Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 07-024

Indirect Similarity based Methods for Effective Scaffold-Hopping in Chemical Compounds

Nikil Wale, Ian A. Watson, and George Karypis

October 15, 2007

Indirect Similarity based Methods for Effective Scaffold-Hopping in Chemical Compounds

Nikil Wale and George Karypis
Department of Computer Science,
University of Minnesota, Twin Cities
{nwale,karypis}@cs.umn.edu

Ian A. Watson
Eli Lilly and Company
Lilly Research Labs, Indianapolis
watson_ian_a@lilly.com

Abstract

Methods that can screen large databases to retrieve a structurally diverse set of compounds with desirable bioactivity properties are critical in the drug discovery and development process. This paper presents a set of such methods that are designed to find compounds that are structurally different to a certain query compound while retaining its bioactivity properties (scaffold hops). These methods utilize various indirect ways of measuring the similarity between the query and a compound that take into account additional information beyond their structure-based similarities. The set of techniques that are presented capture these indirect similarities using approaches based on analyzing the similarity network formed by the query and the database compounds. Experimental evaluation shows that most of these methods substantially outperform previously developed approaches both in terms of their ability to identify structurally diverse active compounds as well as active compounds in general.

Keywords: *descriptor-space, ranked-retrieval, scaffold-hopping, virtual screening.*

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 activity against the specific biomolecular target [22]. This represents a significant hurdle in the early stages of drug discovery.

A popular method for finding these hits is to use a structure-based ranked-retrieval approach. In this approach, an active compound is used as a *query* to identify and rank the compounds of a large database based on how similar their structures is with that of the query. This approach relies on the structure activity relationship (SAR) [9] principle, which states that that compounds sharing key structural features will most likely have similar activity against a biomolecular target. The structural similarity between the compounds is usually computed by first representing their molecular graph as a vector in a particular *descriptor-space* and then using a variety of vector-based methods to compute their similarity [8,9].

The task of identifying hit compounds is complicated by the fact that the query might have undesirable properties such as toxicity, bad ADME (absorption, distribution, metabolism and excretion) properties, or may be promiscuous [18, 24]. These properties will also be shared by most of the highest ranked compounds as they will correspond to very similar structures. In order to overcome this problem, it is important to identify (i.e., rank high) as many chemical compounds as possible that not only show the desired activity for the biomolecular target but also have different structures (come from diverse chemical classes or chemotypes). Finding novel chemotypes using the information of already known bioactive small molecules is termed as *scaffold-hopping* [18, 27, 32].

In this paper we address the problem of scaffold-hopping by developing a set of techniques that measure the similarity between the query and a compound by taking into account additional information beyond their structure-based similarities. This *indirect* way of measuring similarity enables the retrieval of compounds that are structurally different from the query but at the same time possess the desired bioactivity properties. The set of techniques we present derive the indirect similarities by analyzing the similarity network formed by the query

and the database compounds. All of these techniques operate on the descriptor-space representation of the compounds and are independent of the selected descriptor-space.

We experimentally evaluate the performance of the proposed methods using three different descriptor-spaces and six different datasets. Our results show that most of these methods are quite effective in improving the scaffold-hopping performance over the standard retrieval schemes and that they substantially outperform schemes that were specifically developed for addressing the scaffold-hopping problem. Moreover, even though these methods were created to improve the scaffold-hopping performance, our results show that they are quite effective in improving the overall ranked-retrieval performance as well (i.e., rank the active compounds much higher than the inactive).

The rest of the paper is organized as follows. Section 2 describes the problems addressed in this paper. Section 3 introduces the definitions and notations used in this paper. Section 4 introduces the various descriptor-spaces for this problem. Section 5 describes the methods developed in this paper. Section 6 gives an overview of the related work in this field. Section 7 describes the materials used in our experimental methodology. Section 8 compares the results obtained and finally, Section 9 summarizes the results of this paper.

2 Problem Statement and Motivation

The ranked-retrieval and the scaffold-hopping problems that we consider in this paper are defined as follows:

Definition 1 (Ranked-Retrieval Problem). *Given a query compound, rank the compounds in the database based on how similar they are to the query in terms of their bioactivity.*

Definition 2 (Scaffold-Hopping Problem). *Given a query compound and a parameter k, retrieve the k compounds that are similar to the query in terms of their bioactivity but their structure is as dissimilar as possible to that of the query.*

The solution to the ranked-retrieval problem relies on the well known fact that the chemical structure of a compound relates to its activity (SAR) [9]. As such, effective solutions can be devised that rank the compounds in the database based on how structurally similar they are to the query.

However, for scaffold-hopping, the compounds retrieved must be structurally *sufficiently* similar to possess similar bioactivity but at the same time must be structurally *dissimilar* enough to be a novel chemotype. This is a much harder problem than simple ranked-retrieval as it has the additional constraint of maximizing dissimilarity that runs counter to SAR.

Methods that have the ability to rank higher the compounds that are structurally different (different chemotypes) have advantages over methods that do not. They improve the odds of being able to find a compound that is not only active for a biomolecular target but also has all the other desired properties (non-toxicity, good ADME properties, target specificity, etc. [8, 18]) that the reference structure and compounds with similar structures might not possess. One of such compounds is then more likely to become a true drug candidate. Furthermore, scaffold-hopping is also important from the point of view of un-patented chemical space. Many important lead compounds and drug candidates have been already patented. In order to find new therapies and offer alternative treatments it is important for a pharmaceutical company to discover novel leads significantly different from the existing patented chemical space. Methods that perform scaffold-hopping can achieve those objectives.

3 Definitions and Notations

Throughout the paper we will use D to denote a database of chemical compounds, q to denote a query compound, and c to denote a chemical compound present in the database. Given two compounds c_i and c_j, represented in a certain descriptor-space, we will use $\mathrm{dsim}(c_i, c_j)$ to denote their *direct* similarity, $\mathrm{isim}(c_i, c_j)$ to denote their *indirect* similarity, and $\mathrm{sim}(c_i, c_j)$ to denote either their direct or indirect similarity. The direct similarity between a pair of compounds is computed by a suitable similarity measure that takes into account their descriptor-space representation, whereas the indirect similarity is computed by taking additional information beyond their descriptor-space representation. Given a compound c_i and a set of compounds A, we will use $\mathrm{sim}(c_i, A)$ to denote the average pairwise similarity between c_i and all the compounds in A. Given a query compound q, a database D, and a parameter k, we define top-k to be the k compounds in D that are most similar to q. Given a compound c, a set of compounds A, and a similarity measure, its *k-nearest-neighbor list* contains the k compounds in A that are most similar to c. Finally, throughout the paper we will refer to the

task of retrieving active compounds as *ranked-retrieval* and the task of retrieving scaffold-hops as *scaffold-hopping*.

