrary topology can form a part of the descriptor space. Moreover, the rules selected encode domain knowledge in a compact descriptor space. But it also has a disadvantage of potentially not being able to adapt to the characteristics for a particular dataset and classification problem. We will refer to this descriptor space as *MK*.

3.3 Cyclic patterns and Trees (CT)

Horovath *et al* [13] developed a method that is based on representing every compound as a set of cycles and certain kinds of trees. In particular, the idea is to identify all the bi-connected components (blocks) of a chemical graph. Once these blocks are identified, the first set of features is generated by enumerating up to a certain number of simple cycles (bounded cyclicity) for the blocks. Once the cycles are identified, all the blocks of the chemical graph are deleted. The resulting graph is a collection of leftover trees forming a forest. Each such tree is used as a descriptor. The final descriptor space is the union of the cycles and leftover trees. The tree patterns used in this representation are of a specific topology and size that depends on the position of blocks in the chemical graph. We will refer to this descriptor space as *CT*.

3.4 Frequent Sub-structures based Descriptor Space (FS)

A number of methods have been proposed in recent years to find frequently occurring sub-structures in a chemical graph database [4, 15, 21, 37]. Frequent sub-structures of a chemical graph database $\mathbf{D}$ are defined as all sub-structures that are present in at least $\sigma|\mathbf{D}|\%$ of compounds of the database, where σ is the minimum frequency requirement (also called minimum support constraint). These frequent sub-structures can be used as descriptors for the compounds in that database. One of the important properties of the sub-structures generated, like Maccs Keys, is that they can have arbitrary topology. Moreover, every sub-structure generated is connected and frequent (as determined by the minimum support constraint σ).

Descriptor space formed out of frequently occurring sub-structures depends on the value of σ. Therefore, unlike the Maccs keys, the descriptor space can change for a particular problem instance if the value of σ is changed. Moreover, unlike fingerprints, all frequent subgraphs irrespective of their size (number of bonds) form the descriptor space. A potential disadvantage of this method is that it is unclear how to select a suitable value of σ for a given problem. A very high value will fail to discover important sub-structures whereas a very low value will result in combinatorial explosion of frequent subgraphs. We will refer to this descriptor space as *FS*.

4 Acyclic, Tree and Path Fragments (AF, TF, and PF)

A careful analysis of the four descriptor spaces described in Section 3 illustrate four dimensions along which these schemes compare with each other and represent some of the choices that have been explored in designing fragment-based (or fragment-derived) descriptors for chemical compounds. The first dimension is associated with weather the fragments are determined directly from the dataset at hand or they have been pre-identified by domain experts. Maccs keys is an example of a descriptor space whose fragments have been determined a priori whereas in all other schemes, the fragments are determined directly from the dataset. The second dimension is associated with the topological complexity of the actual fragments. On one side of the spectrum, schemes like fingerprints use rather simple topologies consisting of cycles and paths, whereas at the other end of the spectrum, the frequent sub-structure-based descriptors allow fragments that correspond to arbitrarily connected subgraphs. The third dimension is associated with weather or not the fragments are being precisely represented in the descriptor space. Fingerprint-based descriptors, due to the hashing approach that they use, lead to imprecise representations, whereas the other three schemes are precise in the sense that there is a one-to-one mapping between fragments and dimensions of the descriptor space. Finally, the fourth dimension is associated with the ability of the descriptor space to cover all (or nearly all) of the dataset. Descriptor spaces created from fingerprints and cycles & trees are guaranteed to contain fragments or hashed fragments from each one of the compounds. On the other hand, descriptor spaces corresponding to Maccs keys and frequent sub-structures may lead to a descriptor-based representation of the dataset in which some of the compounds have no (or a very small number) of descriptors. Descriptor spaces that are determined dynamically from the dataset, use fragments with complex topologies, lead to precise representations, and have a high degree of coverage are expected to perform better in the context of chemical compound classification and retrieval as they allow for a better representation of the underlying compounds.

In this section we introduce and describe algorithms for efficient generation of a new descriptor space that we believe better captures the desired characteristics along the above four dimensions. This descriptor space consists of all connected acyclic fragments up to a given length l (i.e., number of bonds) that exist in the dataset at hand. The descriptor space is determined dynamically from the dataset, the topology of the fragments that it allows are trees and paths, leads to a precise representation, and has 100% coverage. We will refer to this descriptor space as *Acyclic Fragments (AF)*.

In addition, we also derive two other sets of fragments from the set of all acyclic fragments. The first, termed as *Tree Fragments* (TF), is the collection of all fragments that have at least one node of degree greater than two. This set forms all the tree fragments. The second set, called *Path Fragments* (PF), is just the set of linear paths where the degree of every node in every fragment is less than or equal to two. Note that AF $=$ TF $\cup$ PF and TF $\cup$ PF $= \emptyset$.

Note that Path Fragments are exactly the same patterns as the linear paths in fingerprints. Moreover, any frequent sub-structure based descriptor space is a superset of Acyclic-Fragments when the minimum support threshold (σ) is low enough to generate frequent subgraphs having a frequency of one.

4.1 Efficient Generation of Acyclic Fragments

To generate all connected acyclic fragments, we developed an algorithm that was inspired by the recursive technique for generating all the spanning trees of a graph G [34].

Consider an arbitrary edge e of G, and let $S_e(G)$ be the set of spanning trees of G that contain e and $S_{\neg e}(G)$ be the set of all spanning trees of G that do not contain e. It is easy to see that (i) $S_e(G) \cap S_{\neg e}(G) = \emptyset$ and (ii) $S_e(G) \cup S_{\neg e}(G)$ is equal to the set of all spanning trees of G, denoted by $S(G)$. Now, if $S(G/e)$ denotes an *edge contraction* operation (i.e., the vertices incident on e are collapsed together) then $S_e(G)$ can be obtained from $S(G/e)$ by adding e. If $G\backslash e$ denotes an *edge deletion* operation, then $S_{\neg e}(G)$ is nothing more than $S(G\backslash e)$. From the above observations we can come up with the following recurrence relation for generating $S(G)$

$$S(G) = \begin{cases} \emptyset, \text{ if } G \text{ does not have any edge} \\ eS(G/e) \cup S(G\backslash e), \text{ otherwise,} \end{cases} \quad (1)$$

where e is an arbitrary edge of G, and $eS(G/e)$ denotes the set of all spanning trees obtained by adding e to each spanning tree in $S(G/e)$.

The recurrence relation of Equation 1 can be used to generate all the connected acyclic fragments of a certain length l by modifying it in two different ways. These modifications are needed to ensure that (i) the acyclic fragments that are returned are connected, and (ii) only all the fragments of length l are returned. The first can be achieved by imposing the constraint that the edge e must be incident on a vertex of G that was obtained via an edge contraction operation, if such a vertex exist. If G does not have any such vertex (i.e., it corresponds to the original graph), then e is selected in an arbitrary fashion. The length requirement can be ensured by terminating the recurrence relation when exactly l edges have been selected. In light of these modifications, the new recurrence relation that generates all the connected acyclic fragments of length l, denoted by $F(G,l)$ is given by

$$F(G,l) = \begin{cases} \emptyset, & \text{if } G \text{ has fewer than } l \text{ edges or } l=0 \\ eF(G/e,l-1) \cup F(G\backslash e,l), & \text{otherwise,} \end{cases} \tag{2}$$

where e is satisfies the above constraints.

