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 08-035

Target Identification for Chemical Compounds using Target-Ligand Activity data and Ranking based Methods

Nikil Wale and George Karypis

October 29, 2008

Target Identification for Chemical Compounds using Target-Ligand Activity data and Ranking based Methods

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

Abstract

Drug discovery is an expensive process. It has been estimated that a new drug compound that is introduced in the market after FDA approval carries a cost of approximately $800 million from the conception of target implicated for a disease to successful identification of chemical entity or drug that is successful in human trials. There is a need to cut the cost of developing new drugs (to bring overall cost lower for the producers and consumers alike) by identifying promising candidate targets as well as compounds and to tackle problems such an toxicity, lack of efficacy in humans, and poor physical properties in the early stages of drug discovery.

In order to achieve this objective, in recent years the development of computational techniques that identify all the likely targets for a given chemical compound has been an active area of research. Identification of all the potential targets for a chemical compound provides insights into its potential toxicity, helps in repositioning it, and also provides insights into the behavior and relation among targets themselves from the perspective of small molecules.

In this paper we address this problem of target identification in the context of small molecule. We present a set of techniques whose goal is to rank or prioritize the targets in the context of a given chemical compound such that most targets that have a potential to show activity to this compound appear higher in the ranked list. These methods are motivated by recent advances in category ranking and protein secondary structure prediction approaches and utilize target-ligand activity data to prioritize targets. Our extensive experimental evaluation shows that most of the methods developed in this work are either competitive or substantially outperform previously developed approaches to solve the above problem in drug discovery.

1 Introduction

Target-based drug discovery, which involves selection of an appropriate target (generally a single protein) implicated in a disease state as the first step, has become the primary approach of drug discovery in pharmaceutical industry [43]. This was made possible through the advent of High Throughput Screening (HTS) technology in the late 1980s that enabled rapid experimental testing of a large number of chemical compounds against the target of interest (using target-based assays). HTS is now routinely utilized to identify the most promising compounds (called *hits*) that show desired binding/activity against this target. Some of these compounds then go through the long and expensive process of optimization and eventually one of them goes to clinical trials. If clinical trails are successful then it becomes a drug. HTS technology was considered to usher a new era in drug discovery by reducing time and money taken to find hits that will have a high chance of eventually becoming a drug.

However, the expansion of candidate list of hits via HTS did not result in productivity gains in terms of actual drugs coming out of the drug discovery pipeline. One of the principle reasons ascribed for this failure is that the above approach suffers from a serious drawback of only focusing on the target of interest and therefore taking a very narrow view of the disease. As such, it may lead to unsatisfactory phenotypic effects such as toxicity, promiscuity, and low efficacy in the later stages of drug discovery [41, 43]. More recently, focus is shifting to directly screen molecules to identify desirable phenotypic effects using cell-based assays. This screening evaluates properties such as toxicity, promiscuity and efficacy from the onset rather than in later stages of drug discovery [25, 41, 43]. Moreover, toxicity and off-target effects are also a focus of early stages of conventional target-based drug discovery [7, 13]. But from the drug discovery perspective, target identification and subsequent validation has become the rate limiting step in order to tackle the above issues [18]. Targets must be identified for the hits in phenotypic assay experiments as the activity of hits against targets in the assay sheds light on the toxicity and promiscuity of the compounds [13, 21]. Therefore, the identification of all likely targets for a given chemical compound, also called *Target Fishing* [25], has become an important problem in drug discovery.

In this work we focus on the target identification (fishing) problem by utilizing the available target-ligand activity data matrix. In this approach, we are given a set of targets and a set of ligands (chemical compounds) and a bipartite activity relation between the targets and the ligands in the two sets.

[*] Currently at Pfizer Global Research and Development, Pfizer Inc., Groton, CT. Email: nikil.wale@pfizer.com

Given a new test chemical compound not in the set, the goal is to predict all the activity relations between the test compound and the known targets. We address this problem by formulating it as a category ranking problem. The goal is to learn a model that correctly predicts the ranking of targets for a given test compound such that all the targets that this compound shows activity against (relevant targets) are ranked higher than the targets that the compound does not show activity against (non-relevant targets). In this work, we propose a number of methods that are inspired by research in the area of multiclass/multilabel classification and protein secondary structure prediction. Specifically, we develop four methods based on SVM [45] and ranking perceptrons [15] to solve the above ranking problem. Three of these methods try to explicitly capture dependencies between different categories to build models. Our results show that the methods proposed in this work are either competitive or substantially outperform other methods currently employed to solve this problem.

The rest of this paper is organized as follows. Section 2 describes related research in the area of target identification. Section 3 introduces definition and notations used in this paper. Section 4 describes our methods for target identification. Section 5 discusses the datasets and experimental methodology used in this work. Section 6 describes our results and finally section 7 has concluding remarks.

2 Related Methods

Computational techniques are becoming increasingly popular for target identification due to plethora of data from high-throughput screening (HTS), microarrays, and other experiments [25]. Given a compound whose targets need to be identified, these techniques initially assign a score to each target based on some measure of likelihood that the compound binds to each target. These techniques then select as the potential compound's targets either those targets whose score is above a certain cut-off or a small number of the highest scoring targets. Three general classes of methods have been developed for determining the required compound-target scores. The first, referred to as *inverse docking*, contains methods that score each compound-target pair by using ligand docking approaches [13, 14]. The second, referred to as nearest-neighbor, contains methods that determine the compound-target scores by exploiting structural similarities between the compound and the target's known ligands [31]. Finally the third, referred to as model-based, contains methods that determine these scores using various machine-learning approaches to learn models for each one of the potential targets based on their known ligands [29,32,33].

The first class of methods to derive a score for each compound-target pair comes from the computational chemistry domain. Specifically, inverse docking process docks a single compound of interest to a set of targets and obtains a score for each target against this compound. The highest scoring targets are then considered as most likely targets that this compound will bind to [13]. This approach suffers from a serious drawback that all the targets used in this process must have their three dimensional structure available in order to employ a docking procedure. However, the majority of target proteins do not have such information available [4].

The second class of methods, nearest-neighbor based techniques, rely on the structure-activity relationship principle (SAR) [12, 20] which suggests that very similar compounds will have a higher chance of overlap between the sets of targets that they show activity against [31]. Therefore, identifying targets for a given chemical compound can be solved by utilizing its structural similarity with other chemical compounds that are known to be active or inactive against certain targets. In these approaches, for a given test compound, its nearest-neighbor(s) are identified from a database of compounds with know targets using some notion of structural similarity. The most likely targets for the test compound are then identified as those targets that its nearest neighbors show activity against. In these approaches the solution to target identification problem only depends on the underlying descriptor-space representation, the similarity function employed, and the definition of nearest neighbors. Note that this technique may return more than one likely target but does not provide any preference or ranking within these targets.