4 Descriptor Spaces for Ranked-Retrieval

The similarity between chemical compounds is usually computed by first transforming them into a suitable descriptor-space representation [8,9]. A number of different approaches have been developed to represent each compound by a set of descriptors. These descriptors can be based on physiochemical properties as well as topological and geometric substructures (fragments) [1,3,12,19,23,29,31].

In this study we use three descriptor-spaces that have been shown to be very effective in the context of ranked-retrieval and/or scaffold-hopping. These descriptor-spaces are the graph fragments (GF) [29], extended connectivity fingerprints (ECFP) [19, 23], and the extended reduced graph (ErG) descriptors [27].

GF is a 2D topology-based descriptor-space [29] that is based on all the graph fragments of a molecular graph up to a predefined size. The idea behind this descriptor-space is to capture all possible (arbitrary) fragment shapes up to a certain size in a given database in order to characterize this descriptor-space. We use seven as the upper limit on the size of the graph fragments used in this work. ECFP is also a 2D topological descriptor-space and many flavors of these descriptors have been described by several authors [19,23]. The idea behind this descriptor-space is to capture the topology around each atom in the form of shells whose radius (number of bonds) ranges from 1 to l, where l is a user defined parameter. We use the ECZ3 variation of ECFP in which each atom is assigned a label corresponding to its atomic number and the maximum shell radius is set to three. Both extended connectivity fingerprints (ECFP) and GF have been shown to be highly effective for the ranked-retrieval of chemical compounds [19,29].

Extended reduced graph descriptors (ErG) is a pharmacophoric descriptor-space. A pharmacophore is defined as a critical 3D or 2D arrangement of molecular fragments forming a necessary but not sufficient condition for biological activity. The descriptors that rely only on 2D information are called 2D pharmacophoric descriptors whereas descriptors that utilize 3D information are called 3D pharmacophoric descriptors. ErG is a 2D pharmacophoric descriptor-space that combines the reduced graphs [14, 16] and binding property pairs [21] to generate pharmacophoric descriptor-space. A detailed description on the generation of these pharmacophoric descriptors can be found in [27].

5 Methods

To improve the scaffold-hopping performance of ranked-retrieval approaches we developed techniques that are based on measuring the similarity between the query and a compound by taking into account additional information beyond their descriptor-space-based representation. These methods are motivated by the observation that if a query compound q is structurally similar to a database compound c_i and c_i is structurally similar to another database compound c_j, then q and c_j could be considered as being similar or related even though they may have zero or very low direct similarity. This *indirect* way of measuring similarity can enable the retrieval of compounds that are structurally different from the query but at the same time, due to associativity, possess the same bioactivity properties with the query.

5.1 Nearest-Neighbor Graph-based Similarity

The set of techniques we develop to capture such indirect similarities were inspired by research in the fields of information retrieval and social network analysis. These techniques derive the indirect similarities by analyzing the network formed by a k-nearest-neighbor graph representation of the query and the database compounds.

The network linking the database compounds with each other *and* with the query is determined by using a *k-nearest-neighbor* (NG) and a *k-mutual-nearest-neighbor* (MG) graph. Both of these graphs contain a node for each of the compounds as well as a node for the query. However, they differ on the set of edges that they contain. In the k-nearest-neighbor graph there is an edge between a pair of nodes corresponding to compounds c_i and c_j, if c_i is in the k-nearest-neighbor list of c_j or vice-versa. In the k-mutual-nearest-neighbor graph, an edge exists only when c_i is in the k-nearest-neighbor list of c_j *and* c_j is in the k-nearest-neighbor list of c_i. As a result of these definitions, each node in NG will be connected to at least k other nodes (assuming that each compound has a non-zero similarity to at least k other compounds), whereas in MG, each node will be connected to at most k other nodes.

Since the neighbors of each compound in these graphs cor-

respond to some of its most structurally similar compounds and due to the relation between structure and activity, each pair of adjacent compounds will tend to have similar activity. Thus, these graphs can be considered as network structures for capturing bioactivity relations.

A number of different approaches have been developed for determining the similarity between nodes in social networks that take into account various topological characteristics of the underlying graphs [13, 28]. In our work, we determine the similarity between a pair of nodes as a function of the intersection of their adjacency lists, which takes into account all two-edge paths connecting these nodes. Specifically, the similarity between c_i and c_j with respect to graph G is given by

$$\text{isim}_G(c_i, c_j) = \frac{\text{adj}_G(c_i) \cap \text{adj}_G(c_j)}{\text{adj}_G(c_i) \cup \text{adj}_G(c_j)}, \quad (1)$$

where $\text{adj}_G(c_i)$ and $\text{adj}_G(c_j)$ are the adjacency lists of c_i and c_j in G, respectively.

This measure assigns a high similarity value to a pair of compounds if both are very similar to a large set of common compounds. Thus compounds that are part of reasonably tight clusters (i.e., a set of compounds whose structural similarity is high) will tend to have high indirect similarities as they will most likely have a large number of common neighbors. In such cases, the indirect similarity measure re-enforces the existing high direct similarities between compounds. However, the indirect similarity between a pair of compounds c_i and c_j can also be high even if their direct similarity is low. This can happen when the compounds in $\text{adj}_G(c_i) \cap \text{adj}_G(c_j)$ match different structural descriptors of c_i and c_j. In such cases, the indirect similarity measure is capable of identifying relatively weak structural similarities, making it possible to identify scaffold-hopping compounds.

Given the above graph-based indirect similarity measures, various strategies can be employed to retrieve compounds from the database. In our work we used three such strategies. The first corresponds to that used by the standard ranked-retrieval method, whereas the other two are inspired by information retrieval methods used for automatic relevance feedback [6] and are specifically designed to improve the scaffold-hopping performance.

5.1.1 Best-Sim Retrieval Strategy

This is the most widely used retrieval strategy and it simply returns the compounds that are the most similar to the query. Specifically, if A is the set of compounds that have been retrieved thus far, then the next compound c_{next} that is selected is given by

$$c_{next} = \arg\max_{c_i \in D-A} \{\text{isim}(c_i, q)\}. \quad (2)$$

This compound is added to A, removed from the database, and the overall process is repeated until the desired number of compounds has been retrieved. Depending on whether the indirect similarities are computed on the NG or MG graph, we will refer to this retrieval strategy as BESTSIMNG or BESTSIMMG, respectively.

