Methods for Effective Virtual Screening and 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

UMN-CSE Technical Report 07–010

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, which 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. Two sets of techniques are presented that capture these indirect similarities using approaches based on automatic relevance feedback and on analyzing the similarity network formed by the query and the database compounds. Experimental evaluation shows that many 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 [23]. This represents a significant hurdle in the early stages of drug discovery.

A popular approach for finding these hits is to use a compound, known to possess some of the desired activity properties, as a reference and identify other compounds from a large compound database that have a similar structure. This is nothing more than a ranked-retrieval using the reference compound as a *query*. This approach relies on the well-known fact that compounds sharing key structural features will most likely have similar activity against a biomolecular target. This is referred to as the structure activity relationship (SAR) [9]. The 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].

However, 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 [17, 26]. 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 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 chemotype using the information of already known bioactive small molecules is termed as *scaffold-hopping* [17, 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 that take into account additional information beyond their structure-based similarities. These *indirect* ways 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. We present two sets of techniques to capture such indirect similarities. The first set, contains techniques that are based on automatic relevance feedback, whereas the second set, derives the indirect similarities by analyzing the similarity network formed by the query and the database compounds. Both of these sets of techniques operate on the descriptor-space representation of

the compounds and are independent of the of selected descriptor-space.

We experimentally evaluate the performance of these 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 standard ranked-retrieval. Among them, the methods based on the similarity-network perform the best and substantially outperform previously developed scaffold-hopping schemes. Moreover, even though these methods were created to improve the scaffold-hopping performance, our results show that many of them are quite effective in improving the ranked-retrieval performance as well.

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 and discusses the results obtained. 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 chemical structure of a compound relates to its activity (SAR) [9]. As such, effective solutions can be devised that rank the compounds on 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. [8,17]) 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 discovery novel leads away 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, we will use $\mathrm{sim}(c_i, c_j)$ to denote their (*direct*) similarity which is computed with respect to their descriptor-space representation by a suitable similarity measure.

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 de-

scriptors can be based on physiochemical properties as well as topological and geometric substructures (fragments) [1, 3, 12, 18, 25, 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) [18, 25], 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. ECFP is also a 2D topological descriptor-space and many flavors of these descriptors have been described by several authors [18, 25]. 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 [18, 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, 15] and binding property pairs [22] to generate pharmacophoric descriptor-space. A detailed description on the generation of these pharmacophoric descriptors can be found in [27].

5 Methods

In order to improve the scaffold-hopping performance we developed a set of techniques that measure 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.

We developed two sets of techniques to capture such indirect similarities that were inspired by research in the fields of information retrieval and social network analysis. The first set, contains techniques that use various forms of automatic relevance feedback to identify a set of compounds to be used for creating an indirect similarity measure, whereas the second set, derives the indirect similarities by analyzing the network formed by a k-nearest-neighbor graph representation of the query and the database compounds. Both of these sets of techniques operate on the descriptor-space representation of the compounds and are independent of the of selected descriptor-space.

5.1 Relevance-Feedback-based Methods

5.1.1 Top-k Weighting This approach, which is based on the Rochio [24] scheme for automatic relevance feedback, first retrieves the top-k compounds for a given query q and then uses these compounds to derive an indirect similarity between q and each of the compounds in the database. Specifically, if A is the initial set of top-k compounds, the new similarity, $\mathrm{sim}_A(q, c)$, between q and a compound c is given by

$$\mathrm{sim}_A(q, c) = \alpha\,\mathrm{sim}(q, c) + (1 - \alpha)\,\mathrm{sim}(c, A), \qquad (1)$$

where $0 \leq \alpha \leq 1$ is a user-specified parameter that controls the degree to which the new similarity is affected by the compounds in A. We will refer to this method as TOPKAVG.