Lastly, a number of methods have been proposed that explicitly build models on the given set of compounds with known targets. These techniques treat the target identification problem as an instance of multilabel (or multiclass) prediction [16, 19, 24, 29] in which each of the different targets is treated as one of the potential labels to be predicted and for each given chemical compound all the targets (labels) that it binds to (belongs to) are being predicted. One of these approaches utilize multi-category bayesian models [32] wherein a model is built for every target in the database using SAR data available for each target. Compounds that show activity against a target are used as positives for that target and the rest of the compounds are treated as negatives. The input to the algorithm is a training set consisting of a set of chemical compounds and a set of targets. A model is learned for every target given a descriptor-space representation of training chemical compounds (see [32] for details). For a new chemical compound whose targets have to be predicted, an estimator score is computed for each target that reflects the likelihood of activity against this target using the learned models. The target can be ranked according to their estimator scores and the targets that get high scores can be considered as the most likely targets for this compound. Similar modeling approaches for target identification build one-versus-rest binary SVM models [29] or neural network models [33].

Note that even though the underlying machine learning problem is that of multiclass or multilabel prediction, all of the above model-based methods essentially build one-vs-rest models and then produce a ranking of the possible targets using the outputs of these models.

3 Definitions and Notations

The target identification problem that we consider in this paper is defined as follows:

Definition 1 (Target Identification Problem). *Given a set of compounds (more than one) whose bioactivity is known to be either active or inactive against each of the targets in a given set, learn a model such that it correctly predicts for a test compound a ranking of all the targets according to how likely they are to show activity against the test compound.*

Throughout this paper we will use $\mathcal{C} = \{c_1, \ldots, c_M\}$ to denote a library of chemical compounds, $\mathcal{T} = \{\tau_1, \ldots, \tau_N\}$ to denote a set of protein targets, and will assume that they contain M and N elements, respectively. For each compound c_i, we will use $\mathcal{T}_i$ to denote the set of all targets that c_i shows activity against. Note that $\mathcal{T}_i \subseteq \mathcal{T}$. We will use $\mathcal{T}^*$ to denote a total ordering of the elements of $\mathcal{T}$. Given two sets A and B such that $A \subset \mathcal{T}$ and $B \subset \mathcal{T}$, we will use $A <_{\mathcal{T}^*} B$ to denote that every element of A precedes every element of B in $\mathcal{T}^*$, and $A \not<_{\mathcal{T}^*} B$ otherwise.

Each compound will be represented by a topological descriptor-based representation [11, 12]. In this representation, each compound is modeled as a frequency vector of certain topological descriptors (e.g., subgraphs) present in its molecular graph. Each dimension's frequency counts the number of times (i.e., embeddings) the corresponding topological descriptor is present in the compound's molecular graph. We will use n to represent the dimensionality of descriptor-space representation of the chemical compounds in $\mathcal{C}$. Given a compound c, and a parameter k, we define top-k to be the k targets that are most likely to show activity for c. Lastly, throughout this paper we will use the terms target, category, and labels interchangeably.

4 Methods

Our solution to Target Identification problem relies on the principle of Structure Activity Relationship (SAR). Specifically, we develop solutions for the target identification problem using SAR data by formulating it as a ranking problem. We pursue this approach because in real world situations a practitioner might want to know the top-k most likely targets for a given compound so that they can be tested experimentally or further investigated. Therefore, if the relevant categories fall in one of these top-k predicted categories, they will have a higher chance to be recognized.

In the rest of this section will describe the four approaches we have proposed to solve the category ranking problem in the context of target identification.

4.1 SVM-based Method

One approach for solving the ranking problem is to build for each target $\tau_i \in \mathcal{T}$ a one-versus-rest binary SVM classifier. Given a test chemical compound c, the classifier for each target τ_i will then be applied to obtain a prediction score $f_i(c)$. The ranking of the N targets $\mathcal{T}^*$ will be obtain by simply sorting the targets based on their prediction scores. That is,

$$\mathcal{T}^* = \operatorname*{argsort}_{\tau_i \in \mathcal{T}} \{f_i(c)\}, \tag{4.1}$$

where argsort returns an ordering of the targets in decreasing order of their prediction scores $f_i(c)$. Note that this approach assumes that the prediction scores obtained from the N binary classifiers are directly comparable, which may not necessarily be valid. This is because different classes may be of different sizes and/or less separable from the rest of the dataset indirectly affecting the nature of the binary model that was learned and consequently their prediction scores [36]. This SVM-based sorting method is similar to the approach described previously [29] in Section 2. We refer to this method as SVMR.

4.2 Cascaded SVM-based Method

A limitation of the previous approach is that by building a series of one-vs-rest binary classifiers it does not explicitly couple the information on the multiple categories that each compound belongs to during model training and as such it cannot capture dependencies that might exist between the different categories. A promising approach that has been explored to capture such dependencies is to formulate it as a cascaded learning problem [19, 27, 28]. In these approaches, two sets of binary one-vs-rest classification models for each category, referred to as L_1 and L_2, are connected together in a cascaded fashion. The L_1 models are trained on the initial inputs and their outputs are used as input, either by themselves or in conjunction with the initial inputs, to train the L_2 models. This cascaded process is illustrated in Figure 1. During prediction time, the L_1 models are first used to obtain the required predictions which are used as input to the L_2 models in order to obtain the final predictions. Since the L_2 models incorporate information about the predictions produced by the L_1 models, they can potentially capture inter-category dependencies.

Motivated by the above, we developed a ranking method for the target identification problem in which both the L_1 and L_2 models consist of N binary one-vs-rest SVM classifiers, one for each target in $\mathcal{T}$. The L_1 models correspond exactly to the set of models built by the SVMR method discussed in the previous section. The representation of each compound in the training set for the L_2 models consists of its descriptor-

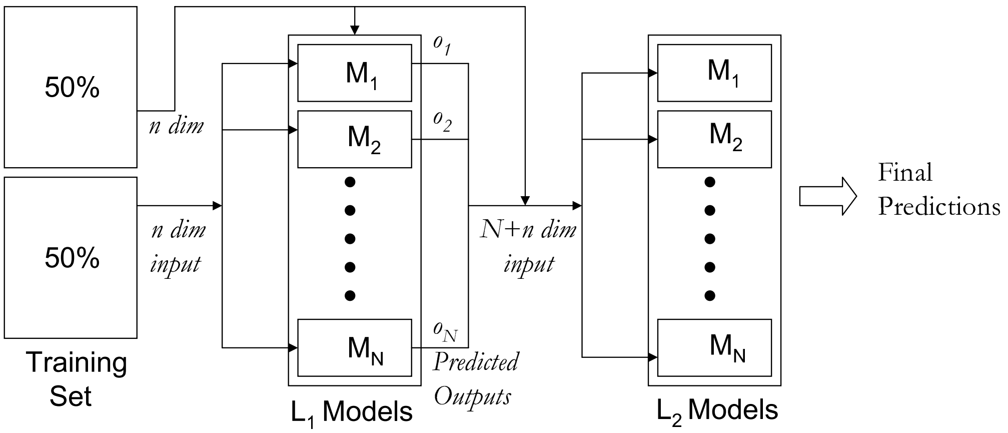

Figure 1: Cascaded SVM Classifiers.

space based representation and its output from each of the N L_1 models. Thus, each compound corresponds to an $n+N$ dimensional vector, where n is the dimensionality of the descriptor space. The final ranking $\mathcal{T}^*$ of the targets for a compound c will be obtained by sorting the targets based on their prediction scores from the L_2 models ($f_i^{L_2}()$). That is,

$$\mathcal{T}^* = \operatorname*{argsort}_{\tau_i \in \mathcal{T}} \left\{ f_i^{L_2}(c) \right\}, \qquad (4.2)$$

We will refer to this approach as SVM2R.