5.1.2 Best-Sum Retrieval Strategy

This retrieval strategy incorporates additional information from the set of compounds retrieved thus far (set A). Specifically, the compound selected, c_{next}, is the one that has the highest average similarity to the set $A \cup \{q\}$. That is,

$$c_{next} = \arg\max_{c_i \in D-A} \{\text{isim}(c_i, A \cup \{q\})\}. \quad (3)$$

The motivation behind this approach is that due to SAR, the set A will contain a relatively large number of active compounds. Thus, by modifying the similarity between q and a compound c to also include how similar c is to the compounds in the set A, we obtain a similarity measure that is re-enforced by A's active compounds. This enables the retrieval of active compounds that are similar to the compounds present in A even if their similarity to the query is not very high; thus, enabling scaffold-hopping. We will refer to this best-sum retrieval strategy as BESTSUMNG and BESTSUMMG, corresponding to the NG and MG graph-based similarity measures, respectively.

5.1.3 Best-Max Retrieval Strategy

A key characteristic of the retrieval strategy described in Section 5.1.2 is that the final ranking of each compound is computed by taking into account *all* the similarities between the compound and the compounds in the set A. Since the compounds in A will tend to be structurally similar to the query compound, this approach is rather conservative in its attempt to identify active compounds that are structurally different from the query (i.e., scaffold-hops).

To overcome this problem, we developed a retrieval strategy that is based on the best-sum approach but instead of selecting the next compound based on its average similarity to

the set $A \cup \{q\}$, it selects the compound that is the most similar to *one* of the compounds in $A \cup \{q\}$. That is, the next compound is given by

$$c_{next} = \arg\max_{c_i \in D-A} \{ \max_{c_j \in A \cup \{q\}} \text{isim}(c_i, c_j) \}. \quad (4)$$

In this approach, if a compound c_j other than q has the highest similarity to some compound c_i in the database, c_i is chosen as c_{next} and added to A irrespective of its similarity to q. Thus, the query-to-compound similarity is not necessarily included in every iteration as in the other schemes, allowing this strategy to identify compounds that are structurally different from the query. Depending on whether the indirect similarities are computed on the NG or MG graph, we will refer to this retrieval strategy as BESTMAXNG or BESTMAXMG, respectively.

5.2 Combining Parameters

One of the drawbacks of the NG and MG graph-based schemes is that given a problem, it is not clear what should be the right number of neighbors (k) used in constructing the NG and MG graphs. Moreover, our experience with this approach has shown that the best-performing value of k is to some extent dataset dependent. In order to increase the robustness of the overall approach we introduce two extensions of the above methods that take as input multiple parameter values. In particular, if the number of input parameters is m, then the corresponding NG or MG graph is derived for each of the input parameter values. The indirect similarity values are calculated using Equation 1 for each pair of compounds c_i and c_j corresponding to each of the m graphs. A unique indirect similarity value is then derived by choosing either the sum of all the m similarities or taking the maximum of these values. Thus, for each of the six schemes BESTSIMNG, BESTSUMNG, BESTMAXNG, BESTSIMMG, BESTSUMMG, and BESTMAXMG we have two extensions, one based on summation and the other based on choosing the maximum of all the similarity values. The two extensions of BESTSIMNG are referred to as BESTSIMNGSUM and BESTSIMNGMAX. The notation for the extensions of the other methods are derived in similar fashion.

6 Related Work

Many methods have been proposed for ranked-retrieval and scaffold-hopping. These can be divided into two groups. The first contains methods that rely on better designed descriptor-space representations, whereas the second contains methods that are not specific to any descriptor-space representation but utilize different retrieval strategies to improve the overall performance.

Among the first set of methods, 2D descriptors such as path-based fingerprints [1,4], dictionary based keys [2,3] and more recently Extended Connectivity fingerprints (ECFP) [19], Graph Fragments (GF) [29] have all been successfully applied for the retrieval problem. Pharmacophore based descriptors such as ErG [27] have been shown to outperform simple 2D topology based descriptors for scaffold-hopping [27, 33]. Lastly, descriptors based on 3D structure or conformations of the molecule have also been applied successfully for scaffold-hopping [24, 33].

The second set of methods include the turbo search schemes (TURBOSUMFUSION and TURBOMAXFUSION) [18] and the structural unit analysis based techniques [32] all of which utilize automatic relevance feedback [6] ideas. These have been shown to be effective for both scaffold-hopping and ranked-retrieval. The turbo search techniques operate as follows. Given a query q, they start by retrieving the top-k compounds from the database. Let A be the $(k+1)$-size set that contains q and the top-k compounds. For each compound $c \in A$, all the compounds in the database are ranked in decreasing order based on their similarity to c, leading to $k+1$ ranked lists. These lists are used to obtain the final similarity of each compound with respect to the initial query. In particular, in TURBOMAXFUSION, the similarity between q and a compound c is equal to the similarity corresponding to the maximum similarity of c in the $k+1$ lists, whereas in TURBOSUMFUSION, the similarity is equal to the sum of all the similarities in these rankings. Thus, these techniques utilize the retrieval strategies similar to those described in Section 5.1.3 and 5.1.2. Similar methods based on consensus scoring, rank averaging, and voting have also been investigated in [33].

7 Materials

7.1 Datasets

We used datasets that contain compounds that bind to six different biomolecular targets: COX2 (cyclooxygenase 2), CDK2 (cyclin-dependent kinase 2), FXa (coagulation factor Xa), PDE5 (phosphodiesterase 5), A1A (alpha-1A adrenoceptor), and MAO (Monoamineoxidase). Each of these sets represent a different activity class.

The datasets for the first five targets are obtained from [5, 20]. The entire set consists of 2142 compounds and there are 50 active compounds for each one of the targets (250 in total). The rest of the compounds are "decoys" (inactive) obtained from the National Cancer Institute diversity set. For each target, we constructed a dataset that contains its 50 active compounds and all the decoys. These datasets are termed as COX2, CDK2, PDE5, FXa and A1A. Note that the decoys are untested molecules in this dataset. Therefore, they are not necessarily inactive but rather have low direct similarity to the five activity classes in this dataset.

The dataset of the sixth target was derived from [11, 29] and after removing compounds with impossible Kekule forms and valence errors it contains 1458 compounds. The compounds in this dataset have been categorized into four different classes, 0, 1, 2, and 3 based on their levels of activity, with 0 indicating no activity. For our experiments we treat all the compounds that have non-zero activity level (268 compounds) as active.