The motivation behind this approach is that for reasonably small values of k, 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 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

5.1.2 Cluster Weighting This method is similar in spirit to TOPKAVG, but employs a clustering-based approach to identify the set of compounds to use for automatic relevance feedback. We will refer to this scheme as CLUSTWT and consists of the following four steps. First, it finds the top-k most similar compounds to a query q. Second, it clusters these compounds into $l = k/m$ sets $\{S_1, \ldots, S_l\}$ each of size m (assuming that k is a multiple of m). Third, it selects among these sets,

the set S^* that has the highest similarity to the query. Fourth, it uses Equation 1 to re-rank all the compounds in the database using S^* as the relevance feedback set (i.e., $A = S^*$).

The clustering is computed using a fixed-capacity heuristic min-cut partitioning algorithm on the complete weighted graph whose nodes are the k compounds and the edge-weights are the similarities between them [20, 21]. Consequently, the inter-cluster compound-to-compound similarities are explicitly minimized leading to clusters in which the intra-cluster similarities are implicitly maximized (i.e., each cluster will end-up containing similar compounds).

By using for relevance feedback the set S^*, which contains compounds that are most similar to the query, CLUSTWT selects the cluster that will most likely have a large number of active compounds. This is similar in spirit to the method that TOPKAVG uses to select its own relevance feedback set A. However, since S^* contains compounds that are also very similar to each-other, the number of active compounds that it contains will tend to be higher than that contained in A (assuming that both A and S^* have the same size). In fact, S^* has already incorporated some form of automatic relevance feedback, since all pairwise similarities between its compounds were taken into account during the clustering process. The fact that objects that are relevant to a query tend to cluster together is well-known within the document retrieval community and is usually referred to as the clustering hypothesis [16].

5.1.3 Sum-based Search The performance of TOPKAVG and CLUSTWT depends on selecting a reasonable value for the size of the set used to provide automatic relevance feedback. If that set is too small, it may not incorporate a sufficiently large number of active compounds and thus lead to limited (if any) performance improvements, whereas if the set is too large, it may degrade the performance by incorporating a relatively large number of inactive compounds. Unfortunately, our initial experiments showed that the right size of the relevance feedback set is dataset dependent.

Motivated by this observation we developed a scheme for automatic relevance feedback, which instead of using a fixed number of compounds, it does so in a progressive fashion. Specifically, if A is the set of compounds that have been retrieved thus far, then the compound selected next, c_{next}, is the one that has the highest average similarity to the set $A \cup \{q\}$. That is,

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

This compound is added in A and the overall process is repeated until the desired number of compounds is retrieved or all the compounds in D have been ranked. Thus, in this scheme, as soon as a compound is retrieved it is used to expand the set of compounds used to provide relevance feedback. We will refer to this method as BESTSUMDESCSIM.

5.1.4 Max-based Search A common characteristic to the three schemes described so far 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 relevance feedback set. Since the compounds in the relevance feedback set will tend to be structurally similar to the query compound (with the CLUSTWT potentially being an exception), 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 best-search scheme that is based on the BESTSUMDESCSIM approach but instead of selecting the next compound based on its average similarity to $A \cup \{q\}$, 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

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

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 BESTMAXDESCSIM to identify compounds that are structurally different from the query. We will refer to this schemes as BESTMAXDESCSIM.

5.2 Nearest-Neighbor Graph-based Methods

These methods, motivated by the field of social (relational) network analysis, determine the similarity between a pair of compounds by taking into account any other compounds that are very similar to either or both of them. Thus, the similarity depends on the structure of the network formed by all highly similar pairs of compounds.

The network linking the database compounds with each other *and* with the query is determined by us-

ing 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 correspond 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 the 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

$$\mathrm{sim}_G(c_i, c_j) = \frac{\mathrm{adj}_G(c_i) \cap \mathrm{adj}_G(c_j)}{\mathrm{adj}_G(c_i) \cup \mathrm{adj}_G(c_j)}, \tag{4}$$

where $\mathrm{adj}_G(c_i)$ and $\mathrm{adj}_G(c_j)$ 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. Since a pair of active compounds will be more similar to other active compounds than an active-inactive pair, their similarity according to Equation 4 will be high. Also, since Equation 4 can potentially assign a high similarity value to a pair of compounds even if their direct similarity is very low (as long as they have a large number of common neighbors), it facilitates scaffold-hopping.