A potential problem with such an approach is that the descriptor-space based representation of a chemical compound and the set of its outputs from L_1 models are not directly comparable. Therefore, in this work, we also experiment with various kernel functions that combine n dimensional descriptor-space representation and the N dimensional L_1 model outputs of a chemical compound. Specifically, we experiment with two forms of the kernel function. The first function is given by

$$\mathcal{K}(c_i, c_j) = \alpha\, \mathcal{K}_A(c_i, c_j) + (1-\alpha)\, \mathcal{K}_B(c_i, c_j), \qquad (4.3)$$

where α is a user defined parameter and $\mathcal{K}_A$ and $\mathcal{K}_B$ are the kernel functions measuring the similarity between compound vector formed using descriptor-space based representation and L_1 SVM classifier outputs respectively. The second kernel function is given by

$$\mathcal{K}(c_i, c_j) = \mathcal{K}_A(c_i, c_j)\, \mathcal{K}_B(c_i, c_j). \qquad (4.4)$$

These approaches are motivated by the work on kernel fusion [30] and tensor product kernels [9].

4.3 Ranking Perceptron Method

We also developed a ranking method that exploits the potential dependencies between the categories based on the ranking perceptron [15]. The ranking perceptron extends Rosenblatt's linear perceptron classifier [39] for the task of learning a ranking function. The perceptron algorithm and its variants have proven to be effective in a broad range of applications in machine learning, information retrieval and bioinformatics [17, 24, 37].

Our approach is based on the online version of the ranking perceptron algorithm proposed to learn a ranking function on a set of categories developed by Crammer and Singer [17]. This algorithm takes as input a set of objects and the categories that they belong to and learns a function that for a given object c it ranks the different categories based on the likelihood that c binds to the corresponding targets. During the learning phase, the distinction between categories is made only via a binary decision function that takes into account whether a category is part of the object's categories (relevant set) or not (non-relevant set). As a result, even though the output of this algorithm is a total ordering of the categories, the learning is only dependent on the partial orderings induced by the set of relevant and non-relevant categories.

The pseudocode of our ranking perceptron algorithm is shown in Algorithm 1. This algorithm learns a linear model W that corresponds to a $N \times n$ matrix, where N is the number of targets and n is the dimensionality of the descriptor space. Using this model, the prediction score for compound c_i and target τ_j is given by $\langle W_j, c_i \rangle$, where W_j is the jth row of W, c_i is the descriptor-space representation of the compound, and $\langle \cdot, \cdot \rangle$ denotes a dot-product operation. Our algorithm extends the work of Crammer and Singer by introducing margin based updates and extending the online version to a batch setting. Specifically, for each training compound c_i that is active for targets belonging to categories $\mathcal{T}_i \subseteq \mathcal{T}$, our algorithm learns W such that the following constraints as satisfied:

$$\forall \tau_j \in \mathcal{T}_i,\ \forall \tau_k \in \mathcal{T} \setminus \mathcal{T}_i;\ \ \langle W_j, c_i \rangle - \langle W_k, c_i \rangle \geq \beta, \qquad (4.5)$$

where β is a user-specified non-negative constant that corresponds to the separation margin. The idea behind these con-

Algorithm 1 Learning Category Weight Vectors with the ranking perceptron algorithm

```
input:
    $\mathcal{C}$: Set of $M$ training compounds.
    $\mathcal{T}$: Set of $N$ targets (categories).
    $(c_i, \mathcal{T}_i)$: Compound $c_i$ and its categories $\mathcal{T}_i$.
    $\beta$: User defined margin constraint.
    $n$: Dimensionality of the compound's descriptor-space representation.

output:
    $W$: $N \times n$ model matrix.

 1: $W$ = 0 {Initial model}
 2: $\eta$ = $1/M$ {Update weight}
 3: while (STOPPING CRITERION == FALSE) do
 4:   for $i$=1 to $M$ do
 5:     $\mathcal{T}^*$ = $\text{argsort}_{\tau_j \in \mathcal{T}} \{\langle W_j, c_i \rangle\}$
 6:     $\tau_j$ = lowest ranked target of $\mathcal{T}_i$ in $\mathcal{T}^*$
 7:     $\tau_k$ = highest ranked target of $\mathcal{T} \setminus \mathcal{T}_i$ in $\mathcal{T}^*$
 8:     if $\langle W_j, c_i \rangle - \langle W_k, c_i \rangle < \beta$ then
 9:       for $\forall \tau_q \in \mathcal{T}_i : \langle W_q, c_i \rangle - \langle W_k, c_i \rangle < \beta$ do
10:         $\lambda = |\{\tau_r \in \mathcal{T} \setminus \mathcal{T}_i : \langle W_q, c_i \rangle - \langle W_r, c_i \rangle < \beta\}|$
11:         $W_q = W_q + \lambda \eta c_i$
12:       end for
13:       for $\forall \tau_r \in \mathcal{T} \setminus \mathcal{T}_i : \langle W_j, c_i \rangle - \langle W_r, c_i \rangle < \beta$ do
14:         $\lambda = |\{\tau_q \in \mathcal{T}_i : \langle W_q, c_i \rangle - \langle W_r, c_i \rangle < \beta\}|$
15:         $W_r = W_r - \lambda \eta c_i$
16:       end for
17:     end if
18:   end for
19:   $\forall \tau_i \in \mathcal{T},\ W_i = W_i/||W_i||$
20: end while
21: return $W$
```

straints is to force the algorithm to learn a model in which the set of relevant categories ($\mathcal{T}_i$) for a given chemical compound c_i are well-separated and ranked higher from all the non-relevant categories ($T \setminus \mathcal{T}_i$). Therefore, our algorithm tries to satisfy $\sum_{c_i \in \mathcal{C}} |\mathcal{T}_i| \times |\mathcal{T} \setminus \mathcal{T}_i|$ constraints for each of the training set compound c_i to enforce a degree of acceptable separation between relevant and non-relevant categories that is controlled by β.