7.2 Definition of Scaffold-Hopping Compounds

Molecular scaffold is a widely cited concept and is used to evaluate the performance of a method with respect to its scaffold-hopping ability. However the definition of a scaffold-hop is highly subjective with numerous papers using different criteria to define what constitutes a scaffold-hop [10, 18, 32, 33].

One of the approaches that is used to define and evaluate scaffold-hopping performance of a method is to "cluster" all the active compounds (global clustering) in a database and assign a structural class to each cluster. The scaffold-hopping performance of a method is then evaluated by identifying how many active compounds from different clusters (structural classes) are obtained [27]. The assumption in this approach is that compounds with different scaffolds are more likely to be in different clusters and the method that retrieves actives from many different clusters among the top hits is good for scaffold-hopping. This evaluation strategy has the drawback that using global clustering masks the local effects and the final clusters depend on clustering parameters [10, 26].

Another approach is to use mean pairwise similarity to assess the diversity of a set of actives from a given activity class (target class like Kinase C inhibitors, COX2 inhibitors etc). If a method performs well for the class that has high diversity then it is good for scaffold-hopping [18]. This approach depends on how the diversity is evaluated. Moreover, there is no baseline measure to objectively define absolute low or high diversity class [10]. Other methods rely on carefully chosen active class compounds for a given target such that they belong to different chemotypes (maximum diversity) [32, 33]. Again, the drawback of such approach that it is highly subjective and depends on the view point of the domain expert.

In this paper we define scaffold-hops by using an approach that directly relies on the scaffold-hopping problem definition (Section 3). In particular, for a given query q, the active compounds are ranked based on their structural similarity to q, and the lowest 50% of them are defined to be the scaffold-hops for q. Thus, this approach identifies a set of scaffold-hopping compounds that are specific to each query and represent the 50% most dissimilar active compounds to the query. We use the 2048-bit path-based fingerprint generated by Chemaxon's screen program [4] for measuring the structural similarity between a query and an active compound although any other path-based fingerprint like daylight fingerprints could be used. These fingerprints are well-designed to capture structural similarity between two compounds [27]. By directly defining scaffold-hops as described above, our approach captures compounds that are structurally different from the query and therefore more likely to be a scaffold-hop. We believe that this is a more objective approach to define scaffold-hops for a given compound.

7.3 Experimental Methodology

All the experiments were performed on dual core AMD Opterons with 4 GB of memory. We used the descriptor-spaces GF, ECZ3, and ErG (described in Section 4) for the evaluating the methods introduced in this paper. Each method is tested against six datasets (Section 7.1) using three different descriptor-spaces (Section 4) leading to a total of 18 different

combinations of datasets and descriptor-spaces. We will refer to them as 18 different problems.

We use the Tanimoto similarity [8, 30, 31] for all direct similarity calculations. The Tanimoto similarity function is given by

$$\text{dsim}(c_i, c_j) = \frac{\sum_{k=1}^{n} c_{ik}c_{jk}}{\sum_{k=1}^{n}(c_{ik})^2 + \sum_{k=1}^{n}(c_{jk})^2 - \sum_{k=1}^{n} c_{ik}c_{jk}}, \quad (5)$$

where c_{ik} and c_{jk} are the values for the kth dimension in the n-dimensional descriptor-space representation for the compounds c_i and c_j, respectively. This similarity function was selected because it has been shown to be an effective way of measuring the similarity between chemical compounds [30, 31] for ranked-retrieval and is the most widely-used similarity function in cheminformatics.

For each dataset we used each of its active compounds as a query and evaluated the extent to which the various methods lead to effective retrieval of the other active compounds and scaffold-hops.

We varied the parameter values for the methods described in Section 5 and obtained results for each of the four different sets of values. For the nearest-neighbor methods which depend on the number of neighbors, we used k = 4, 6, 8, and 10 for the BESTSIMNG, BESTSUMNG and BESTMAXNG, and k = 12, 16, 20, and 24 for the BESTSIMMG, BESTSUMMG and BESTMAXMG schemes. For NG the value of k less than 4 leads to graphs with many connected components whereas for MG this value is 12. Hence, we decided not to use values below these thresholds. Note that the threshold for NG is less than that of MG because the criterion for an edge to exist between two nodes of the neighborhood graph is stricter for MG as opposed to NG (Section 5.1). We also derive results for the two parameter combination schemes of the graph-based methods (Section 5.2) using all of these parameter values. For example, BESTSIMNG has six sets of results corresponding to the four parameter values (k = 4, 6, 8, and 10) and the two combination schemes BESTSIMNGSUM and BESTSIMNGMAX. Similarly we derive six sets of results for other schemes as well.

We compared our schemes against TURBOMAXFUSION and TURBOSUMFUSION [18]. For both these methods, we used $k = 5, 10, 15,$ and 20. These values gave the best results and the results degraded as k was further increased. For a particular value of k, for example $k = 5$, we denote these schemes by TURBOMAXFUSION5 and TURBOSUMFUSION5. The notation for the two schemes for other values of parameter k are derived in similar fashion.

7.4 Standard Retrieval

For each problem (dataset and descriptor-space combination), we obtain a baseline performance by ranking all the compounds with respect to each active compound using the standard retrieval strategy described in Section 5.1.1 and the Tanimoto similarity measure (direct similarity measure). We will refer to this scheme as *standard retrieval* and denote it by STDRETTM.

7.5 Performance Assessment Measures

We measure ranked-retrieval and scaffold-hopping performance using *uninterpolated precision* [17]. This is calculated as follows. For each active that appears in the top 50 retrieved compounds we compute the precision value. For ranked-retrieval this is defined as the ratio of the number of actives retrieved over the number of compounds retrieved thus far. For scaffold-hopping it is defined as the number of scaffold-hops retrieved over the number of compounds retrieved thus far. For both ranked-retrieval and scaffold-hopping we sum all their precision values and normalize them by dividing them with 50. This is called the total uninterpolated precision for a query. Similar values are obtained for all the queries for a dataset and the total uninterpolated precision is the average of all these values. Note that the total uninterpolated precision captures the number of active compounds (scaffold-hops) for each query as well as the position (rank) information of the actives (scaffold-hops).