5 Kernel Functions for chemical compound classification

Given the descriptor space, each chemical compound can be represented by a vector X whose i^{th} dimension will have a non-zero value if the compound contains that descriptor and will have a value of zero otherwise. The value for each descriptor that is present can be either one, leading to a vector representation that captures presence or absence of the various descriptors (referred to as binary vectors) or the number of times that each descriptor occurs in the compound, leading to a representation that also captures the frequency information (referred to as frequency vectors).

Given the above vector representation of the chemical compounds, the classification algorithms that we develop in this paper use support vector machines (SVM) [32] as the underlying learning methodology, as they have been shown to be highly effective, especially in high dimensional spaces.

One of the key parameters that affects the performance of SVM is the choice of the kernel function ($\mathcal{K}$), that measures the similarity between pairs of compounds. Any function can be used as a kernel as long as, for any number n and any possible set of distinct compounds $\{X_1,\ldots,X_n\}$, the $n \times n$ Gram matrix defined by $\mathcal{K}_{i,j} = \mathcal{K}(X_i,X_j)$ is symmetric positive semidefinite. These functions are said to satisfy Mercer's conditions and are called Mercer kernels, or simply valid kernels.

In this paper we use two different classes of kernel functions that are derived from the widely used RBF kernel function, and the less widely used Tanimoto coefficient[1] [2,3,5,35]. The Tanimoto coefficient was selected because it is used extensively in cheminformatics and has been shown to be an effective way to measure the similarity between chemical compound pairs [36].

Given the vector representation of two compounds X and Y, the RBF and Tanimoto kernel functions are given by

$$\mathcal{K}_{rbf}(X,Y) = \exp(-\frac{\|X-Y\|}{2\sigma^2}) \tag{3}$$

$$\mathcal{K}_{tm}(X,Y) = \frac{\sum_{i=1}^{M} min(x_i,y_i)}{\sum_{i=1}^{M} max(x_i,y_i)}, \tag{4}$$

where σ is a user supplied parameter and the terms x_i and y_i are the values along the i^{th} dimension of the X and Y vectors, respectively. Note that in the case of binary vectors, these will be either zero or one, whereas in the case of frequency vectors these will be equal to the number of times the i^{th} descriptor exists in the two compounds. Moreover, note that Tanimoto kernel is a valid kernel as it has been shown to satisfy Mercer's conditions [28].

One of the potential problems in using the above kernels with descriptor spaces that contain fragments of different lengths is that they contain no mechanism to ensure that descriptors of various lengths contribute in a non-trivial way to the computed kernel function values. This is especially true for the AF, TF, and PF descriptor spaces in which each compound tends to have a much larger number of longer length fragments (e.g. length six and seven) than shorter length (e.g. length two and three). To overcome this problem we modified the above kernel functions to give equal weight to the fragments of each length. In the context of the RBF kernel function, this is obtained as follows. Let X^l and Y^l be the feature vectors of X and Y with respect to only the features of length l, and let L be the length of the largest feature. Then, the length-differentiated RBF kernel function $\mathcal{K}^*_{rbf}(X,Y)$ is given by

$$\mathcal{K}^*_{rbf}(X,Y) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{K}_{rbf}(X^l,Y^l). \tag{5}$$

The length-differentiated kernels for Tanimoto is derived in a similar fashion. We will refer to these as the *length-differentiated kernel functions*, and we will refer to the ones that do not differentiate between different length fragments as *pooled kernel functions*.

In summary, we studied four different flavors for each kernel functions, one that is binary and pooled, frequency and pooled, binary and length-differentiated and frequency and length-differentiated. We will follow the convention of using the symbols $\mathcal{K}_b$, $\mathcal{K}_f$, $\mathcal{K}^*_b$, and $\mathcal{K}^*_f$ to refer to binary and pooled, frequency and pooled, binary and length-

[1] We also experimented with the linear kernel function but the results were worse that either RBF or Tanimoto, so we are not including them here.

differentiated and frequency, and length-differentiated kernel functions, respectively.

6 Results

6.1 Datasets

The performance of the different descriptors and kernel functions was assessed on 28 different classification problems from 17 different datasets.

The size, distribution and compound characteristics of the 28 classification problems are shown in Table 1. Each of the 28 classification problems is unique in that it has different distribution of positive class (ranging from 1% in H2 to 50% in C1), different number of compounds (ranging from the smallest with 559 compounds to largest with 78,995 compounds) and compounds of different average sizes (ranging from the 14 atoms per compound to 37 atoms per compound on an average in C1 and H3 respectively).

Table 1: Properties of classification problems and Datasets.

D	N	$N+$	N_A	N_{A+}	N_{A-}	N_B	N_{B+}	N_{B-}
NCI1	39001	1881	26	34	25	28	37	27
NCI109	39168	1893	26	34	25	28	37	27
NCI123	39497	2885	26	32	25	28	34	27
NCI145	38665	1786	26	34	25	28	37	27
NCI167	78995	9416	21	24	21	22	25	22
NCI220	866	282	24	24	25	26	25	26
NCI33	38649	1500	26	35	25	28	38	27
NCI330	41152	2266	22	28	21	23	30	23
NCI41	26425	1395	26	35	26	28	38	28
NCI47	38922	1840	26	34	25	28	37	27
NCI81	39199	2201	26	33	25	28	36	27
NCI83	26636	2092	26	33	25	28	35	28
H1	42389	1498	27	37	26	29	39	28
H2	41313	422	27	43	26	29	45	28
A1	34836	12376	25	25	25	25	25	25
H3	1498	422	37	43	34	39	45	37
D1	1309	116	24	27	23	25	28	25
D2	1305	112	24	25	23	25	27	25
D3	1501	308	26	36	23	28	38	25
D4	1728	536	26	32	23	28	34	25
P1	567	212	18	17	19	19	18	20
P2	574	164	19	17	19	19	18	20
P3	572	193	18	16	19	19	17	20
P4	559	170	18	17	19	19	17	20
C1	640	320	14	13	15	14	14	15
M1	1596	285	16	14	16	16	15	17
M2	1596	172	16	13	16	16	14	17
M3	1596	88	16	13	16	16	13	17

N is the total number of compounds in the dataset. $N+$ is the number of positives in the dataset. N_A and N_B are the average number of atoms and bonds in each compound. N_{A+} is the average number of atoms in each compound belonging to the positive class and N_{A-} is the average number of atoms in each compound belonging to the negative class. Similarly N_{B+} and N_{B-} are the corresponding numbers for bonds. The numbers are rounded off to the nearest integer.

The first dataset is a part of the Predictive Toxicology Evaluation Challenge [27]. There are four classification problems one corresponding to each of the rodents MaleRats, FemaleRats, MaleMice and FemaleMice and will be referred as *P1*, *P2*, *P3*, and *P4*.

The second dataset is mutagenicity data from [12]. The compounds in this dataset are classified as mutagens or nonmutagens as determined by the *Salmonella*/microsome assay. We will refer this dataset as *C1*.