During each outer iteration (lines 3–20) the algorithm iterates over all the training compounds (lines 4–18) and for each compound c_i it obtains a ranking $\mathcal{T}^*$ of all the categories (line 5) based on the current model W, and updates the model if any of the constraints in Equation 4.5 are violated. The check for the constraint violation is done in line 8 by comparing the lowest ranked target $\tau_j \in \mathcal{T}_i$ with the highest ranked target $\tau_k \in \mathcal{T} \setminus \mathcal{T}_i$. If there are any constraint violations, the condition on line 8 will be true and lines 9–16 of the algorithm will be executed. The model W is updated by adding/subtracting a multiple of c_i from the rows of W involved in the pair of targets of the violated constraints. Instead of updating the model's vectors by using a constant multiple, which is usually done in perceptron training, our algorithm uses a multiple that is proportional to the number of constraints that each target violates in $\mathcal{T}^*$. Specifically, for each target $\tau_q \in \mathcal{T}_i$, our algorithm (line 10) finds the number λ of targets $\tau_r \in \mathcal{T} \setminus \mathcal{T}_i$ that violate the margin constraint with τ_q and adds in the qth row of W (which is the portion of the model corresponding to target τ_q) a λ multiple of ηc_i, where η is a small constant set to $1/M$ in our experiments. The motivation behind this proportional update is that if a relevant target τ_q follows a large number of non-relevant targets in the ordering $\mathcal{T}^*$, τ_q's model (W_q) needs to move towards the direction of c_i more than the model for another relevant target $\tau_{q'}$ which is followed only by a small number of non-relevant targets in $\mathcal{T}^*$. Note that the term "follows" in the above discussion needs to be considered within the context of the margin β. A similar approach is used to determine the multiple of c_i to be subtracted from the rows of W corresponding to the non-relevant targets that are involved in violated constraints (lines 13–16). Our experiments (not reported here) showed that this proportional update achieved consistently better results than those achieved by constant update rules.

Since the ranking perceptron algorithm is not guaranteed to converge when the training instances are not β-linearly separable, Algorithm 1 incorporates an explicit *stopping criterion*. After every pass over the entire training set, it computes the average uninterpolated precision (Section 5.2.4) over all the compounds using the weights W, and terminates when this precision has not improved in N consecutive iterations. The algorithm returns the W that achieved the highest training precision over all iterations. We directly apply the above method on the descriptor-space representation of the training set of chemical compounds.

The predicted ranking for a test chemical compound c is given by

$$\mathcal{T}^* = \underset{\tau_j \in \mathcal{T}}{\text{argsort}} \{\langle W_j, c \rangle\}. \tag{4.6}$$

We will refer to this approach as RP.

4.4 SVM+Ranking Perceptron-based Method

A limitation of the above ranking perceptron method over the SVM-based methods is that it is a weaker learner as (i) it learns a linear model, and (ii) it does not provide any guarantees that it will converge to a good solution when the dataset is not linearly separable. In order to partially overcome these limitations we developed a scheme that is similar in nature to the cascaded SVM-based approach, but the L_2 models are replaced by a ranking perceptron. Specifically, N binary one-vs-rest SVM models are trained, which form the set of L_1 models, and their outputs are used to obtain a ranking model W learned using the ranking perceptron if Algorithm 1. Since the L_2 model is based only on the inputs of the L_1 models, the size of W is $N \times N$. A recent study within the context of remote homology prediction and fold recognition has shown

Table 1: Multi-target activity of the compounds.

# compounds	19,154	5,363	1,697	648	129	139	29	14	7	25
# targets	1	2	3	4	5	6	7	8	9	10–41

The number of compounds that are active against different number of targets. The last entry indicates that there are a total of 25 compounds that are active against at least 10 different targets and that there are no compounds that are active against more that 41 targets.

that this way of coupling SVM and ranking perceptrons improves the overall performance [37]. We will refer to this approach as SVMRP.

5 Materials

5.1 Datasets

We evaluated the methods proposed in this work using a set of assays derived from a wide variety of databases that store the bioactivity relationship between a target and a set of small chemical molecules or ligands. In particular, these databases provide us target-ligand activity relationship pairs.

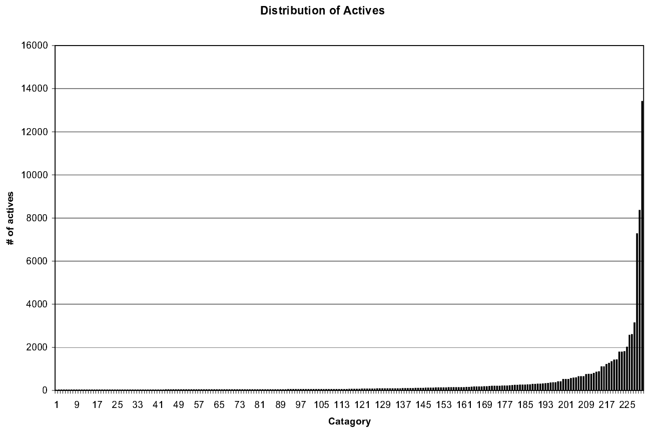

Figure 2: Distribution of Actives in various categories.

We use the PubChem [6] database to extract target-specific dose-response confirmatory assays. For each assay we choose compounds that show the desired activity and confirmed as active by the database curators. We filter compounds that show different activity signals in different experiments against the same targets and they are deemed to be inconclusive and so not used in the study. Duplicate compound entries are removed by comparing the canonical SMILES [2] representations of these molecules. We also incorporate target-ligand pairs from the following databases: BindingDB [42], DrugBank [51], PDSP K_i database [40], KEGG BRITE database [1], and an evaluation sample of the WOMBAT database [34]. All the protein targets that can be mapped to an identifier in PDB database [4] are extracted from this set. We then eliminate all the targets that have $\leq$ 15 actives to ensure that there is some amount of activity information available for each target in our database. Note that a minority of databases report binding affinity between compound and targets (that can be converted to binds to or does not bind to a target) instead of activity. In this work we do not distinguish between the two.

After the above integration and filtering steps our final dataset contains 231 targets, and 27,205 compounds or ligands with a total of 40,170 target-ligand active pairs. Note that certain compounds may show activity in relation with two or more targets as well. Table 1 shows the number of compounds that belong to one or more categories. Out of the 27,205 compounds, 19,154 belong to a single category, 5,363 belong to two categories, 1,697 belong to three categories and the rest of the compounds belong to greater than three categories. It should also be noted that most of these compounds have been experimentally tested for activity against a very small subset of 231 targets. Thus, if a compound belongs to a very small number of categories, it may be simply due to the fact that it has not been experimentally tested against many targets. Figure 2 shows the distribution of actives against the 231 targets in this dataset. Note that the dataset has a skewed distribution wherein most targets have few active compounds (less than hundred) but a small number of targets have a large number active compounds (in thousands).

5.2 Experimental Methodology

All the experiments were performed on dual core AMD Opterons with 4 GB of memory. The following sections will review our experimental methodology to assess the performance of various methods proposed in Section 4.

5.2.1 Descriptor Spaces and Similarity Measures

The similarity between chemical compounds is usually computed by first transforming them into a suitable descriptor-space representation [11, 12]. 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) [2, 3, 23, 38, 47, 50].