To compare the ranked-retrieval performance of two methods, we evaluate their relative performance over all the 18 problems. This is achieved as follows. Let r_i and q_i represent the ranked-retrieval performance achieved by methods r and q on the ith problem respectively. We calculate the log-ratio, $\log_2(r_i/q_i)$, for every problem and take the average of these values. We call this quantity the *Average Relative Performance* or ARP of r with respect to q. On the average, if the ARP is less than zero, r performs worse than q for the task of ranked-retrieval whereas if the ARP is greater than zero, r performs better than q for that task. Note that the reason that we use log-ratios as opposed to simply the ratios is

that the distribution of the ratios of two random variables is not symmetric whereas the distribution of their log-ratios is normally distributed. This allows us to compute their average and compare them in an unbiased way. We also compare the scaffold-hopping performance of two methods against each other in a similar fashion. We assess whether the ARP for a given pair of methods is statistically significant using the student's t-test [7], which is well-suited to assess statistical significance of a sample of values drawn out of a normal distribution.

8 Results

In this paper we choose to present the best results obtained for each of the methods described Section 5.1 and TURBOSUMFUSION and TURBOMAXFUSION methods described in Section 6 (the complete set of results are available online[1]). For each of the three retrieval strategies (i.e., best-sim, best-sum, and best-max) we only present the results obtained by the best-max combination scheme (Section 5.2) for the NG and MG graphs. This combination method performed quite well for both scaffold-hopping and ranked-retrieval and achieved better results than those obtained by the schemes using a single value of k or those obtained by the best-sum combination method. For the fusion-based schemes, we present the results obtained for $k = 5$ as it gave the best results for the scaffold-hopping and ranked-retrieval task among the values tested.

8.1 Overall Performance Assessment

Tables 1 and 2 compare the performance of all the methods in a pairwise fashion for scaffold-hopping and ranked-retrieval, respectively. In each of these tables we present two statistics. The first is the ARP of the row method (r) with respect to the column method (q) as described in Section 7.5. The second statistic, shown immediately below the ARP value in parenthesis, is its p-value obtained from the student's t-test. Note that for the remainder of this section we will consider the ARP of the two methods to be statistically significant if $p \leq 0.01$.

Comparing the performance of the nearest-neighbor methods we observe that all of them show good performance for scaffold-hopping as well as ranked-retrieval. Among them, the best performing method is BESTSUMMGMAX. It achieves the best balance between the ranked-retrieval and scaffold-hopping performance. Moreover, the results comparing best-sim retrieval strategies (BESTSIMNGMAX and BESTSIMMGMAX) to the corresponding best-sum and best-max retrieval strategies (BESTSUMNGMAX, BESTSUMMGMAX, BESTMAXNGMAX and BESTMAXMGMAX) show that the latter are statistically better in all cases for the scaffold-hopping task. The best-sum and best-max retrieval methods also outperform the best-sim strategy for ranked-retrieval (although most of these differences are not statistically significant). These results indicate that automatic relevance feedback based strategies (best-sum and best-max) improve scaffold-hopping performance over the best-sim retrieval strategy without degrading the ranked-retrieval performance.

Comparing the results obtained by using the nearest-neighbor graphs over those obtained by using the mutual-nearest-neighbor graphs we see that the relative performance of the various retrieval strategies on these two graphs varies. For example, BESTMAXNGMAX is worse than BESTMAXMGMAX for scaffold-hopping but better for ranked-retrieval. However, most of these differences are not statistically significant.

Comparing the scaffold-hopping performance of the graph-based methods with that of STDRETTM (baseline scheme), we see that the former are on average 117% better. For example, BESTSIMMGMAX outperforms STDRETTM by as much as 120%. Note that since the retrieval strategy in both of these methods is the same (best-sim), this particular result indicates that the rather dramatic improvement in performance is a direct consequence of the indirect similarity used by the BESTSIMMGMAX scheme. The performance advantage of the graph-based methods over STDRETTM also carry for the problem of ranked-retrieval for which they always achieve better results (ranging from 3% to 26% better). Moreover, these improvements are statistically significant for the results obtained by the best-sum and best-max retrieval strategies.

Finally, comparing the scaffold-hopping performance of the graph-based methods with that of the two fusion-based schemes, we observe that the graph-based schemes perform consistently better (ranging from 6% to 125% better). In addition, for the best-sum and best-max nearest-neighbor schemes these performance differences are also statistically signifi-

[1] http://bioinfo.cs.umn.edu/supplements/scaffold-hopping/Results_complete.xls

Table 1: Performance for Scaffold-Hopping.

	StdRetTM	TurboSumFusion5	TurboMaxFusion5	BestSimNGMax	BestSumNGMax	BestMaxNGMax	BestSimMGMax	BestSumMGMax	BestMaxMGMax
StdRetTM		-0.52 (0.042)	-0.87 (0.008)	-0.94 (0.001)	-1.62 (0.000)	-1.69 (0.000)	-1.15 (0.001)	-1.82 (0.000)	-1.78 (0.000)
TurboSumFusion5	0.52 (0.042)		-0.34 (0.082)	-0.42 (0.017)	-1.10 (0.000)	-1.17 (0.000)	-0.63 (0.004)	-1.30 (0.000)	-1.25 (0.000)
TurboMaxFusion5	0.87 (0.008)	0.34 (0.082)		-0.08 (0.577)	-0.75 (0.001)	-0.83 (0.000)	-0.28 (0.017)	-0.96 (0.000)	-0.91 (0.000)
BestSimNGMax	0.94 (0.001)	0.42 (0.017)	0.08 (0.577)		-0.67 (0.001)	-0.75 (0.001)	-0.20 (0.069)	-0.88 (0.001)	-0.83 (0.001)
BestSumNGMax	1.62 (0.000)	1.10 (0.000)	0.75 (0.001)	0.67 (0.001)		-0.07 (0.561)	0.47 (0.014)	-0.20 (0.181)	-0.16 (0.404)
BestMaxNGMax	1.69 (0.000)	1.17 (0.000)	0.83 (0.000)	0.75 (0.001)	0.07 (0.561)		0.54 (0.006)	-0.13 (0.327)	-0.09 (0.615)
BestSimMGMax	1.15 (0.001)	0.63 (0.004)	0.28 (0.017)	0.20 (0.069)	-0.47 (0.014)	-0.54 (0.006)		-0.67 (0.001)	-0.63 (0.001)
BestSumMGMax	1.82 (0.000)	1.30 (0.000)	0.96 (0.000)	0.88 (0.001)	0.20 (0.181)	0.13 (0.327)	0.67 (0.001)		0.04 (0.692)
BestMaxMGMax	1.78 (0.000)	1.25 (0.000)	0.91 (0.000)	0.83 (0.001)	0.16 (0.404)	0.09 (0.615)	0.63 (0.001)	-0.04 (0.692)	

The top entry in each cell corresponds to the average of the $\log_2$ ratios of the uninterpolated precision of the row method to the column method for the 18 problems. The number below this entry, in parenthesis, corresponds to the p-value obtained from the student's t-test for that entry.

cant. As it was the case with StdRetTM, the performance advantage of the graph-based schemes over the fusion-based schemes holds for ranked-retrieval as well. However, the gains are somewhat smaller than those achieved for scaffold-hopping.

