PROTEIN-CHEMICAL INTERACTION PREDICTION VIA KERNELIZED SPARSE LEARNING SVM

YI SHI*[1], XINHUA ZHANG[1], XIAOPING LIAO[2], GUOHUI LIN[1], DALE SCHUURMANS[1]

[1]*Department of Computing Science, University of Alberta,*
Edmonton, Alberta T6G 2E8, Canada
E-mail: {ys3,xinhua2,guohui,daes}@ualberta.ca
[2]*Department of Agricultural, Food and Nutritional Science, University of Alberta,*
Edmonton, Alberta T6G 2P5, Canada
E-mail: xliao2@ualberta.ca

Given the difficulty of experimental determination of drug-protein interactions, there is a significant motivation to develop effective *in silico* prediction methods that can provide both new predictions for experimental verification and supporting evidence for experimental results. Most recently, classification methods such as support vector machines (SVMs) have been applied to drug-target prediction. Unfortunately, these methods generally rely on measures of the maximum "local similarity" between two protein sequences, which could mask important drug-protein interaction information since drugs are much smaller molecules than proteins and drug-target binding regions must comprise only small local regions of the proteins. We therefore develop a novel sparse learning method that considers sets of short peptides. Our method integrates feature selection, multi-instance learning, and Gaussian kernelization into an L_1 norm support vector machine classifier. Experimental results show that it not only outperformed the previous methods but also pointed to an optimal subset of potential binding regions. Supplementary materials are available at "`www.cs.ualberta.ca/~ys3/drug_target`".

Keywords: Drug-target interaction; SVM; Sparse learning; Kernelization.

1. Introduction

Proteins operate in highly interconnected networks ("interactome networks") that play a central role in governing cell functions. If a protein's conformation is changed, its function can be altered, thus affecting cell function. Drugs are small molecules that bind to target proteins to intensionally change the protein conformation, ultimately achieving treatment effects. The function of many classes of pharmaceutically useful protein targets, such as enzymes, ion channels, G protein coupled receptors (GPCRs), and nuclear receptors, can be modulated by ligand interaction. Identifying interaction between ligands and proteins is therefore a key to genomic drug discovery.

Various high-throughput technologies for analyzing the genome, the transcriptome, and the proteome have enhanced our understanding of the space populated by protein classes. Meanwhile, the development of high-throughput screening technology has enabled broader exploration of the space of chemical compounds.[1–3] The goal of the chemical genomics research is to identify potentially useful compounds, such as imaging probes and drug leads, by relating the chemical space to the genomic space. Unfortunately, our understanding of the relationship between the chemical and the genomic spaces remains insufficient. For example, the PubChem database at NCBI[4] contains information of millions of chemical compounds, but the number of compounds with known target proteins is limited. The lack of documented protein-chemical interactions suggests that many remain to be discovered, which motivates the need for improved methods for inferring potential drug-target interactions automatically and efficiently. To facilitate the study of protein-chemical interactions, Kuhn et al. created a protein-chemical interaction database called STITCH,[5] which, up to now, contains interactions for between 300,000 small molecules and 2.6 million proteins from 1,133 organisms.

By elucidating the interaction between proteins and drug molecules, 3D-structure based "docking analysis" has been the principle method for drug discovery.[6–8] In docking analysis, drug-protein binding affinities are modeled by non-covalent intermolecular interactions, such as hydrogen bonding, electrostatic interactions, hydrophobic and Van der Waals forces. Through establishing equations that model the physical interaction between a receptor and potential ligand, the potential energy of binding can be calculated. There are many docking software tools available, including DOCK,[8] GOLD,[6] and AutoDock.[7] All these methods require complete 3D structural information for the target, which might not be available in practice. Such a major disadvantage makes docking analyses infeasible for genome wide application.

Given the difficulty of experimental determination of compound-protein interactions,[9,10] there is a significant motivation to develop effective *in silico* prediction methods that can provide both new predictions for experimental verification and supporting evidence for experimental results. To predict compound-protein interactions various computational approaches have been developed. Keiser et al.[11] propose using the known structure of a set of ligands to predict target protein families. This method does not take advantage of available protein sequence information, and is thus limited to those between known ligands and protein families. Campillos et al.[12] propose predicting drug-target interaction based on similarities between side-effects of known drugs. Some results of this approach have been verified by *in vitro* binding assays, but the approach remains limited to predictions involving drugs with known side-effects. Yamanishi et al.[13] have investigated the relationship between drug chemical structure, target protein sequence, and drug-target network topology, and developed a regression-based learning method for predicting unknown drug-target interactions. In particular, they integrated the chemical and the genomic spaces into a unified space, referred to as the "pharmacological space", wherein chemical-chemical, protein-protein, and chemical-protein similarities can be modeled. Perlman et al.[14] used a combination of Smith-Waterman score, protein-protein interaction, and Gene Ontology information to measure the gene-gene similarity (similarity between targets), but these ancillary information is not always available making the prediction hard to extend to general case, and the way of combining different information sources is somehow tricky.

Most recently, classification methods have been adopted in drug-target prediction.[15–17] These methods firstly calculate the similarities between targets and/or drugs, then use these similarities to construct kernel matrices for the classifiers, such as the support vector machines (SVMs) for predicting novel drug-target interactions. The prediction can be cast into two ways, one for drug side or drug-to-target and the other for target side. For drug-to-target prediction, drug-drug similarities are first obtained, based on structural or pharmacological information; then a bipartite known drug-target interaction graph is constructed; for a new drug with known structural or pharmacological information, its similarities to known drugs are calculated to predict its interactions with known targets using the bipartite interaction graph.

Similarly for target-to-drug prediction, target-target similarities are first obtained using the primary amino acid sequences;[13,17,18] then for a new target with known primary sequence, its similarities to known targets are calculated to predict its interactions with known drugs again using the bipartite interaction graph.