In this study we use the ECFP descriptor-space [23, 38] that has been shown to be very effective in the context of classification, ranked-retrieval and scaffold-hopping [47, 48]. We utilize scitegic's Pipeline Pilot [5] to generate ECFP4, a variation of ECFP descriptor-space for our dataset. We also experimented with the GF descriptor-spaces [47] and the results were found to be nearly identical to ECFP. Therefore, we will not discuss results based on GF descriptor-space in this paper.

We use the Tanimoto coefficient [50] (extended Jacquard similarity coefficient) for similarity calculations between chemical compounds represented by the ECFP descriptor-space representation. Specifically, we utilize this measure as a kernel function in SVMR as well as the kernel $\mathcal{K}_A$ in SVM2R (Equations 4.3 and 4.4). We also utilized the cosine similarity to measure similarity between two compounds represented by the L_1 SVM outputs ($\mathcal{K}_B$ in Equations 4.3 and 4.4). Tanimoto coefficient was selected because it has been shown to be an effective way of measuring the similarity between chemical compounds using a sparse descriptor space representation like ECFP [47, 49, 50]. We choose cosine similarity to measure between vectors formed by outputs from SVM classifiers because for this task it achieved slightly better performance as compared to Tanimoto coefficient and variants of Eucledian distance.

5.2.2 Multi-Category Bayesian Predictor

We implemented the multi-category bayesian models as described in [32] and will call this approach BAYESIAN as the original authors call their approach Laplacian-corrected multi-category bayesian models [32]. We compare this scheme to our methods developed in this paper. We have experimented with nearest-neighbor based scheme [31] as well but do not report the results for this scheme as they were found to be consistently inferior to the model-based schemes introduced and compared in this paper.

5.2.3 Training Methodology

For each dataset we separated the compounds into test and training sets, ensuring that the test set is never used during any parts of the learning phase. We split the entire data randomly into ten parts and use nine parts for training and one part for test. We will refer to the nine parts training set as $\mathcal{C}_r$ and the one part test set as $\mathcal{C}_s$.

In order to learn models using BAYESIAN, SVMR, and RP, we utilize the entire set $\mathcal{C}_r$ for training. To learn the models for the cascaded methods (SVM2R and SVMRP) we experiment with an approach that allows us to use the entire training set ($\mathcal{C}_r$) to build both L_1 as well as L_2 models. This approach is motivated by the cross-validation methodology and is similar to that used in previous works [24, 37]. In this approach we further partition the entire training set $\mathcal{C}_r$ into ten equal-size parts. Each part is then predicted using the N first level (L_1) SVM classifiers (one for each target) that are trained on the remaining nine parts. At the end of this process, each training instance in $\mathcal{C}_r$ has been predicted by a set of N binary classifiers, and these prediction outputs serve as training samples for the second-level learning (using the SVMRP or the SVM2R algorithm). Having learned the second-level (L_2) models, we take the entire training set $\mathcal{C}_r$ and retrain the L_1 models. These models are then used to obtain output values (that form the input representation for L_2 models) for the independent test set $\mathcal{C}_s$.

In our setup we use each part ($\mathcal{C}_s$) from the initial ten way split as the test set exactly once. Therefore we have ten different variants of $\mathcal{C}_r$ and $\mathcal{C}_s$. In order to be consistency with the methodology described in [32] for learning BAYESIAN models, we assumed all the compound-target pairs with unknown activity as inactives. Moreover, we did not have inactivity information for many targets in our dataset so it was impossible to model most of our targets using true actives and inactives.

5.2.4 Evaluation Metrics

During the evaluation stage, we compute the prediction for our untouched test dataset. These predictions are then evaluated using the uninterpolated precision [22] and precision/recall values in top-k [8].

To calculate uninterpolated precision for each test compound we utilize the following methodology. For a test compound we obtain top-k ranked targets (using one of the five schemes described in this paper), where k is equal to the number of targets that this test compound is active against. Now, for every correctly predicted target that appears at a position j the top-k predictions we compute precision value at that position. This precision value is defined as the ratio of the number of correct targets identified up to that position j over the total number of targets seen thus far (which is the same as the number j). We calculate this value for every one of the positions in the top-k ranking that corresponds to a correctly predicted target for the given test compound. The final uninterpolated precision value is given by summing up all the precision values obtained above and dividing it by k.

We also calculate precision and recall values in the top one to fifteen ranks of the target for each test compound. This precision value for a test compound is defined as the fraction of correct targets in the top-k ranked targets (where k ranges from one to fifteen). Note that this precision is different from the uninterpolated precision described above. The recall value is defined as the number of correctly predicted targets in the top-k ranked predictions divided by the total number of true targets for a test compound. Note that a high recall value indicates the ability of a scheme to identify a high fraction of true targets for a given compound in top-k ranks. The final values of uninterpolated precision, precision and recall reported in this work are averaged over all the compounds in the test set. We statistically compare the performance of two methods for the task of target identification by using the pairwise student's t-test [10] over the average values obtained for the ten different test sets.

To compare the performance of a set of schemes using precision and recall across the different test sets, we also compute a summary statistic that we refer to as the *Average Relative Quality to the Best (ARQB)*. It is computed as follows: Let $r_{i,j}$ be the precision (recall) value achieved by the scheme j on the test set i, and let r_i^* be the maximum (i.e. the best) precision (recall) value achieved for this test set over all the schemes. Then the ARQB for scheme j is equal to $(1/10)\left(\sum_i (r_{i,j}/r_i^*)\right)$, where we divide by ten to average over the ten test sets. An ARQB value of one indicates that the scheme achieved the best results for all the test sets compared to the other schemes. Furthermore, a high

ARQB value shows that a scheme is consistently close to the top performing scheme and a low ARQB value indicates a poorly performing scheme for the given test sets.

We used pairwise students t-test to determine the statistical significance of the performance difference achieved by a pair of methods over the different cross-validation folds. We used a cut-off value of $p < 0.01$ for statistical significance of the t-test in all comparisons.

5.2.5 Model Selection

The performance of the SVM depends on the parameter that controls the trade-off between the margin and the misclassification cost ("C" parameter in SVMlight [26]), whereas the performance of ranking perceptron depends on the margin β in Algorithm 1.

We perform a model selection or parameter selection step. To perform this exercise fairly, we split our test set into two equal halves of similar distributions, namely sets A and B. Using set A, we vary the controlling parameters and select the best performing model for set A. We use this selected model and compute the accuracy for set B. We repeat the above steps by switching the roles of A and B. The final results are the average of the two runs. While using the SVMlight program we let C take values from the set $\{0.0001, 0.001, 0.01, 0.1, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 15.0, 25.0\}$. While using the perceptron algorithm we let the margin β take values in the set $\{0.00001, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 1.0, 2.0, 5.0, 10.0\}$.

For SVM2R, we experiment with kernel functions described by Equations 4.3 and 4.4. For the kernel function described by equation 4.4 no parameter tuning is required. For the kernel function described by Equation 4.3 we experiment with five different values of the parameter α ($\alpha = 0.2$, 0.4, 0.5, 0.6, and 0.8). Since Equation 4.3 with parameter $\alpha = 0.5$ showed the best performance, we only present results for SVM2R using this kernel and parameter value.