8.2 Qualitative Comparisons

To illustrate the differences between StdRetTM, TurboMaxFusion5, and BestSumMGMax, Table 3 shows the active compounds that were present in the top-25 compounds retrieved by each of these methods on the A1A dataset and ECZ3 descriptor space using Indoramin as the query compound (compound id 50). For each of the compounds retrieved, Table 3 also shows the chemical class that this compound belongs to. These chemical classes were obtained from literature as described in [20], and are available in the online supplement of [20].

These results show that although all three methods retrieve compounds belonging to classes 3, 6, and 7, the first two methods retrieved far fewer compounds (six and seven actives for StdRetTM and TurboMaxFusion5, respectively) as opposed to BestSumMGMax that retrieved 19 actives. Moreover, BestSumMGMax also retrieves compounds from two additional chemical classes (2 and 11) than the other two methods. The structure of some of these compounds is shown in Figure 1. Note that BestSumMGMax not only retrieves a larger number of active compounds, but it is also able to retrieve active compounds that have much lower direct similarity to the query (e.g., compounds 47, 24, 14, and 8) than the other two schemes. These low similarity compounds tend to have different scaffolds (as illustrated in Figure 1), which explains the improvements in scaffold-hopping that is achieved by BestSumMGMax and the other graph-based schemes. The above trends also holds for most of the other queries and datasets that we analyzed.

8.3 Performance of Descriptor-Spaces

Our discussion so far focused on evaluating the average performance of the different methods across the various descriptor-space representations and datasets. In this section we analyze the performance of the methods on the individual descriptor-spaces and datasets. We limit our evaluation to only the BestSumMGMax method as this methods achieve the best scaffold-hopping and ranked-retrieval performance among the graph-based methods.

The results of these evaluations are shown in Figures 2 and

Table 2: Performance for Ranked-Retrieval.

	StdRetTM	TurboSumFusion5	TurboMaxFusion5	BestSimNGmax	BestSumNGmax	BestMaxNGmax	BestSimMGmax	BestSumMGmax	BestMaxMGmax
StdRetTM		-0.02 (0.576)	0.07 (0.083)	-0.06 (0.362)	-0.23 (0.038)	-0.18 (0.063)	-0.10 (0.150)	-0.27 (0.040)	-0.11 (0.318)
TurboSumFusion5	0.02 (0.576)		0.09 (0.003)	-0.04 (0.471)	-0.21 (0.029)	-0.16 (0.043)	-0.08 (0.195)	-0.25 (0.036)	-0.09 (0.346)
TurboMaxFusion5	-0.07 (0.083)	-0.09 (0.003)		-0.13 (0.085)	-0.30 (0.008)	-0.26 (0.011)	-0.18 (0.030)	-0.34 (0.012)	-0.19 (0.119)
BestSimNGmax	0.06 (0.362)	0.04 (0.471)	0.13 (0.085)		-0.17 (0.005)	-0.12 (0.030)	-0.04 (0.294)	-0.20 (0.015)	-0.05 (0.431)
BestSumNGmax	0.23 (0.038)	0.21 (0.029)	0.30 (0.008)	0.17 (0.005)		0.05 (0.257)	0.13 (0.059)	-0.04 (0.569)	0.12 (0.065)
BestMaxNGmax	0.18 (0.063)	0.16 (0.043)	0.26 (0.011)	0.12 (0.030)	-0.05 (0.257)		0.08 (0.233)	-0.08 (0.305)	0.07 (0.289)
BestSimMGmax	0.10 (0.150)	0.08 (0.195)	0.18 (0.030)	0.04 (0.294)	-0.13 (0.059)	-0.08 (0.233)		-0.16 (0.020)	-0.01 (0.867)
BestSumMGmax	0.27 (0.040)	0.25 (0.036)	0.34 (0.012)	0.20 (0.015)	0.04 (0.569)	0.08 (0.305)	0.16 (0.020)		0.15 (0.012)
BestMaxMGmax	0.11 (0.318)	0.09 (0.346)	0.19 (0.119)	0.05 (0.431)	-0.12 (0.065)	-0.07 (0.289)	0.01 (0.867)	-0.15 (0.012)	

The top entry in each cell corresponds to the average of the $\log_2$ ratios of the uninterpolated precision of the row method to the column method for the 18 problems. The number below this entry, in parenthesis, corresponds to the p-value obtained from the student's t-test for that entry.

3, which compare the performance of StdRetTM and TurboMaxFusion5 respectively against BestSumMGmax. In these figures, the left Y-axis represents uninterpolated precision values for ranked-retrieval, whereas the right Y-axis represents uninterpolated precision values for scaffold-hopping.

The results in Figure 2 show that for scaffold-hopping, BestSumMGmax performs consistently better than StdRetTM for all the descriptor-space and dataset combinations. However, the actual gains are dataset and descriptor-space dependent. For example, the gains are particularly high for the FXa, A1A, and COX2 datasets using the ErG descriptor-space and for FXa, COX2 and PDE5 datasets using the ECZ3 descriptor-space. Similar trends can be observed with the ranked-retrieval results, with BestSumMGmax outperforming StdRetTM in most of the 18 problems with FXa, COX2 and A1A showing maximum gains and only MAO dataset showing a significant performance degradation in terms of ranked-retrieval task in all the three descriptor-spaces.

Finally, the results in Figure 3 show that for scaffold-hopping, BestSumMGmax outperforms TurboMaxFusion5 in most dataset and descriptor-space combinations. Moreover, the trends are similar to than of Figure 2, in that, BestSumMGmax is particularly better for the FXa, A1A, and COX2 datasets.

9 Discussion and Conclusion

In this paper we focused on the problem of improving the scaffold-hopping performance of the ranked-retrieval systems used for chemical compound databases. We introduced a number of methods that utilize the compounds' nearest neighbor graph to indirectly determine their pairwise similarity and devised various retrieval strategies that combine these indirect similarity measurements with retrieval strategies originally developed for automatic relevance feedback. Our results showed that these nearest-neighbor graph-based methods consistently and quite often substantially outperform previously developed rank-retrieval and scaffold-hopping approaches.