The third dataset is obtained from the National Cancer Institutes's DTP AIDS Anti-viral Screen program [20, 26]. Three classification problems are formulated out of this dataset. The first problem is designed to classify between CM+CA and CI; the second between CA and CI, and the third between CA and CM. We will refer to these problems as *H1*, *H2*, and *H3*, respectively.

The fourth dataset was obtained from the Center of Computational Drug Discovery's anthrax project at the University of Oxford [25]. The classification problem for this dataset is: given a chemical compound classify it in to one of these two classes, i.e., will the compound bind the anthrax toxin or not. This classification problem is referred as *A1*.

A fifth dataset is provided by Dr. Ian Watson from Eli Lilly Inc. and is described in [33]. Each drug compound in this dataset is marked as Oral (O), Topical (T), Absorbent (A) or Injectable (I) depending on the mode of administration of that drug. Four classification tasks are defined from this dataset: between Oral and Absorbent *D1*, between Oral and Topical *D2*, between Oral and Injectable *D3* and between Oral and everything else (Topical + Absorbent + Injectable) as *D3*. This dataset is particularly different from the rest, in that we try to distinguish between the 1728 marketed drugs with different modes of administration.

Another dataset used in this study is the MAO (Monoamine Oxidase) dataset [7]. The compounds of this dataset have been categorized into four different classes (0, 1, 2 and 3) based on the levels of activity, with the lowest labeled as 0 and the highest labeled as 3. We define three classification problems based on this dataset: *M1* with positive class compounds as labels 1, 2 and 3 and negative class as compounds with label 0, *M2* with positive class as labels 2 and 3 and negative class compounds as labels 0 and 1, and finally the last problem *M3* with positive class compounds as label 3 and rest of the compounds in negative class.

The rest of the datasets are derived from the PubChem website that pertain to the cancer cell lines [24]. Twelve datasets are selected from the bioassay records for cancer cell lines. Each of the NCI anti-cancer screens forms a classification problem. The datasets that are selected belong to 12 different types of cancer screen. Since there is more than one screen available for any particular types of cancer (for example colon cancer, breast cancer *etc*.), we decided to use the screen that had most number of compounds tested on it. The class labels on these datasets is either active or inactive and we used the original class labels associated with each compound. Table 2 proves details of the 12 different bioassays used for this study.

All the datasets required some data cleaning as for some of the compounds we were unable to generate all of the seven descriptor spaces. All such compounds were removed from their respective datasets. This made the sets of compounds used for different descriptors exactly the same and allowed objective comparison of the seven descriptor spaces.

6.2 Experimental Methodology

The classification results were obtained by performing a 5-way cross validation on the dataset, ensuring that the class distribution in each fold is identical to the original dataset. In

Table 2: Description of NCI cancer screen datasets.

Name (Bioassay-ID or AID)	Description
NCI-H23 (NCI1)	Human tumor (Non-Small Cell Lung) cell line growth inhibition assay
OVCAR-8 (NCI109)	Human tumor (Ovarian) cell line growth inhibition assay
MOLT-4 (NCI123)	Human tumor (Leukemia) cell line growth inhibition assay
SN12C (NCI145)	SN12C Renal cell line
Yeast anti-cancer (NCI167)	Yeast anti-cancer screen bub3 strain
CD8F1 (NCI220)	In Vivo Anticancer Screen Tumor model Mammary Adenocarcinoma
UACC257 (NCI33)	Human tumor (Melanoma) cell line growth inhibition assay
P388 in CD2F1 (NCI330)	In Vivo Anticancer Screen tumor model P388 Leukemia (intraperitoneal)
PC-3 (NCI41)	Human tumor (Prostate) cell line growth inhibition assay
SF-295 (NCI47)	Human tumor (Central Nervous System) cell line growth inhibition assay
SW-620 (NCI81)	Human tumor (Colon) cell line growth inhibition assay
MCF-7 (NCI83)	Human tumor (Breast) cell line growth inhibition assay

each one of the cross validation experiments, the test-set was never considered and the algorithm used only the training-set to generate the descriptor space representation and to build the classification model. The exact same training and test sets were used in descriptor generation and cross validation experiments for all the different schemes. For the SVM classifier we used the SVMLight library [17] with all the default parameter settings except the kernel.

The performance of the newly developed descriptor spaces was compared against the descriptors generated by fingerprints, Maccs Keys, Cycles & Trees, and frequent substructures. For fingerprints, we used Chemaxon's fingerprint program called Screen [14]. We experimented using 256-, 512-, 1024-, 2048-, 4196- and 8192-bit length fingerprints. We used default settings of the two parameters: number of bonds or maximum length of the pattern generated (up to seven) and number of bits set by a pattern (three). We found that 8192-bits produced better results (even though their performance advantage was not statistically significant compared to 2048- and 4196-bit fingerprints). For this reason, we use 8192-bit fingerprints in all the comparisons against other descriptors. To generate MDL Maccs keys (166 keys) we use the MOE suite by Chemical Computing Group [11] For Cyclic patterns and Trees, we use 1000 as the upper bound on the number of cycles to be enumerated as described in [13]. To generate frequent sub-structures, we use the FSG algorithm described in [21]. Table 3 contains the values of σ used for positive and negative classes in each dataset.

In the context of fp-8192 the only kernel applicable is the binary and pooled ($\mathcal{K}_b$) extension of RBF and Tanimoto kernels. This is because hashed fingerprints are inherently binary and not provide frequency information. In the context of MK, only two kernels ($\mathcal{K}_b$ and $\mathcal{K}_f$) are applied. Also for the RBF kernel, we normalize the vectors to be unit length prior to learning the SVM models. We found that this normalization lead to somewhat better results.

6.3 Performance Assessment Measures

The classification performance was assessed by computing the ROC50 values [10], which is the area under the ROC curve up to the first 50 false positives. This is a much more appropriate performance assessment measure than traditional ROC value for datasets with very small positive classes. This is because for such problem settings, a user will most likely stop examining the highest scoring predictions as soon as he/she starts encountering a certain number of false positives [10].

Table 3: Support values for FS.

Datasets	σ_-%	σ_+%	*Datasets*	σ_-%	σ_+%
NCI1	5.0	7.0	A1	5.0	3.0
NCI109	4.0	4.0	H3	8.0	8.0
NCI123	4.0	5.0	D1	5.0	10.0
NCI145	4.0	6.0	D2	5.0	32.0
NCI167	2.0	2.0	D3	5.0	10.0
NCI220	5.0	8.0	D4	5.0	12.0
NCI33	4.0	4.0	P1	3.0	3.0
NCI330	4.0	8.0	P2	3.0	3.0
NCI41	4.0	6.0	P3	3.0	3.0
NCI47	4.0	5.0	P4	3.0	3.0
NCI81	5.0	6.0	C1	2.0	2.0
NCI83	4.0	4.0	M1	1.5	1.75
H1	8.0	5.0	M2	1.45	1.5
H2	8.0	8.0	M3	1.25	3.0

We assess the ability of a particular descriptor set to identify positive compounds in the context of database screening experiment by looking at the fraction of positive compounds that were recovered in the top k hits. Specifically, we report the fraction of positives recovered in the top k hits in a database screening experiment in which every positive compound is used as query. We call this metric *normalized hit rate* (NHR) and it is computed as follows. Suppose N is the number of compounds in a dataset, N_+ is the number of positive (active) compounds in that dataset and $hits_k$ is the number of positives found in the top k hits over all queries. Then, the normalized hit rate is given by

$$\text{NHR} = \frac{hits_k}{(kN_+)}. \tag{6}$$

To compare the performance of a set of schemes across the different datasets, we compute a summary statistics that we refer to as the *Average Relative Quality to the Best (ARQB)* as follows: Let $r_{i,j}$ be the ROC50 (NHR) value achieved by the scheme j on the dataset i, and let r_i^* be the maximum (i.e. the best) ROC50 (NHR) value achieved for this dataset over all the schemes. Then the ARQB for scheme j is equal to $\frac{1}{T}\left(\sum_i \frac{r_{i,j}}{r_i^*}\right)$, where T is the number of datasets. An ARQB value of one indicates that the scheme achieved the best results for all the datasets compared to the other schemes, and a low ARQB value indicates a poorly performing scheme.