6 Results

In this section, we will evaluate the performance of the BAYESIAN scheme as well as the four methods described in Section 4: SVMR, SVM2R, RP, and SVMRP.

We compare these methods along three directions. The first direction compares the overall performance of these methods with each other on the entire dataset. The second direction compares the effect of the number of categories that a compound belongs to on the results obtained. We utilize uninterpolated precision described in Section 5.2.4 as our metric for these comparisons. The third direction compares these five methods on their ability to retrieve all the relevant categories in the top-k ranks. In order to evaluate and compare the performance of these category ranking algorithms for this task we utilize two measures - precision in top-k and recall in top-k (also described in Section 5.2.4).

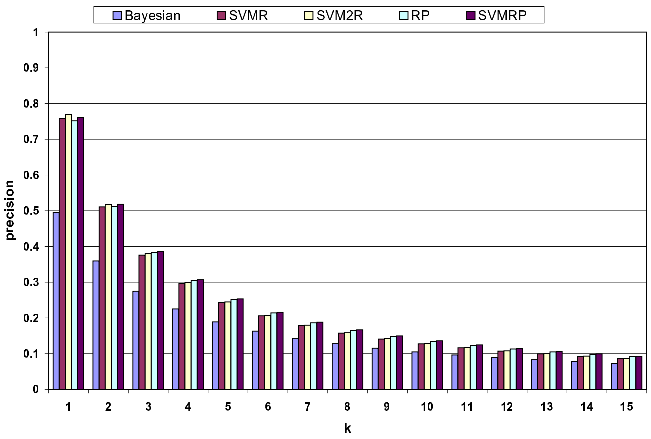

Figure 3: Precision in Top-k

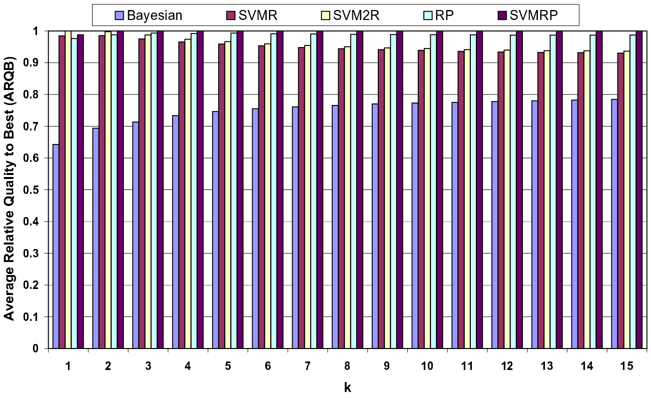

Figure 4: ARQB for Precision in Top-k

6.1 Overall Comparison

Table 2 compares the uninterpolated precision of the five methods using the ECFP4 descriptor-space for each of the ten tests sets.

From Table 2 it can be observed that the best performing scheme over the ten test sets is the SVM2R. This scheme statistically outperforms all the other methods tested in our work over these ten splits. It is followed by SVMRP which is the second best performing method and is statistically better

Table 2: Performance of various schemes on ECFP4 descriptor-space

Test Set #	BAYESIAN	SVMR	SVM2R	RP	SVMRP
1	0.461	0.710	**0.725**	0.716	0.725
2	0.467	0.728	**0.739**	0.724	0.738
3	0.474	0.718	**0.736**	0.726	0.726
4	0.471	0.728	**0.739**	0.728	0.735
5	0.465	0.724	**0.735**	0.716	0.727
6	0.473	0.731	**0.739**	0.723	**0.739**
7	0.473	0.714	**0.728**	0.707	0.716
8	0.465	0.735	**0.747**	0.731	0.735
9	0.468	0.717	**0.733**	0.723	0.728
10	0.471	0.740	**0.749**	0.731	0.741
Avg	0.469	0.725	**0.737**	0.723	0.731

The entry in each cell corresponds to the uninterpolated precision of the column method for a given test set. Entries in bold represent winners for every test set.

Table 3: Performance of various schemes on ECFP4 descriptor-space with compound that belong to L or more categories (targets).

L	Bayesian	SVMR	SVM2R	RP	SVMRP	Statistical significance test, p-value=0.01
≥ 1	0.469	0.725	**0.737**	0.723	0.731	Bayesian $\ll$ (RP,SVMR) $\ll$ SVMRP $\ll$ SVM2R
≥ 2	0.504	0.726	0.736	0.731	**0.742**	Bayesian $\ll$ (SVMR, RP) $\ll$ (RP, SVM2R) $\ll$(SVM2R, SVMRP)
≥ 3	0.488	0.713	0.725	0.724	**0.737**	Bayesian $\ll$ (SVMR, RP) $\ll$ (RP, SVM2R) $\ll$(SVM2R, SVMRP)
≥ 4	0.565	0.723	0.735	0.727	**0.745**	Bayesian $\ll$ (SVMR, RP) $\ll$ (RP, SVM2R, SVMRP)

The entry in each cell corresponds to the uninterpolated precision of the column method averaged over compounds belonging to L or more targets in all the ten test set. Entries in bold represent winners for a given L. Note that $\ll$ indicates that methods on the right are significantly better than the methods on the left, and "( )" indicates that the relationship is not significant. The order of the methods within parentheses represent the order of the weak relationship. Also notice that if a method appears in multiple parentheses (such as RP), the comparison with other methods (such as SVMR,SVM2R) within the parentheses overrides other comparison.

than Bayesian, SVMR, and RP. The next two methods, SVMR and RP show statistically equivalent performance among each other. Finally, all of the above four methods are statistically significantly better than the Bayesian approach.

Table 2 also indicates that the absolute gain in performance achieved by SVM2R over the other methods is not very high (with the exception of its performance over the Bayesian scheme). The gains are relatively modest with SVM2R gaining 0.8% over SVMRP, 2.0% over RP and nearly 1.5% over SVMR on an average. Similarly, SVMRP achieves a gain of only about 0.5% over SVMR as well as about 1.0% over RP. However, the gains are consistent across all the ten test sets highlighting the power of SVM2R and SVMRP in capturing the interclass dependencies.

Finally, it can be observed that RP performs as well as SVMR (a statistically insignificant difference of 0.6% between the two methods). RP is a simple linear learning method that does not guarantee convergence. However, it tries to capture dependencies between the different categories by explicitly trying to rank them in the context of a given test compound. SVMR on the other hand is based on the powerful SVM methodology which employs a sophisticated Tanimoto kernel function shown to be the most effective kernel function for chemical compounds [46,50]. However, it builds independent one-vs-rest classifiers that fail to capture dependencies among different categories and therefore does not perform better than RP.

6.2 Effect of the number of categories

We also investigated the effect of the number of categories (L) that a compound belongs to on the performance of the different methods. This will shed light on the ability of these methods to identify compounds that hit more than one target and may pose problems in terms of toxicity or promiscuity.