Our results suggest that the nearest-neighbor graph formed by the query and the compounds in the database contains additional information on how the various compounds are related that goes well beyond the information that can be extracted by looking at their direct similarities and the set of the structural descriptors that they share. We believe that this

Table 3: Number of actives retrieved in top 25 ranks using different methods for Indoramin (class 13) as query.

# of pos	StdRetTM				TurboMaxFusion5				BestSumMGMax				
	rank	id	dsim	class	rank	id	dsim	class	rank	id	dsim	class	isim
1	3	30	0.8494	7	4	30	0.8494	7	2	16	0.8204	3	0.2222
2	7	35	0.8320	9	7	16	0.8204	3	3	15	0.8100	3	0.1892
3	12	29	0.8264	7	8	29	0.8264	7	4	49	0.8023	12	0.2353
4	14	16	0.8204	3	9	15	0.8100	3	5	41	0.7768	11	0.1818
5	16	25	0.8185	6	11	25	0.8185	6	6	13	0.7798	2	0.2121
6	23	15	0.8100	3	18	49	0.8023	12	7	48	0.7867	12	0.2059
7					19	35	0.8320	9	8	43	0.7759	12	0.1667
8									9	46	0.7852	12	0.1714
9									10	29	0.8264	7	0.2162
10									11	44	0.7597	12	0.0968
11									12	8	0.7129	2	0.1176
12									13	14	0.7050	2	0.1143
13									15	24	0.7024	6	0.1250
14									17	25	0.8185	6	0.2857
15									18	45	0.7838	12	0.1622
16									19	26	0.7295	7	0.0400
17									20	47	0.6953	12	0.0312
18									21	30	0.8494	7	0.2000
19									23	42	0.7141	12	0.0952

The table shows the rank, compound identifier (id), direct Tanimoto similarity (dsim), and the chemical class of compounds retrieved for this query. For BestSumMGMax the table also shows the indirect similarity values (isim) between Indoramin and the actives retrieved.

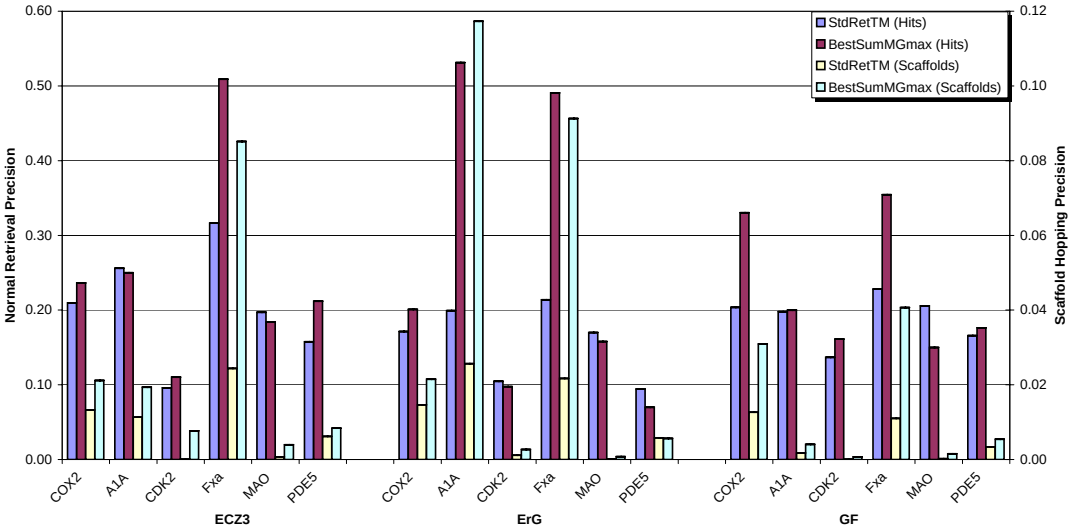

Figure 2: StdRetTM versus BestSumMGmax.

information can be further exploited to obtain additional performance improvements for the ranked-retrieval and scaffold-hopping problems and to also address other problems arising in cheminformatics.

Acknowledgements

This work was supported by NSF ACI-0133464, IIS-0431135, NIH RLM008713A, and by the Digital Technology Center at the University of Minnesota.

References

[1] http://www.daylight.com. *Daylight Inc.*

[2] http://www.digitalchemistry.co.uk/. *Digital Chemistry Inc.*

[3] http://www.mdl.com. *MDL Information Systems Inc.*

[4] www.chemaxon.com. *ChemAxon Inc.*

[5] www.cheminformatics.org. *Cheminformatics*.

[6] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern information retrieval. *Addison Wesley 1999.*

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

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

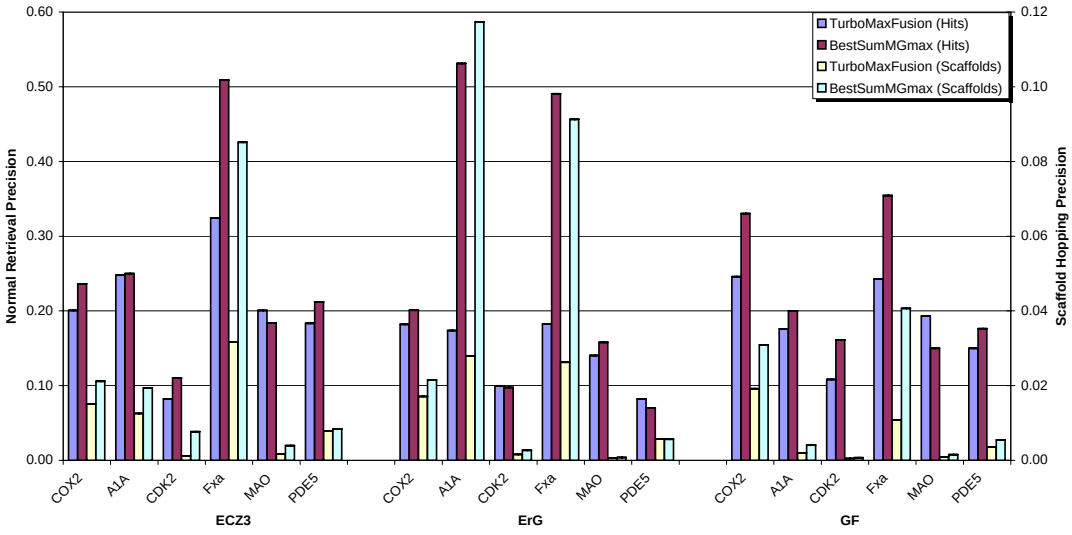

Figure 3: TurboMaxFusion5 versus BestSumMGmax.