For each of the NG and MG graphs we developed two retrieval schemes that use Equation 4 as the similarity measure in the sum- and max-based search strategies represented in Equations 2 and 3. For example, in the case of the NG graph and the sum-based search strategy, the next compound c_{next} to be retrieved is given by

$$c_{next} = \arg\max_{c_i \in D-A} \{\mathrm{sim}_{NG}(c_i, A \cup \{q\})\}, \tag{5}$$

where $\mathrm{sim}_{NG}(c_i, A \cup \{q\})$ is the average pairwise similarity between c_i and the compounds in A computed using Equation 4 for the NG graph. The equations for the other schemes are derived in a similar fashion. We will refer to these four schemes as BESTSUMNG, BESTMAXNG, BESTSUMMG, and BESTMAXMG, respectively.

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 search 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) [18], 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 [26, 33].

The second set of methods include the turbo search schemes (TURBOSUMFUSION and TURBOMAXFUSION) [17] and the structural unit analysis based techniques [32] all of which utilize 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 ranking of c in the $k+1$ lists, whereas in TURBOSUMFUSION, the similarity is equal to the sum of all the similarities in these rankings. Similar methods based on consensus scoring, rank averaging, and voting have been investigated in [33].

The TURBOSUMFUSION approach is similar to that of the TOPKAVG described in Section 5.1.1 as it utilizes relevance feedback mechanism to re-rank a database with respect to a query. However, the TURBOSUMFUSION approach treats every compound in the top-k set as equally important along with the query, whereas in TOPKAVG, each compound in A is given a weight of $(1-\alpha)(1/|A|\alpha)$ relative to q.

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, 19]. 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.

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, 17, 32, 33].

In this paper we use an objective way of defining which compounds can be considered as 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. These fingerprints are well-designed to capture structural similarity between two compounds [27].

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

$$\mathrm{sim}(c_i, c_j) = \frac{\sum_{k=1}^{n} c_{ik}c_{jk}}{\sum_{k=1}^{n}(c_{ik})^2 + \sum_{k=1}^{n}(c_{jk})^2 - \sum_{k=1}^{n} c_{ik}c_{jk}}, \quad (6)$$

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. For CLUSTWT we used hMETIS [20, 21] to perform the clustering into fixed sized clusters.

We varied the parameter values for the methods described in Section 5 and obtained results by averaging over four different sets of values. For TOPKAVG, which depends on the number of compounds k used in relevance feedback, we used $k = 5$, 10, 15, and 20. For CLUSTWT, which depends on the cluster size m and the number of compounds k on which the clustering was performed, we used $m = 25$ and 40 and $k = 200$ and 400. These parameter values were selected because

they gave the best results in our experiments. For the nearest-neighbor methods which depend on the number of neighbors, we used $k = 4$, 6, 8, and 10 for the BESTSUMNG and BESTMAXNG, and $k = 12$, 16, 20, and 24 for the BESTSUMMG and BESTMAXMG schemes. These values were chosen because they gave good results. Moreover, 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 5.2).

We also compared our schemes against TURBOMAXFUSION and TURBOSUMFUSION [17]. 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.

7.4 Standard Retrieval

For each problem, we obtain a baseline performance by ranking all the compounds with respect to each active compound using the Tanimoto similarity. We call this *Standard Retrieval* and denote it by STDRET.

7.5 Performance Assessment Measures

We measure ranked-retrieval and scaffold-hopping performance using *uninterpolated precision* [16]. 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 normalized 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 or scaffold-hopping 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 or scaffold-hopping 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 whereas if the ARP is greater than zero, r performs better than q. 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 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

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 define the ARP of the two methods to be statistically significant if $p \leq 0.01$.

The rest of this section highlights some of the key observations that can be made by analyzing the results in these tables.

8.1.1 Performance of Relevance Feedback Methods Comparing the performance of the four relevance-feedback-based methods described in Section 5.1 against STDRET, we see that all of them lead to better scaffold-hopping results. Among them, the results achieved by CLUSTWT and BESTSUMDESCSIM are 63% and 94% better than STDRET, respectively and also these improvements are statistically significant. However, all four of these methods achieve somewhat worse ranked-retrieval performance (3% to 15%). Moreover, these differences are statistically significant for BESTSUMDESCSIM and BESTMAXDESCSIM.