We used the Wilcoxon's paired signed-rank test [16] to compare the statistical significance of any two descriptors based on the performance measures described above. This test takes into account not only the sign of differences but also magnitude of these differences. It is generally a more powerful test than student t-test especially for small number of samples with unknown distributions. A p-value of 0.01 is used as threshold for all comparisons.

6.4 Sensitivity on the Length of AF Descriptors

To evaluate the impact of the fragment length in the classification performance achieved by the AF descriptors, we performed a study in which we varied the maximum fragment length l from two to seven bonds. The results of this study are shown in Table 4. These results were obtained using the $\mathcal{K}_f^*$ Tanimoto-based kernel, which as will be shown later, is one of the best performing kernels.

Table 4: ROC50 results for the Tanimoto $\mathcal{K}_f^*$ kernel for different lengths using AF descriptors.

D	up to $l = 2$	up to $l = 3$	up to $l = 4$	up to $l = 5$	up to $l = 6$	up to $l = 7$
NCI1	0.282	0.282	0.297	0.305	0.312	0.317
NCI109	0.266	0.266	0.278	0.285	0.290	0.296
NCI123	0.246	0.246	0.256	0.259	0.264	0.262
NCI145	0.292	0.292	0.306	0.319	0.328	0.334
NCI167	0.061	0.061	0.062	0.064	0.064	0.065
NCI220	0.252	0.252	0.247	0.244	0.240	0.238
NCI33	0.268	0.268	0.289	0.306	0.314	0.318
NCI330	0.327	0.327	0.338	0.343	0.343	0.341
NCI41	0.311	0.311	0.329	0.340	0.350	0.355
NCI47	0.269	0.269	0.284	0.294	0.302	0.305
NCI81	0.269	0.269	0.277	0.286	0.272	0.294
NCI83	0.293	0.293	0.306	0.314	0.316	0.316
H1	0.256	0.256	0.262	0.267	0.271	0.274
H2	0.603	0.603	0.615	0.624	0.634	0.641
A1	0.138	0.138	0.154	0.170	0.201	0.203
H3	0.602	0.602	0.613	0.620	0.626	0.632
D1	0.324	0.324	0.340	0.357	0.363	0.374
D2	0.552	0.552	0.566	0.577	0.580	0.583
D3	0.509	0.509	0.518	0.528	0.532	0.534
D4	0.466	0.466	0.479	0.485	0.489	0.490
P1	0.586	0.586	0.588	0.589	0.596	0.598
P2	0.516	0.516	0.514	0.506	0.501	0.500
P3	0.551	0.551	0.553	0.554	0.555	0.553
P4	0.634	0.634	0.642	0.649	0.651	0.653
C1	0.776	0.795	0.798	0.807	0.813	0.818
M1	0.446	0.446	0.443	0.439	0.436	0.438
M2	0.623	0.623	0.618	0.611	0.612	0.616
M3	0.775	0.775	0.773	0.770	0.773	0.777
ARQB	0.930	0.931	0.955	0.973	0.985	0.995

From these results we can see that the classification performance tends to improve as l increases, and the scheme that use up to length seven fragments achieve the best overall performance. Most of these differences are statistically significant with the only exception being $l = 2$ and $l = 3$, which are not statistically different for $p = 0.01$.

Table 5 shows the number of acyclic fragments of various length that were generated for each dataset, as well as the time required to generate the fragments of length seven. These results show that the number of fragments does increase considerably with l, which essentially puts a practical upper bound on the length of the fragments that can be used for classification. In fact, for $l = 8$ (not shown here), the number of fragments were about three to five times more than that for $l = 7$, which made it impractical to build SVM-based classifier for many of the datasets. However, on the positive side, the amount of time required to generate these fragments is quite small, and is significantly lower than that required for learning the SVM models.

Table 5: Numbers of AF for different lengths l.

	# of fragments			runtime (in sec)
D	$l = 3$	$l = 5$	$l = 7$	for $l = 7$
NCI1	6258	95835	1033757	1022
NCI109	6286	96124	1035681	1007
NCI123	6177	94701	1021345	1008
NCI145	6258	95403	1027123	998
NCI167	8537	123165	1250149	1338
NCI220	1568	13082	82992	22
NCI33	6203	95105	1026732	1030
NCI330	7378	101201	954487	796
NCI41	5313	80157	835764	724
NCI47	6237	95552	1030241	1028
NCI81	6278	95900	1035657	1015
NCI83	5349	80674	840101	716
H1	14369	170230	1389487	1312
H2	14248	168488	1371833	1251
A1	3231	66357	725401	434
H3	2757	23655	137779	61
D1	2127	18888	103159	26
D2	2118	18540	100798	28
D3	2243	20575	117385	35
D4	2336	21636	123910	42
P1	1217	7968	37164	8
P2	1238	8098	37914	9
P3	1239	7959	36774	8
P4	1239	8004	37243	8
C1	1135	6495	29110	6
M1	1301	9531	38812	10
M2	1301	9531	38812	10
M3	1301	9531	38812	10

Due to space constraints we omitted the results for l equal to 2, 4 and 6.

6.5 Effectiveness of Different Kernels for AF Descriptor

Table 6 shows the classification performance of the different kernel functions described in Section 5 for the AF descriptors. These results were obtained for AF descriptors containing fragments of length up to seven.

Two key observations can be made from analyzing these results. First, the classification performance obtained by the Tanimoto-based kernel functions is in general higher than that obtained by the RBF-based kernels. This result is to a large extent in agreement with the widely accepted opinion within the cheminformatics community that Tanimoto coefficient is a good similarity measure for chemical compounds [36]. Second, the best performing kernel function among those based on Tanimoto, is the $\mathcal{K}_f^*$ (length-differentiated-frequency vectors), which is different from the best performing kernel function in the case of RBF, which is $\mathcal{K}_b^*$ (length-differentiated-binary vectors). However, for both classes of kernels, giving equal weights to the fragments of various lengths leads to better results.

Note that based on the Wilcoxon statistical test of $p = 0.01$, the differences between $\mathcal{K}_b^*$ and $\mathcal{K}_f^*$ for Tanimoto are not significant, but $\mathcal{K}_f^*$ is statistically better than $\mathcal{K}_b$ and $\mathcal{K}_f$. Also, in the case of RBF, $\mathcal{K}_b^*$ is statistically better than the other three, which are statistically equivalent among them.