Table 3 summarizes the average performance of the five different algorithms over subsets of compounds that belong to L or more targets in the ten test sets utilizing the uninterpolated precision metric. The first row in this table corresponds to $L \geq 1$, which is the set of all the compounds in the dataset. Therefore the average results in the first row of Table 3 are identical to the averages reported in Table 2. The subsequent rows show the performance of different methods over compounds that belong to two or more categories.

From this table it can be observed that, in general, as the number of targets (L) that a compound belongs to increases, the methods that try to capture dependencies among the different categories (SVM2R, RP, and SVMRP) perform better than the methods that do not (Bayesian, SVMR). Moreover, ranking perceptron based methods (RP and SVMRP) show more improvement in their performance as compared to SVM2R. Specifically, as L increases from one to four, the statistical edge of SVM2R over the two ranking perceptron based schemes disappears. Finally, simple sorting based scheme SVMR which performs slightly better than RP in terms of absolute performance for $L \geq 1$ performs worse than RP for $L \geq 2, 3, 4,$ and 5 (although all of these differences are not statistically significant).

Overall, these results indicate that the schemes that capture interclass dependencies tend to outperform schemes that do not for compounds that belong to more than one category.

6.3 Ability to retrieve all relevant categories

In this section we compare the ability of the five methods to identify relevant categories in the top-k ranks. We analyze the results along this direction because this directly corresponds to the use case scenario where a user may want to look at top-k predicted targets for a test compound and further study or analyze them for toxicity, promiscuity, off-target effects, pathway analysis *etc*. If all the true targets fall in the top-k ranks, there is a high likelihood of successfully recognizing them for the above analysis.

For this comparison we utilize precision and recall metric in top-k for each of these schemes as shown in Figures 3–6. Figures 3 and 5 show the actual precision and recall values obtained for the five methods whereas Figures 4 and 6 show the corresponding ARQB values for precision and recall respectively (ARQB is described in Section 5.2.4). We analyze these results by obtaining precision and recall values

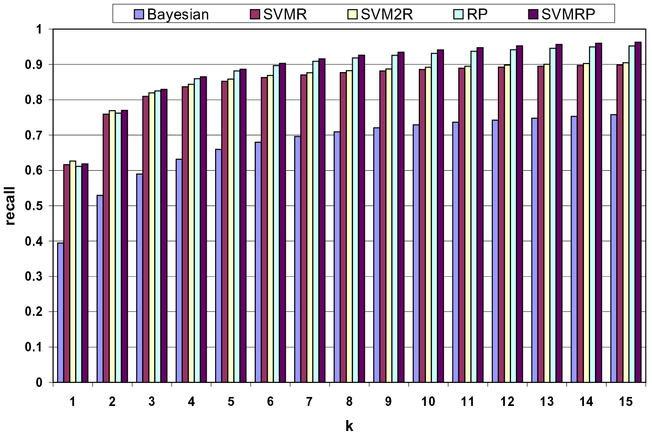

Figure 5: Recall in Top-k

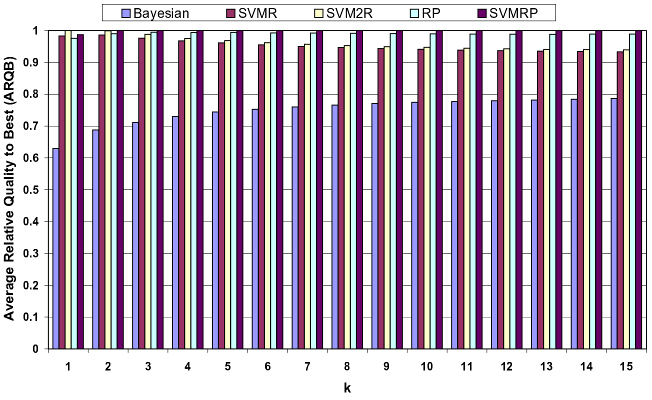

Figure 6: ARQB for Recall in Top-k

in the top-k ranks by varying k from one to fifteen. These figures are obtained by averaging precision and recall results over the test sets used in this study. Therefore, these results are averaged over all the 27,205 compounds present in the dataset.

A number of trends can be observed from these figures. First, for identifying one of the correct categories or targets in the top 1 predictions, SVM2R statistically outperforms all the other schemes in terms of both precision and recall. This is followed by SVMRP, RP and SVMR in that order. Second, BAYESIAN is still the worst performing method among the five compared and the performance order of these schemes is exactly the same as the performance order for the uninterpolated results in Table 2.

However, as k increases from one to fifteen, the precision and recall results indicate that the best performing scheme is now SVMRP and it statistically outperforms all other schemes for both precision as well as recall. This is followed by RP, which statistically outperforms the other three schemes (BAYESIAN, SVMR, SVM2R) for both precision and recall in the top fifteen. The performance of RP is followed by SVM2R, SVMR, and finally BAYESIAN. The ranking perceptron based methods achieve an average recall of approximately 0.96 and 0.95 for SVMRP and RP respectively for $k = 15$ and is statistically better than the other schemes for the ten test sets (with average recall rates of 0.90, 0.89, and 0.76 respectively for SVM2R, SVMR, and BAYESIAN). Moreover, ARQB values in figure 6 show that as k increases from one to fifteen, ranking perceptron based schemes start performing consistently better that others in identifying all the correct categories. The two ranking perceptron based schemes also achieve average precision values that are statistically significantly better than other schemes in the top fifteen (Figures 3 and 4).

In summary, these result indicate that ranking perceptron based methods because they have a higher recall will tend to find more of the correct categories in top ranks than other schemes. Thus, these methods present a better chance of finding and analyzing targets in the context of a chemical compound.

7 Discussion and Conclusion

In this paper we describe a number of techniques that are aimed at solving the problem of target identification. We addressed this problem by formulating it as a category (target) ranking problem and proposed effective solutions based on ranking perceptron and SVM. Specifically, we proposed methods based on SVM and ranking perceptrons to capture interclass dependencies for this problem. Extensive experiments and analysis comparing our methods to other previously developed methods for target identification showed that our methods are either as good as or superior to the current state-of-the-art.

However, a number of issues still need to be addressed. First, in this work we assumed the compound with no activity information against a target to be inactive against that target. The primary reason for such assumption was almost no inactive data available for many of the targets in our dataset collected from publicly available sources. Similar assumption has been made by previous studies [32]. However, this is not a very satisfactory assumption from the point of view of drug discovery as it may result in the method missing some rare compound-target activity. Therefore, in our future work, we will try to address the issue of unknown compound-target activity explicitly in order to come up with practical methods for drug discovery.

Second, in this work we have not utilized target category information (for example target sequence, structure, family) in building these ranking perceptron or SVM based models. Identification of key characteristics common across two target themselves (similar geometry of binding sites, similar biochemical characteristics of binding residues, *etc*) might identify two targets that will likely bind to the same compound. Therefore, effective solutions can be devised using both SAR data and target information. A number of recent approaches for chemogenomics utilize SAR data as well as target information to build predictive models on the target-ligand graph

[35,52]. Our initial studies in trying to include target information did not yield promising results for the problem of target identification. They were found to be no better as compared to only SAR data based approaches proposed in this paper. So we did not pursue approaches that include target information in this work. However, we believe that exploring a good way of including target information is a worthwhile effort and plan to investigate it rigorously as a part of our future work.