Comparing the four methods against TURBOSUMFUSION and TURBOMAXFUSION, we observe that the relative performance of most of these methods varies, with some methods doing better for scaffold-hopping and others doing better for ranked-retrieval. However, with the exception of TOPKAVG, which is statistically better than the two fusion-based scheme for ranked-

Table 1: Performance for Scaffold-Hopping.

	StdRet	TurboSumFusion	TurboMaxFusion	TopKAvg	ClustWt	BestSumDescSim	BestMaxDescSim	BestSumNG	BestMaxNG	BestSumMG	BestMaxMG
StdRet		-0.44 (0.031)	-0.82 (0.006)	-0.31 (0.127)	-0.71 (0.007)	-0.96 (0.002)	-0.89 (0.024)	-1.51 (0.000)	-1.52 (0.000)	-1.6 (0.000)	-1.59 (0.000)
TurboSumFusion	0.44 (0.031)		-0.38 (0.073)	0.13 (0.024)	-0.26 (0.029)	-0.52 (0.068)	-0.44 (0.298)	-1.07 (0.000)	-1.07 (0.000)	-1.16 (0.000)	-1.15 (0.000)
TurboMaxFusion	0.82 (0.006)	0.38 (0.073)		0.51 (0.013)	0.11 (0.467)	-0.14 (0.547)	-0.07 (0.835)	-0.69 (0.002)	-0.7 (0.005)	-0.78 (0.001)	-0.77 (0.000)
TopKAvg	0.31 (0.127)	-0.13 (0.024)	-0.51 (0.013)		-0.4 (0.001)	-0.65 (0.032)	-0.57 (0.177)	-1.2 (0.000)	-1.2 (0.000)	-1.29 (0.000)	-1.28 (0.000)
ClustWt	0.71 (0.007)	0.26 (0.029)	-0.11 (0.467)	0.4 (0.001)		-0.25 (0.316)	-0.18 (0.645)	-0.8 (0.000)	-0.81 (0.000)	-0.9 (0.000)	-0.88 (0.000)
BestSumDescSim	0.96 (0.002)	0.52 (0.068)	0.14 (0.547)	0.65 (0.032)	0.25 (0.316)		0.07 (0.754)	-0.55 (0.038)	-0.56 (0.064)	-0.65 (0.039)	-0.63 (0.038)
BestMaxDescSim	0.89 (0.024)	0.44 (0.298)	0.07 (0.835)	0.57 (0.177)	0.18 (0.645)	-0.07 (0.754)		-0.62 (0.109)	-0.63 (0.140)	-0.72 (0.053)	-0.7 (0.071)
BestSumNG	1.51 (0.000)	1.07 (0.000)	0.69 (0.002)	1.2 (0.000)	0.8 (0.000)	0.55 (0.038)	0.62 (0.109)		-0.01 (0.947)	-0.1 (0.577)	-0.08 (0.579)
BestMaxNG	1.52 (0.000)	1.07 (0.000)	0.7 (0.005)	1.2 (0.000)	0.81 (0.000)	0.56 (0.064)	0.63 (0.140)	0.01 (0.947)		-0.09 (0.620)	-0.08 (0.614)
BestSumMG	1.6 (0.000)	1.16 (0.000)	0.78 (0.001)	1.29 (0.000)	0.9 (0.000)	0.65 (0.039)	0.72 (0.053)	0.1 (0.577)	0.09 (0.620)		0.01 (0.886)
BestMaxMG	1.59 (0.000)	1.15 (0.000)	0.77 (0.000)	1.28 (0.000)	0.88 (0.000)	0.63 (0.038)	0.7 (0.071)	0.08 (0.579)	0.08 (0.614)	-0.01 (0.886)	

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.

retrieval, all other differences are not statistically significant.