Table 6: ROC50 values for the AF descriptors using kernels derived from Tanimoto and RBF.

	Tanimoto				RBF			
Datasets	($\mathcal{K}_b$)	($\mathcal{K}_f$)	($\mathcal{K}_b^*$)	($\mathcal{K}_f^*$)	($\mathcal{K}_b$)	($\mathcal{K}_f$)	($\mathcal{K}_b^*$)	($\mathcal{K}_f^*$)
NCI1	0.312	0.313	0.304	**0.317**	0.303	0.286	0.305	0.271
NCI109	0.296	**0.297**	**0.297**	0.296	0.271	0.265	0.292	0.256
NCI123	0.253	0.253	0.251	**0.262**	0.252	0.241	0.247	0.235
NCI145	0.330	0.330	0.323	**0.334**	0.283	0.293	0.322	0.284
NCI167	0.062	0.063	0.062	**0.065**	0.060	0.061	0.062	0.059
NCI220	0.230	0.221	0.254	0.238	0.266	0.281	0.263	**0.299**
NCI33	0.311	0.311	0.303	**0.318**	0.285	0.288	0.304	0.288
NCI330	0.320	0.320	0.321	**0.341**	0.301	0.302	0.317	0.309
NCI41	0.353	0.353	0.347	**0.355**	0.316	0.314	0.346	0.306
NCI47	0.302	0.303	0.296	**0.305**	0.277	0.271	0.295	0.263
NCI81	0.288	0.288	0.284	**0.294**	0.263	0.266	0.284	0.253
NCI83	0.303	0.302	0.303	**0.316**	0.280	0.272	0.301	0.276
H1	0.268	0.265	0.263	**0.274**	0.258	0.214	0.264	0.230
H2	**0.645**	0.643	0.634	0.641	0.581	0.577	0.636	0.567
A1	0.180	**0.207**	0.188	0.203	0.178	0.185	0.195	0.186
H3	0.634	**0.635**	0.630	0.632	0.610	0.603	0.631	0.601
D1	**0.377**	0.369	0.356	0.374	0.354	0.342	0.357	0.329
D2	0.577	0.586	**0.604**	0.583	0.551	0.545	0.592	0.572
D3	0.504	0.501	0.509	**0.534**	0.493	0.486	0.506	0.491
D4	0.466	0.471	0.480	**0.490**	0.445	0.434	0.470	0.443
P1	0.597	**0.610**	0.608	0.598	0.572	0.563	0.599	0.576
P2	0.498	0.505	**0.507**	0.500	0.492	0.486	0.500	0.497
P3	0.567	0.574	**0.587**	0.553	0.552	0.540	0.582	0.559
P4	0.624	0.632	0.628	**0.653**	0.620	0.617	0.625	0.611
C1	0.810	0.811	0.805	0.818	0.812	**0.820**	0.815	0.819
M1	0.432	0.434	**0.444**	0.438	0.417	0.423	0.439	0.440
M2	0.610	0.605	0.607	**0.616**	0.584	0.592	0.606	0.608
M3	**0.788**	0.775	0.774	0.777	0.758	0.754	0.773	0.750
ARQB1	0.970	0.976	0.973	0.990				
ARQB2					0.951	0.940	0.994	0.942
ARQB3	0.965	0.970	0.967	0.986	0.923	0.912	0.965	0.914

Best performing scheme(s) for each classification problem is shown in bold. ARQB1 is the ARQB using Tanimoto-based kernels only, ARQB2 is ARQB using RBF-based kernels only and ARQB3 is the ARQB calculated using both Tanimoto- and RBF-based kernels.

6.6 Comparison with Previously Developed Descriptor Spaces

6.6.1 Classification Performance

To compare the classification performance of the AF descriptor space against the classification performance of the four previously developed descriptor spaces (fp-8192, MK, CT, and FS) and the TF and PF subsets of AF (described in Section 4) we performed a series of experiments in which we used the various kernels described in Section 5 to classify the various datasets. Table 7 and 8 show the ROC50 results achieved by the best kernels for each descriptor space. In addition, Table 9 shows weather or not these schemes achieve ROC50 results that are statistically different from each other. The results for AF, TF, and PF were obtained for fragments up to length seven.

These results show that the AF descriptors lead to ROC50 results that are statistically better than that achieved by all other previously developed schemes, for both the Tanimoto and RBF-based kernels. In addition, the performance achieved by both TF and PF is also good and in general better than that achieved by the earlier approaches.

Comparing between fp-8192, CT, MK, and FS, we can see that the fingerprint-based descriptors achieve the best overall results, whereas MK and CT tend to perform the worst. However, from a statistical significance standpoint CT, MK, and FS are equivalent.

Another interesting observation is that the PF scheme achieves better results than fp-8192 (even though the difference is not significant at $p = 0.01$ but it is at $p = 0.05$). Since the fp-8192 descriptors were also generated by enumerating paths of length up to seven (and also cycles), the performance difference suggests that the folding that takes place due to the fingerprint's hashing approach negatively impacts the classification performance.

Finally, comparing Tanimoto- with RBF-based kernels, we can see that the former does better and these differences are in general statistically significant at $p = 0.01$.

Table 7: ROC50 values for the seven descriptors using kernels derived from Tanimoto.

Datasets	AF	TF	PF	fp-8192	CT	MK	FS
	($\mathcal{K}_f^*$)	($\mathcal{K}_f^*$)	($\mathcal{K}_b$)	($\mathcal{K}_b$)	($\mathcal{K}_f$)	($\mathcal{K}_f$)	($\mathcal{K}_b^*$)
NCI1	**0.317**	0.314	0.309	0.277	0.266	0.231	0.263
NCI109	**0.296**	0.293	0.287	0.269	0.235	0.225	0.238
NCI123	**0.262**	0.255	0.253	0.242	0.228	0.219	0.240
NCI145	**0.334**	0.333	0.323	0.278	0.270	0.232	0.265
NCI167	**0.065**	0.060	0.063	0.060	0.047	0.049	0.054
NCI220	0.238	0.250	0.241	0.258	0.208	**0.441**	0.217
NCI33	**0.318**	0.311	0.306	0.260	0.243	0.220	0.251
NCI330	**0.341**	0.321	0.319	0.329	0.315	0.178	0.242
NCI41	0.355	**0.357**	0.345	0.310	0.275	0.251	0.300
NCI47	0.305	**0.306**	0.296	0.268	0.235	0.228	0.243
NCI81	**0.294**	0.289	0.291	0.262	0.238	0.232	0.239
NCI83	**0.316**	0.315	0.304	0.274	0.262	0.229	0.267
H1	**0.274**	0.270	0.266	0.258	0.232	0.224	0.228
H2	**0.641**	0.638	**0.641**	0.600	0.571	0.562	0.581
A1	**0.203**	0.183	0.183	0.138	0.138	0.134	0.147
H3	0.632	0.630	**0.637**	0.614	0.599	0.586	0.576
D1	0.374	**0.387**	0.374	0.368	0.311	0.318	0.327
D2	**0.583**	0.550	0.573	**0.583**	0.547	0.559	0.507
D3	**0.534**	0.522	0.493	0.500	0.460	0.440	0.474
D4	**0.490**	0.477	0.461	0.461	0.439	0.391	0.399
P1	**0.598**	0.591	0.592	0.576	0.558	0.569	0.546
P2	0.500	0.508	0.501	**0.537**	0.499	0.526	0.459
P3	0.553	0.539	**0.571**	0.569	0.506	0.544	0.552
P4	**0.653**	0.622	0.621	0.566	0.554	0.558	0.590
C1	0.818	0.816	0.816	**0.829**	0.751	0.793	0.818
M1	0.438	0.419	0.425	**0.453**	0.347	0.413	0.409
M2	**0.616**	0.586	0.595	0.600	0.490	0.592	0.604
M3	0.777	0.782	0.782	0.777	0.745	0.789	**0.801**
ARQB	0.975	0.956	0.950	0.909	0.829	0.827	0.846