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

[10] N. Brown and E. Jacoby. On scaffolds and hopping in medicinal chemistry. *Mini Rev Medicinal Chemistry*, 6(11):1217–1229, 2006.

[11] R. Brown and Y. Martin. Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection. *J. Chem. Info. Model.*, 36(1):576–584, 1996.

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

[13] F. Fouss, A. Pirotte, J. Renders, and M. Saerens. Random walk computation of similarities between nodes of a graph with application to collaborative filtering. *IEEE TKDE*, 19(3):355–369, 2007.

[14] V. J. Gillet, P. Willet, and J. Bradshaw. Similarity searching using reduced graphs. *J. Chem. Inf. Comput. Sci.*, 43:338–345, 2003.

[15] C. M. R. Ginn, D. B. Turner, P. Willett, A. M. Ferguson, and T. W. Heritage. Similarity searching in files of three-dimensional chemical structures: Evaluation of the eva descriptor and combination of rankings using data fusion. *J. Chem. Info. Model.*, 37(1):23–37, 1997.

[16] G. Harper, G.S. Bravi, S.D. Pickett, J. Hussain, and D.V. Green. The reduced graph descriptor in virtual screening and data-driven clustering of high-throughput screening data. *J. Chem. Info. Model.*, 44(6):45–56, 2004.

[17] Marti Hearst and Jan Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. *ACM/SIGIR*, 1996.

[18] J. Hert, P. Willet, and D. Wilton. New methods for ligand based virtual screening: Use of data fusion and machine learning to enchance the effectiveness of similarity searching. *J. Chem. Info. Model.*, (46):462–470, 2006.

[19] J. Hert, P. Willet, D. Wilton, P. Acklin, K. Azzaoui, E. Jacoby, and A. Schuffenhauer. Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Organic and Biomolecular Chemistry*, 2:3256–3266, 2004.

[20] Robert N. Jorissen and Michael K. Gibson. Virtual screening of molecular databases using support vector machines. *J. Chem. Info. Model.*, 45(3):549–561, 2005.

[21] S. K. Kearsley, S. Sallamack, E. M. Fluder, J. D. Andose, R. T. Mosley, and R. P. Sheridan. Chemical similarity using physiochemical property descriptors. *J. Chem. Inf. Comput. Sci.*, 36:118–127, 1996.

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

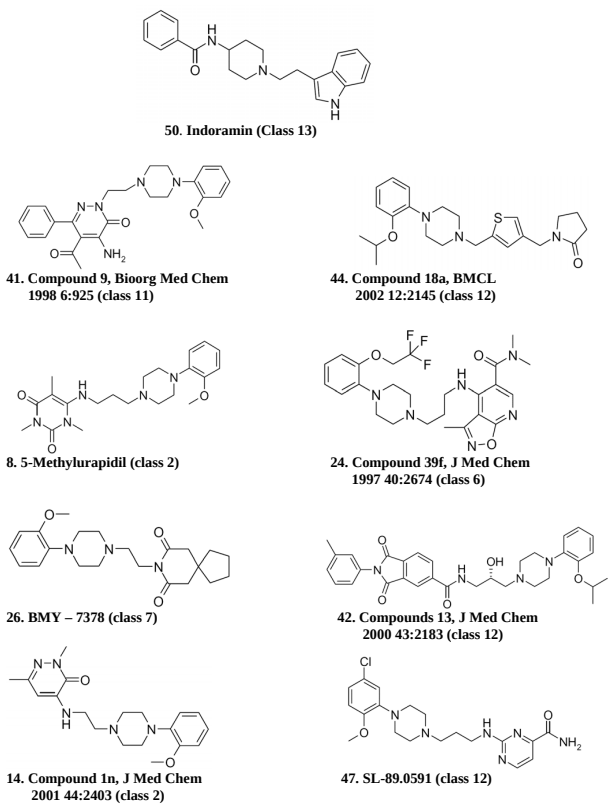

Figure 1: A subset of the α_{1A} inhibitors (A1A dataset) retrieved using BestSumMGmax and missed by StdRetTM and TurboMaxFusion5 with Indoramin used as query.

[23] D. Rogers, R. Brown, and M. Hahn. Using extended-connectivity fingerprints with laplacian-modified bayesian analysis in high-throughput screening. *J. Biomolecular Screening*, 10(7):682–686, 2005.

[24] Jamal C. Saeh, Paul D. Lyne, Bryan K. Takasaki, and David A. Cosgrove. Lead hopping using svm and 3d pharmacophore fingerprints. *J. Chem. Info. Model.*, 45:1122–113, 2005.

[25] N. Salim, J. D. Holliday, and P. Willett. Combination of fingerprint-based similarity coefficients using data fusion. *J. Chem. Inf. Comput. Sci.*, 43(2):435–442, 2003.

[26] V. Schnecke and J. Bostrom. Computational chemistry-driven decision making in lead generation. *Drug Discovery Today*, 11:43–50, 2006.

[27] Nikolaus Stiefl, Ian A. Watson, Kunt Baumann, and Andrea Zaliani. Erg: 2d pharmacophore descriptor for scaffold hopping. *J. Chem. Info. Model.*, 46:208–220, 2006.

[28] B. Teufel and S. Schmidt. Full text retrieval based on syntactic similarities. *Information Systems*, 31(1), 1988.

[29] Nikil Wale and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *International Conference in Datamining. (ICDM)*, 2006.

[30] Martin Whittle, Valerie J. Gillet, and Peter Willett. Enhancing the effectiveness of virtual screening by fusing nearest neighbor list: A comparison of similarity coefficients. *J. Chem. Info. Model.*, 44:1840–1848, 2004.

[31] Peter Willett. Chemical similarity searching. *J. Chem. Info. Model.*, 38(6):983–996, 1998.

[32] P. N. Wolohan, L. B. Akella, R. J. Dorfman, P. G. Nell, S. M. Mundt, and R. D. Clark. Structural units analysis identifies lead series and facilitates scaffold hopping in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.*, 46:1188–1193, 2005.

[33] Qiang Zhang and Ingo Muegge. Scaffold hopping through virtual screening using 2d and 3d similarity descriptors: Ranking, voting and consensus scoring. *J. Chem. Info. Model.*, 49:1536–1548, 2006.