Comparing the four relevance-feedback-based methods against each other we see that most of them perform the same for both scaffold-hopping and ranked-retrieval and whatever differences that exist are not statistically significant. Despite of this, the average performance of BestSumDescSim is better than BestMaxDescSim, indicating that the sum-based search strategy leads to better results. The results also show that the ClustWt is better than TopKAvg for scaffold-hopping and that this difference is statistically significant.

8.1.2 Performance of Nearest-Neighbor Graph-Based Methods Comparing the performance of the nearest-neighbor methods, we observe that all of these schemes show good performance for scaffold-hopping as well as ranked-retrieval. Among them, the best performing method is BestSumNG. It achieves the best balance between the ranked-retrieval and scaffold-hopping performance. Furthermore, similar to the relevance feedback-based methods, the sum-based search methods outperform the corresponding max-based methods. However, these differences are not statistically significant.

The results also show that the nearest-neighbor methods performs significantly better than all the other methods for scaffold-hopping and most of these differences are statistically significant (BestSumDescSim and BestMaxDescSim are the two exceptions). In particular, the performance of the nearest-neighbor methods are 62% to 300% better than the StdRet and the fusion-based methods and 46% to 244% better than the relevance-feedback-based methods.

The nearest-neighbor methods also achieve better performance than all of the methods for ranked-retrieval, although most of these differences are not statistically significant. BestSumNG is a clear exception as its ranked-retrieval performance is also significantly and statistically better than all the other non graph-based techniques. For example, compared to the fusion-based techniques its ranked-retrieval performance is 62% to 209% better.

Table 2: Performance for Ranked-Retrieval.

	StdRet	TurboSumFusion	TurboMaxFusion	TopKAvg	ClustWt	BestSumDescSim	BestMaxDescSim	BestSumNG	BestMaxNG	BestSumMG	BestMaxMG
StdRet		0.14 (0.019)	0.21 (0.001)	0.04 (0.332)	0.06 (0.415)	0.17 (0.009)	0.27 (0.002)	-0.25 (0.015)	-0.12 (0.179)	-0.18 (0.151)	-0.08 (0.434)
TurboSumFusion	-0.14 (0.019)		0.07 (0.156)	-0.1 (0.001)	-0.08 (0.113)	0.03 (0.502)	0.13 (0.137)	-0.39 (0.001)	-0.26 (0.003)	-0.32 (0.016)	-0.22 (0.037)
TurboMaxFusion	-0.21 (0.002)	-0.07 (0.156)		-0.17 (0.001)	-0.15 (0.101)	-0.04 (0.426)	0.06 (0.419)	-0.46 (0.001)	-0.33 (0.002)	-0.39 (0.013)	-0.29 (0.028)
TopKAvg	-0.04 (0.332)	0.1 (0.001)	0.17 (0.001)		0.02 (0.725)	0.13 (0.017)	0.23 (0.016)	-0.29 (0.004)	-0.16 (0.054)	-0.22 (0.080)	-0.12 (0.226)
ClustWt	-0.06 (0.415)	0.08 (0.113)	0.15 (0.101)	-0.02 (0.725)		0.11 (0.168)	0.21 (0.071)	-0.31 (0.009)	-0.18 (0.027)	-0.24 (0.047)	-0.14 (0.158)
BestSumDescSim	-0.17 (0.009)	-0.03 (0.502)	0.04 (0.426)	-0.13 (0.017)	-0.11 (0.168)		0.1 (0.121)	-0.42 (0.001)	-0.29 (0.004)	-0.35 (0.021)	-0.25 (0.051)
BestMaxDescSim	-0.27 (0.002)	-0.13 (0.137)	-0.06 (0.419)	-0.23 (0.016)	-0.21 (0.071)	-0.1 (0.121)		-0.52 (0.001)	-0.39 (0.002)	-0.45 (0.008)	-0.35 (0.019)
BestSumNG	0.25 (0.015)	0.39 (0.001)	0.46 (0.001)	0.29 (0.004)	0.31 (0.009)	0.42 (0.001)	0.52 (0.001)		0.13 (0.148)	0.07 (0.519)	0.17 (0.079)
BestMaxNG	0.12 (0.179)	0.26 (0.003)	0.33 (0.002)	0.16 (0.054)	0.18 (0.027)	0.29 (0.004)	0.39 (0.002)	-0.13 (0.148)		-0.06 (0.484)	0.04 (0.591)
BestSumMG	0.18 (0.151)	0.32 (0.016)	0.39 (0.013)	0.22 (0.080)	0.24 (0.047)	0.35 (0.021)	0.45 (0.008)	-0.07 (0.517)	0.06 (0.484)		0.1 (0.036)
BestMaxMG	0.08 (0.434)	0.22 (0.037)	0.29 (0.028)	0.12 (0.226)	0.14 (0.158)	0.25 (0.051)	0.35 (0.019)	-0.17 (0.079)	-0.04 (0.591)	-0.1 (0.036)	