Best performing scheme(s) for each classification problem is shown in bold. AF refers to Acyclic fragments, TF to Tree fragments, PF to Path fragments, fp-8192 refers to fingerprints of length 8192 bits, CT to Cycles & Trees, MK to Maccs keys, and finally FS to frequent substructures.

6.6.2 Retrieval Performance

We also compare the effectiveness of the different descriptor spaces for the task that is commonly referred to as a database screening [35]. The goal of this is given a compound that has been experimentally determined to be active, find other compounds from a database that are active as well. Since the activity of a chemical compound depends on its molecular structure, and compounds with similar molecular structure tend to have similar chemical function, this task essentially maps to ranking the compounds in the database based on how similar they are to the *query* compound.

In our experiments, for each dataset we used each of its active compounds as a query and evaluated the extent to which the various descriptor spaces along with the kernel functions studied in this paper lead to similarity measures that can successfully retrieve the other active compounds.

As it was with the study presented in the previous section, our experimental evaluation was comprehensive using all possible combinations of descriptor spaces and kernel func-

Table 8: ROC50 values for the seven descriptors using kernels derived from RBF.

Datasets	AF	TF	PF	fp-8192	CT	MK	FS
	$(\mathcal{K}_b^*)$	$(\mathcal{K}_b^*)$	$(\mathcal{K}_b^*)$	$(\mathcal{K}_b)$	$(\mathcal{K}_b)$	$(\mathcal{K}_f)$	$(\mathcal{K}_b)$
NCI1	**0.305**	0.302	0.303	0.198	0.256	0.192	0.249
NCI109	0.292	**0.293**	0.288	0.199	0.228	0.202	0.232
NCI123	0.247	0.240	**0.249**	0.177	0.223	0.173	0.234
NCI145	**0.322**	0.321	0.317	0.203	0.255	0.194	0.258
NCI167	**0.062**	**0.062**	0.053	0.043	0.041	0.043	0.047
NCI220	0.263	0.266	0.261	0.272	0.218	**0.393**	0.198
NCI33	**0.304**	0.297	0.286	0.186	0.238	0.210	0.242
NCI330	**0.317**	0.306	0.311	0.235	0.305	0.241	0.241
NCI41	**0.346**	0.344	0.344	0.237	0.267	0.213	0.294
NCI47	**0.295**	0.271	0.289	0.194	0.232	0.186	0.227
NCI81	0.284	0.279	**0.286**	0.188	0.230	0.194	0.231
NCI83	**0.301**	0.298	0.300	0.197	0.258	0.204	0.253
H1	0.264	0.259	**0.265**	0.229	0.223	0.233	0.220
H2	**0.636**	0.629	0.635	0.573	0.556	0.545	0.575
A1	0.195	0.167	**0.212**	0.125	0.128	0.062	0.123
H3	0.631	0.628	**0.632**	0.578	0.589	0.584	0.554
D1	0.357	**0.362**	0.358	0.345	0.317	0.340	0.307
D2	**0.592**	0.571	0.545	0.567	0.558	0.551	0.486
D3	0.506	0.497	**0.507**	0.430	0.454	0.424	0.482
D4	**0.470**	0.458	0.460	0.401	0.426	0.380	0.400
P1	0.599	**0.604**	**0.604**	0.544	0.542	0.563	0.553
P2	0.500	0.468	0.492	**0.532**	0.493	0.512	0.465
P3	0.582	0.458	0.580	0.553	0.499	**0.583**	0.558
P4	**0.625**	0.605	0.622	0.542	0.559	0.536	0.594
C1	**0.815**	0.810	0.808	0.794	0.744	**0.815**	0.813
M1	**0.439**	0.414	0.429	0.428	0.343	0.411	0.410
M2	**0.606**	0.573	0.600	0.567	0.484	0.577	0.584
M3	0.773	0.779	0.768	0.785	0.749	**0.788**	0.775
ARQB	0.981	0.951	0.968	0.805	0.835	0.803	0.845

Best performing scheme(s) for each classification problem is shown in bold.

tions. Table 10 and Table 11 show the NHR results achieved by the best kernels for each descriptor space, whereas Table 12 shows the extent to which the relative performance of various schemes are statistically significant.

Comparing these results with those for the classification task shows similar trends with respect to the relative performance of the various descriptor spaces. For both Tanimoto- and RBF-based kernels AF statistically outperforms the previously developed schemes. The only exception is with respect to the CT descriptor space and RBF for which AF's higher average performance is not statistically significant at $p = 0.01$ but it is at $p = 0.05$. Also the average performance of the TF and PF descriptors (as measured by AQRB) is higher than earlier schemes as well.

7 Conclusion & Discussion

In this paper we presented a new class of descriptors for representing molecular graphs that are based on connected acyclic fragments and illustrated their effectiveness for the tasks of building classification models and retrieving active compounds from chemical libraries.

This work was primarily motivated by our desire to understand which aspects of the molecular graph are important in providing effective descriptor-based representations for the above two tasks given the four design choices described in Section 4 (dataset specificity, fragment complexity, preciseness, and coverage) and the fact that no scheme, including AF, leads to a descriptor space that is strictly superior (in terms of what it captures) to the rest of the schemes. Each one of the seven descriptor spaces (AF, TF, PF, fp-n, MK, CT, and FS) make some compromises along at least one of these dimensions. We believe that our experimental results help in providing some answers. Specifically, the results comparing PF and fp-8192, suggest that a precise representation is a key property and helps PF outperform fp-8192 even though the former utilizes only path-based fragments, whereas fp-8192 also uses fragments corresponding to cycles. Similarly, the results comparing AF against FS suggest that the 100% coverage of AF is a critical property as it helps outperform the FS approach, which leads to descriptor spaces with much more complex fragments (i.e., arbitrary connected substructures). Also, the results comparing the schemes that utilize dataset specific fragment discovery approaches against the MK scheme show that relying on pre-identified fragments will lead to lower performance. Finally, the results comparing AF against TF and PF show that everything else being the same, more complex fragments do lead to better results; however, these gains are not substantial.

Table 9: Wilcoxon statistical test for the seven descriptors in Table 7 and Table 8.