Finally, recent approaches have shown that the interclass dependencies could be learned within structural learning framework that utilizes structural SVM [24, 37, 44]. We also experimented with this approach in our work using SVMstruct algorithm [44]. However, our preliminary results showed that this approach did not yield any significant gains over the ranking perceptron based approach over the large number of categories in our domain. Moreover, the computational cost of employing structural learning was much higher than that of utilizing ranking perceptrons. Therefore we did not pursue structural SVM based approach in the present work.

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.genome.jp/kegg/brite.html.

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

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

[4] http://www.pdb.org/. *RCSB Protein Data Bank.*

[5] http://www.scitegic.com. *Scitegic Inc.*

[6] pubchem.ncbi.nlm.nih.gov. *The PubChem Project.*

[7] K. Azzaoui, J. Hamon, B. Faller, S. Whitebread, E. Jacoby, A. Bender, J. L. Jenkins, and L. Urban. Modeling promiscuity based on in vitro safety pharmacology profiling data. *ChemMedChem*, 2(6):874–880, Jun 2007.

[8] R. Baeza-Yates and B. Ribeiro-Neto. Modern information retrieval. *Addison Wesley 1999.*

[9] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. *ICML*, pages 9–17, 2004.

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

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

[12] G. Bravi, E. Gancia, D. Green, V. Hann, and M. Mike. Modelling structure-activity relationship. *Virtual Screening for Bioactive Molecules*, 2000.

[13] Y. Z. Chen and C. Y. Ung. Prediction of potential toxicity and side effect protein targets of a small molecule by a ligand-protein inverse docking approach. *J Mol Graph Model*, 20(3):199–218, 2001.

[14] Y. Z. Chen and D. G. Zhi. Ligand-protein inverse docking and its potential use in the computer search of protein targets of a small molecule. *Proteins*, 43(2):217–226, May 2001.

[15] M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures and the voted perceptron. *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics*, pages 263–270, 2002.

[16] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[17] K. Crammer and Y. Singer. A new family of online algorithms for category ranking. *Journal of Machine Learning Research.*, 3:1025–1058, 2003.

[18] U. S. Eggert and T. J. Mitchison. Small molecule screening by imaging. *Curr Opin Chem Biol*, 10(3):232–237, Jun 2006.

[19] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. *PAKDD.*, pages 22–30, 2004.

[20] C. Hansch, T. F. P. P. Maolney, and R. M. Muir. Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature*, 194:178–180, 1962.

[21] C. P. Hart. Finding the target after screening the phenotype. *Drug Discov Today*, 10(7):513–519, Apr 2005.

[22] M. Hearst and J. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. *ACM/SIGIR*, 1996.

[23] 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.

[24] E. Ie, J. Weston, W. S. Noble, and C. Leslie. Multi-class protein fold recognition using adaptive codes. In *ICML*, 2005.

[25] J. L. Jenkins, A. Bender, and J. W. Davies. In silico target fishing: Predicting biological targets from chemical structure. *Drug Discovery Today: Technologies*, 3(4):413–421, 2006.

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

[27] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matricies. *J. Mol. Biol.*, 292:195–202, 1999.

[28] G. Karypis. Yasspp: Better kernels and coding schemes lead to improvements in svm-based secondary structure prediction. *Proteins: Structure, Function and Bioinformatics*, 64(3):575–586, 2006.

[29] K. Kawai, S. Fujishima, and Y. Takahashi. Predictive activity profiling of drugs by topological-fragment-spectra-based support vector machines. *J. Chem. Info. Model.*, 48(6):1152–1160, 2008.

[30] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-denite programming. *ICML*, pages 323–330, 2002.

[31] J. H. Nettles, J. L. Jenkins, A. Bender, Z. Deng, J. W. Davies, and M. Glick. Bridging chemical and biological space: "target fishing" using 2d and 3d molecular descriptors. *J Med Chem*, 49(23):6802–6810, Nov 2006.

[32] Nidhi, M. Glick, J. W. Davies, and J. L. Jenkins. Prediction of biological targets for compounds using multiple-category bayesian models trained on chemogenomics databases. *J Chem Inf Model*, 46(3):1124–1133, 2006.

[33] T. Niwa. Prediction of biological targets using probabilistic neural networks and atom-type descriptors. *J Med Chem*, 47(10):2645–2650, May 2004.

[34] M. Olah, M. Mracec, L. Ostopovici, R. Rad, A. Bora, N. Hadaruga, I. Olah, M. Banda, Z. Simon, M. Mracec, and T. Oprea. Wombat: World of molecular bioactivity. *Chemoinformatics in Drug Discovery. Wiley-VCH, New York*, pages 223–239, 2004.

[35] K. Park and D. Kim. Binding network similarity of ligand. *Proteins*, 71:960–971, 2008.

[36] J. C. Platt. Advances in kernel methods: Support vector learning. 1999.

[37] H. Rangwala and G. Karypis. Building multiclass classifiers for remote homology detection and fold recognition. *BMC Bioinformatics*, 7:455–471, 2006.

[38] 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.

[39] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[40] B. L. Roth, W. K. Kroeze, S. Patel, and E. Lopez. The multiplicity of serotonin receptors: Uselessly diverse molecules or an embarrasment of riches? *The Neuroscientist*, 6:252–262, 2000.

[41] F. Sams-Dodd. Target-based drug discovery: is something wrong? *Drug Discov Today*, 10(2):139–147, Jan 2005.

[42] Y. L. T. Liu, X. Wen, R. Jorrisen, and M. K. Gilson. Bindingdb: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research 00(Database Issue)*, pages D1–D4, 2006.

[43] G. C. Terstappen, C. Schlpen, R. Raggiaschi, and G. Gaviraghi. Target deconvolution strategies in drug discovery. *Nat Rev Drug Discov*, 6(11):891–903, Nov 2007.

[44] I. Tsochantaridis, T. Homann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[45] V. Vapnik. Statistical learning theory. *John Wiley, 1998.*

[46] N. Wale and G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *IEEE ICDM*, pages 678–689, 2006.

[47] N. Wale, I. A. Watson, and G. Karypis. Comparison of descriptor spaces for chemical compounds retrieval and classification. *Journal of Knowledge and Information Systems*, 14(3):347–375, 2008.

[48] N. Wale, I. A. Watson, and G. Karypis. Indirect similarity based methods for effective scaffold-hopping in chemical compounds. *J. Chem. Info. Model.*, 48(4):730–741, 2008.

[49] M. Whittle, V. J. Gillet, and P. 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.

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

[51] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucl. Acids Res.*, 34(suppl1):D668–672, 2006.

[52] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa. Prediction of drug-target interaction networks from integration of chemical and genomics spaces. *Bioinformatics*, 24:232–240, 2008.