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.

8.2 Performance of Descriptor-Spaces and Datasets

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 ClustWT and the BestSumNG methods as these methods achieve the best scaffold-hopping and ranked-retrieval performance among the relevance-feedback- and graph-based methods, respectively.

The results of these evaluations are shown in Figures 1 and 2, which compare the performance of StdRet against ClustWT and BestSumNG, respectively. 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. For ClustWT and BestSumNG we also show error bars that correspond to the standard deviation of the results obtained for the four sets of parameter values used for these schemes.

These results show that for scaffold-hopping, ClustWT outperforms StdRet in most dataset and descriptor-space combinations. However, the actual performance gains are dataset and descriptor-space dependent. For example, ClustWT achieves significant gains on the A1A and FXa datasets for the ErG and ECZ3 descriptor-spaces, whereas the gains for the other datasets and/or descriptor-spaces are not as dramatic. In terms of ranked-retrieval performance, these results show that in the case of the GF descriptor-space, ClustWT performs consistently better than StdRet across all datasets. However, ClustWT's ranked-retrieval performance for the other two descriptor-spaces is somewhat mixed.

Finally, the results in Figure 2 show that for scaffold-hopping, BestSumNG performs consistently better than StdRet for all the descriptor-space and dataset combinations. However, similarly to ClustWT, the actual gains are dataset and descriptor-space dependent. For example, the gains are particularly high for the FXa, A1A, and COX2 datasets and for the ErG descriptor space. Similar trends can be observed with

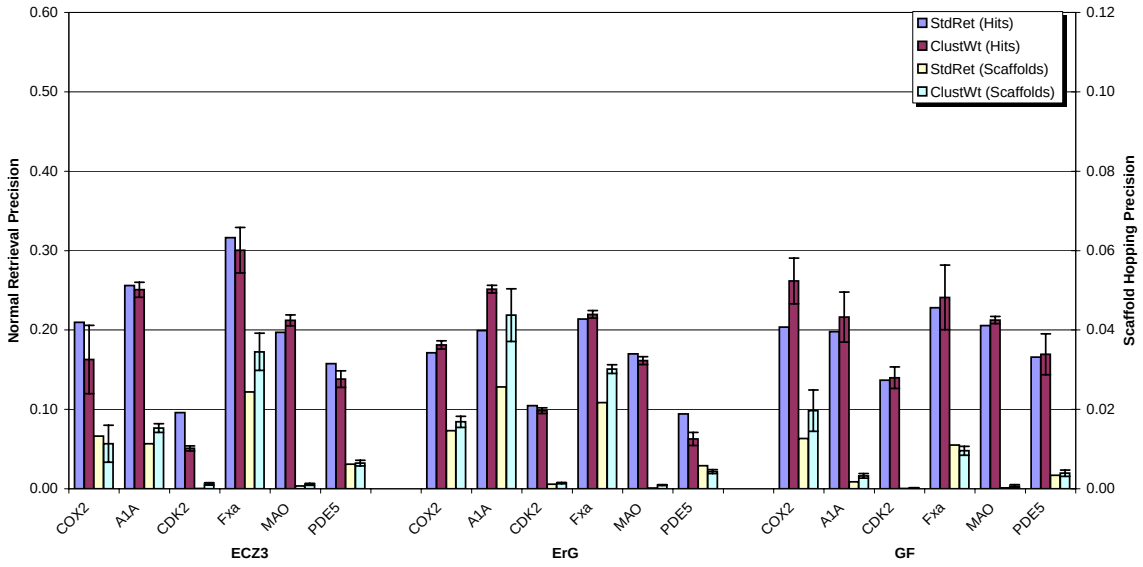

Figure 1: StdRet versus ClustWt.