	Tanimoto							
	AF	TF	PF	fp-8192	CT	MK	FS	W/E/L
AF		>	>	>	>	>	>	6 / 0 / 0
TF	<		=	=	>	>	>	3 / 2 / 1
PF	<	=		=	>	>	>	3 / 2 / 1
fp-8192	<	=	=		>	>	>	3 / 2 / 1
CT	<	<	<	<		=	=	0 / 2 / 4
MK	<	<	<	<	=		=	0 / 2 / 4
FS	<	<	<	<	=	=		0 / 2 / 4

	RBF							
	AF	TF	PF	fp-8192	CT	MK	FS	W/E/L
AF		>	>	>	>	>	>	6 / 0 / 0
TF	<		=	>	>	>	>	3 / 2 / 1
PF	<	=		>	>	>	>	3 / 2 / 1
fp-8192	<	<	<		=	=	=	0 / 2 / 4
CT	<	<	<	=		=	=	0 / 2 / 4
MK	<	<	<	=	=		=	0 / 2 / 4
FS	<	<	<	=	=	=		0 / 2 / 4

The sign '$>$' denotes that row outperforms column descriptor, '$<$' denotes that column outperforms row descriptor and '$=$' denotes that row and column descriptors are statistically indistinguishable. W/E/L is Wins, Equal, and Losses for each scheme.

The work in this paper has been primarily focused on classification approaches based on descriptor spaces. However, another approach was recently investigated by Kashima *et al* [18] that uses a random-walk based approach to directly construct a kernel function between two graphs. The experiments presented in [18] showed promising results (even though they are worse than those reported in this paper for the common datasets), and we believe that such direct graph kernels coupled with information as to what aspects of the molecular graphs are important, can potentially lead to effective classification algorithms.

Finally, the fact that acyclic fragments, and tree fragments in particular, can be useful in classifying chemical compounds, has been known for quite a while. Palyulin and his collaborators [31, 38] used certain types of tree fragments for classification and reported good results for QSAR and QSPR prediction problems.

Table 10: NHR for $k = 10$ using kernels derived from Tanimoto

Datasets	AF	TF	PF	fp-8192	CT	MK	FS
	($\mathcal{K}_f$)	($\mathcal{K}_b$)	($\mathcal{K}_b^*$)	($\mathcal{K}_b$)	($\mathcal{K}_b$)	($\mathcal{K}_f$)	($\mathcal{K}_b$)
NCI1	**0.493**	0.477	0.479	0.467	0.400	0.438	0.443
NCI109	**0.481**	0.473	0.467	0.457	0.378	0.435	0.439
NCI123	**0.448**	0.438	0.440	0.431	0.367	0.284	0.408
NCI145	**0.751**	0.737	0.731	0.683	0.668	0.678	0.686
NCI167	**0.704**	0.676	0.690	0.679	0.675	0.656	0.690
NCI220	0.328	0.335	0.328	0.310	0.329	**0.445**	0.328
NCI33	**0.436**	0.423	0.429	0.416	0.346	0.391	0.379
NCI330	**0.512**	0.479	0.492	0.507	0.437	0.435	0.436
NCI41	**0.476**	0.469	0.473	0.455	0.378	0.453	0.443
NCI47	**0.491**	0.485	0.474	0.457	0.388	0.452	0.384
NCI81	**0.483**	0.471	0.476	0.465	0.393	0.369	0.438
NCI83	**0.477**	0.472	0.470	0.461	0.390	0.335	0.444
H1	0.366	**0.367**	0.358	0.351	0.352	0.304	0.326
H2	0.560	**0.566**	0.560	0.511	0.487	0.528	0.466
A1	**0.685**	0.677	0.682	0.682	0.683	0.660	0.680
H3	0.624	**0.631**	0.628	0.616	0.612	0.576	0.570
D1	**0.219**	0.217	0.210	0.189	0.214	0.213	0.213
D2	0.325	**0.345**	0.311	0.342	0.343	0.316	0.338
D3	0.415	0.417	0.401	0.404	0.403	**0.421**	0.387
D4	0.493	0.504	0.484	0.485	**0.521**	0.455	0.476
P1	0.440	**0.487**	0.427	0.442	0.486	0.415	0.433
P2	0.353	**0.420**	0.324	0.338	0.391	0.399	0.350
P3	0.363	0.400	0.358	0.435	**0.452**	0.404	0.342
P4	0.515	0.542	0.508	0.398	**0.557**	0.488	0.491
C1	0.650	0.647	0.648	0.659	**0.675**	0.636	0.536
M1	**0.387**	0.379	0.380	0.382	0.307	0.369	0.328
M2	0.422	0.408	**0.447**	0.413	0.297	0.400	0.394
M3	0.520	0.453	**0.557**	0.501	0.426	0.508	0.508
ARQB	0.961	0.961	0.946	0.926	0.883	0.893	0.886

Best performing scheme(s) for each classification problem is shown in bold.

Table 11: NHR for $k = 10$ using kernels derived from RBF

Datasets	AF	TF	PF	fp-8192	CT	MK	FS
	($\mathcal{K}_b$)	($\mathcal{K}_b$)	($\mathcal{K}_b$)	($\mathcal{K}_b$)	($\mathcal{K}_b^*$)	($\mathcal{K}_f$)	($\mathcal{K}_b^*$)
NCI1	**0.477**	0.472	0.474	0.464	0.404	0.439	0.430
NCI109	**0.470**	0.468	0.463	0.455	0.375	0.434	0.436
NCI123	0.431	**0.434**	**0.434**	0.430	0.373	0.287	0.415
NCI145	**0.733**	0.729	0.723	0.703	0.677	0.685	0.676
NCI167	**0.686**	0.668	0.677	0.672	0.639	0.639	0.676
NCI220	0.340	0.338	0.337	0.327	0.335	**0.425**	0.361
NCI33	**0.423**	**0.423**	0.419	0.414	0.334	0.386	0.395
NCI330	0.488	0.476	0.490	**0.506**	0.435	0.404	0.434
NCI41	**0.469**	0.464	0.464	0.454	0.379	0.447	0.444
NCI47	**0.480**	0.479	0.469	0.455	0.400	0.451	0.391
NCI81	**0.471**	0.467	0.468	0.463	0.393	0.374	0.441
NCI83	**0.469**	0.467	0.465	0.460	0.392	0.342	0.441
H1	**0.365**	**0.365**	0.358	0.350	0.360	0.319	0.332
H2	0.567	**0.568**	0.559	0.509	0.506	0.494	0.476
A1	0.666	0.671	0.676	0.677	**0.682**	0.632	0.679
H3	0.629	**0.634**	0.632	0.615	0.620	0.624	0.564
D1	0.218	0.218	0.209	0.197	**0.234**	0.181	0.201
D2	0.328	0.344	0.301	0.338	0.348	**0.358**	0.339
D3	0.407	0.416	0.401	0.406	**0.442**	0.414	0.379
D4	0.494	**0.504**	0.488	0.486	0.496	0.491	0.465
P1	0.453	0.487	0.460	0.448	**0.511**	0.419	0.433
P2	0.345	0.428	0.340	0.343	**0.390**	0.330	0.344
P3	0.364	0.391	0.367	0.349	**0.472**	0.351	0.340
P4	0.511	0.541	0.528	0.504	**0.626**	0.482	0.503
C1	0.635	0.646	0.629	0.659	0.633	**0.709**	0.536
M1	0.394	0.381	0.385	0.389	0.306	0.370	**0.399**
M2	0.435	0.406	**0.449**	0.419	0.293	0.400	0.394
M3	0.518	0.458	**0.557**	0.503	0.400	0.515	0.531
ARQB	0.947	0.954	0.942	0.926	0.888	0.881	0.890

Best performing scheme(s) for each classification problem is shown in bold.