the ranked-retrieval results, with BESTSUMNG outperforming STDRET. Moreover, the performance gains achieved on some problems by BESTSUMNG are usually much higher than the performance degradations in others.

9 Conclusion

In this paper we introduced a number of methods based on relevance feedback and social (relational) network analysis to improve scaffold-hopping and ranked-retrieval. Our results showed that among these methods, the ones based on social network analysis consistently and substantially outperform the standard retrieval as well as previously introduced methods for these problems.

10 Acknowledgement

This work was supported by NSF EIA-9986042, ACI-0133464, IIS-0431135, NIH RLM008713A, the Army High Performance Computing Research Center contract number DAAD19-01-2-0014, 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.*

[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] 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.

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

[17] 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.

[18] J. Hert, P. Willet, D. Wilton, P. Acklin, K. Azzaoui, E. Jacoby, and A. Schuffenhauer. Comparison of topo-

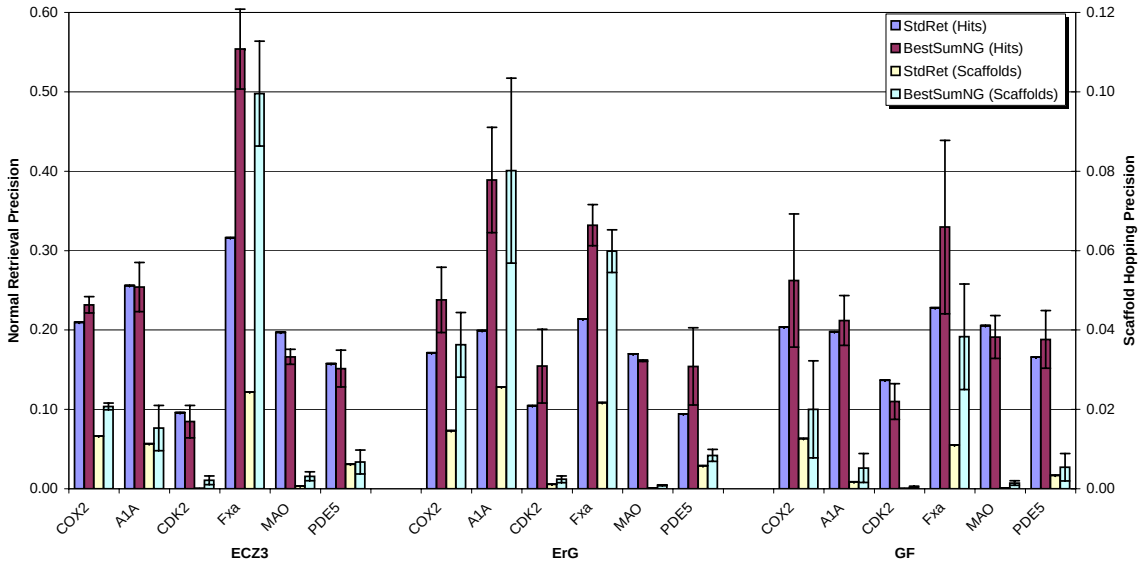

Figure 2: StdRet versus BestSumNG.

logical descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Organic and Biomolecular Chemistry*, 2:3256–3266, 2004.

[19] 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.

[20] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: Applications in vlsi domain. *Design and Automation Conference*, pages 526–529, 1997.

[21] George Karypis and Vipin Kumar. Multilevel k-way hypergraph partitioning. *Design and Automation Conference*, pages 343–348, 1999.

[22] 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.

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

[24] J. J. Rochio. Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall*, Chapter 14, 1971.

[25] 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.

[26] 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.

[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.