Acknowledgment

The authors will like to thank Dr. Ian Watson from Lilly Research Laboratories and Dr. Peter Henstock from Pfizer Inc. for the numerous discussions on the practical aspects of virtual screening. The authors would also like to thank Minnesota Supercomputing Institute for the access to their resources.

References

[1] G. W. Adamson, J. Cowell, M. F. Lynch, A. H. McLure, W. G. Town, and A. M. Yapp. Strategic considerations in the design of a screening system for substructure searches of chemical structure file. *Journal of Chemical Documentation*, 1973.

[2] Jurgen Bajorath. Integration of virtual and high throughput screening. *Nature Review Drug Discovery*, 2002.

[3] John M. Barnard, Geoffery M. Downs, and Peter Willet. Descriptor-based similarity measures for screening chemical databases. *Virtual Screening for Bioactive Molecules*, 2000.

[4] Michael R. Berthold and Christian Borgelt. Mining molecular fragments: Finding relevant substructures of molecules. *Proc. of the ICDM*, 2002.

[5] H.J. Bohm and G. Schneider. Virtual screening for bioactive molecules. *Wiley-VCH, 2000.*

[6] Gianpaolo Bravi, Emanuela Gancia, Darren Green, V.S. Hann, and M. Mike. Modelling structure-activity relationship. *Virtual Screening for Bioactive Molecules*, 2000.

[7] R. Brown and Y. Martin. Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection. *Journal of Chemical Information and Modeling*, 36(1):576–584, 1996.

[8] Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050, 2005.

[9] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of Chemical Information and Modeling*, 42(6):1273–1280, 2002.

[10] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computational Chemistry*, 20:25–33, 1996.

[11] Chemical Computing group Inc. (http://www.chemcomp.com).

[12] Christoph Helma, Tobias Cramer, Stefan Kramer, and Luc De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *Journal of Chemical information and Computer Science*, 44(4):1402–1411, 2004.

[13] Tamas Horvath, Thomas Grtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167, 2004.

[14] ChemAxon Inc. www.chemaxon.com.

[15] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *Proc. of The 4th European Conf. on Principles and Practice of Knowledge Discovery in Databases PKDD'00*, pages 13–23, Lyon.

[16] Bland J.M. An introduction to medical statistics. *(1995) 2nd edn. Oxford University Press*.

[17] T. Joachims. Advances in kernel methods: Support vector learning, making large-scale svm learning practical. *MIT-Press, 1999.*

[18] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs,. *In Proc. 20th International Conference on Machine Learning.*, 2003.

Table 12: Wilcoxon statistical test for the four schemes in Table 10 and Table 11

				Tanimoto				
	AF	TF	PF	fp-8192	CT	MK	FS	W/E/L
AF		=	=	>	>	>	>	4 / 2 / 0
TF	=		=	=	>	>	>	3 / 3 / 0
PF	=	=		=	=	>	>	2 / 4 / 0
fp-8192	<	=	=		=	=	>	1 / 3 / 2
CT	<	<	=	=		=	=	0 / 3 / 3
MK	<	<	<	=	=		=	0 / 3 / 3
FS	<	<	<	<	=	=		0 / 2 / 4

				RBF				
	AF	TF	PF	fp-8192	CT	MK	FS	W/E/L
AF		=	=	>	=	>	>	3 / 3 / 0
TF	=		=	=	>	>	>	3 / 3 / 0
PF	=	=		>	=	>	>	3 / 3 / 0
fp-8192	<	=	<		=	>	>	2 / 2 / 2
CT	=	<	=	=		=	=	0 / 5 / 1
MK	<	<	<	<	=		=	0 / 2 / 4
FS	<	<	<	<	=	=		0 / 2 / 4

The sign '$>$' denotes that row scheme outperforms column scheme, '$<$' denotes that column scheme outperforms row scheme and '$=$' denotes that row scheme and column scheme are statistically indistinguishable. W/E/L is Wins, Equal and Losses for each scheme.

[19] L. Kier and L. Hall. Molecular structure description. *ic Press, 1999.*

[20] S. Kramer, L. De Raedt, and C. Helma. Molecular feature mining in hiv data. *7th International Conference on Knowledge Discovery and Data*, 2001.

[21] Michihiro Kuramochi and George Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1038–1051, 2004.

[22] Andrew R. Leach. Molecular modeling: Principles and applications. *Prentice Hall, Englewood Cliffs, NJ, 2001.*

[23] OpenBabel. (http://openbabel.sourceforge.net).

[24] The PubChem Project. pubchem.ncbi.nlm.nih.gov.

[25] Graham W. Richards. Virtual screening using grid computing: the screensaver project. *Nature Reviews: Drug Discovery*, 1:551–554, July 2002.

[26] The Aids Antiviral Screen. (http://dtp.nci.nih.gov).

[27] A. Srinivasan, R. D. King, S. H. Muggleton, and M. Sternberg. The predictive toxicology evaluation challenge. *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1–6, 1997.

[28] S. Joshua Swamidass, Jonathan Chen, Jocelyne Bruand, Peter Phung, Liva Ralaivola, and Pierre Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(1):359–368, 2005.

[29] Daylight Inc. Mission Viejo CA USA. (http://www.daylight.com).

[30] MDL Information Systems Inc. San Leandro CA USA. (http://www.mdl.com).

[31] Palyulin V.A., Baskin I.I., Petelin D.E., and Zefirov N.S. Novel descriptors of molecular structure in qsar and qspr studies. *QSAR and Molecular Modeling: Concepts, Computational tools and Biological Applications*, pages 51–52, Sanz F.

[32] V. Vapnik. Statistical learning theory. *John Wiley, New York, 1998.*

[33] Michal Vieth, Miles G Siegel, Richard E. Higgs, Ian A. Watson, Daniel H. Robertson, Kenneth A. Savin, Gregory L. Durst, and Philip A. Hipskind. Characteristic physical properties and structural fragments of marketed oral drug. *Journal of Medicinal Chemistry*, 47(1):224–232, 2004.

[34] Douglas B. West. Introduction to graph theory. *Prentice Hall (2001)*.

[35] Martin Whittle, Valerie J. Gillet, and Peter Willett. Enhancing the effectiveness of virtual screening by fusing nearest neighbor list: A comparison of similarity coefficients. *Journal of Chemical Information and Modeling*, 44:1840–1848, 2004.

[36] Peter Willett. Chemical similarity searching. *Journal of Chemical Information and Modeling*, 38(6):983–996, 1998.

[37] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. *ICDM*, 2002.

[38] Nikolai S. Zefirov and Vladimir A. Palyulin. Fragmental approach in qspr. *Journal of Chemical Information and Modeling*, 42(5):1112–1122, 2002.