It should be pointed out that in the state-of-the-art works of target-to-drug prediction, the target-target similarity is defined out of the normalized Smith-Waterman score.[17] This S-W score measures the maximum "local similarity" between two protein sequences,[19] thus reasonable, but the local similarity still uses the whole sequences and consequently might involve *long* substrings, which is unreasonable. In fact, long substrings could mask important interaction information, since drugs are usually much smaller molecules than proteins and the drug-target binding sites mostly comprise of only small local regions of the target.

In this work, we focus on the latter target-to-drug prediction to address the issues in the existing works. We first attempt to identify key local binding regions from the *common short* substrings shared by proteins that interact with the same drug. These key short substrings are then used to construct a vector representation for a protein sequence, to be used in the training and testing phases of a classifier. The use of key short substrings (i.e. potential binding regions) as features for the targets is a more direct and meaningful representation for drug interaction prediction. Additionally, the explicit vector representation of targets, as opposed to assessing similarity based on the S-W score, maps the targets into higher dimensional spaces, thus increasing the effectiveness of kernel-based classifiers. We remark that our use of common short substrings differ from the substring composition representation for proteins,[15] which uses all substrings while disregarding whether interactions exist.

The rest of the paper is organized as follows. In Section 2, we introduce the details of our prediction method, in which we focus on the SVM classifiers. We demonstrate in Section 3 the performance of our method compared against the existing ones. Lastly, in Section 4, we discuss the advantages and disadvantages of our method and propose future work.

2. Methods

The drug-target interaction prediction framework is the same as in Bleakley et al.,[17] in which we assume a dataset containing m drugs $d_1, d_2, \ldots, d_m$ and n targets $t_1, t_2, \ldots, t_n$, and the binary indicator on whether or not drug d_i interacts target t_j. The goal is to predict which of the drugs a new target t_c will interact.

2.1. *Target Vectorization*

In the *bipartite local model* (BLM) by Bleakley et al.,[17] to which our method will compare against, the similarity between two targets t and t' is defined as the normalized Smith-Waterman score:[17]

$$s(t, t') = \frac{SW(t, t')}{\sqrt{SW(t, t)}\sqrt{SW(t', t')}}, \tag{1}$$

where $SW(\cdot, \cdot)$ denotes the original Smith-Waterman score.[19] As we mentioned in the introduction, such a similarity measure might overlook the key short sequence regions to which a drug binds.

To address this issue, we want to identify the common short substrings of the targets that interact the same drug. We consider one drug, say d_i, at a time. From the dataset, we first retrieve the set of targets $T_i = \{t_{i1}, t_{i2}, \ldots, t_{in_i}\}$ interacting with d_i. By including the new target t_c, we obtain another set $T_i \cup \{t_c\}$. Using a substring length lower bound, we compute for each of the two sets T_i and $T_i \cup \{t_c\}$ the multi-set of pairwise maximal common substrings, denoted as *withoutSS* $= \{s_{i1}, s_{i2}, \ldots, s_{iq'}\}$ and *withSS* $= \{s_{i1}, s_{i2}, \ldots, s_{ip'}\}$, respectively. In each of the two multi-sets, if two substrings differ at at most one position, they are merged into one and their frequencies are summed together. This way, we obtain two *reduced* sets *withoutSS* $= \{s_{i1}, s_{i2}, \ldots, s_{iq}\}$ and *withSS* $= \{s_{i1}, s_{i2}, \ldots, s_{ip}\}$, containing q and p unique substrings respectively, and each substring is associated with its number of occurences.

Using the substrings in set *withSS* and their occurrences, we can map the n training targets and the new target t_c into the p dimensional Euclidean space, where each substring represents a dimension and the coordinate of target t in dimension s is calculated as the normalized match score between t and s in set *withSS*:

$$M(t, s) = \frac{L(t, s) \cdot c_s}{\sum_{i=1}^{p} c_{s_i}} \ , \tag{2}$$

where $L(\cdot, \cdot)$ is length of the longest common substring between the two sequences and c_s is the number of occurrence of substring s. Intuitively, if target t_c contains a long substring that is also frequent in the binding targets, then its match score for this feature substring will be high indicating a high likelihood of binding. We use $(M(t, s_1), M(t, s_2), \ldots, M(t, s_p))$ as the vector representation for target t.

This way we obtain an $n \times p$ training matrix X, where each row represents a training target, and a $p \times 1$ testing vector $\mathbf{x}_c$ representing the new target t_c, along with the $n \times 1$ binary training label vector $\mathbf{y}$ (with 1 indicating the target interacts with drug d_i and -1 otherwise). The task is to construct a classifier to return 1 if the new target t_c interacts with drug d_i, or -1 otherwise.

The classification problem can be analogously formulated using set *withoutSS* substring set. Next we show how to construct a classifier from the training data.

2.2. *Classification with Feature Selection*

In any classification problem, the quality of features used determines the accuracy of predictions. Here, features correspond to substrings of target proteins, which comprise potential binding regions between the proteins and drugs. Thus, selecting good features not only improves classification accuracy, but also provides candidate drug-target binding sites for further investigation. We investigated an approach that integrates feature selection in L_1-norm based support vector machine (SVM) classification method.

The primal form of L_1-norm SVM is:

$$\begin{array}{rc} & \min\limits_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \beta\|\mathbf{w}\|_1 + \mathbf{1}^T\boldsymbol{\xi} \\ \textbf{s.t.} : & \boldsymbol{\xi} \geq \mathbf{1} - \triangle(\mathbf{y})(X\mathbf{w} - b\mathbf{1}), \\ & \boldsymbol{\xi} \geq \mathbf{0}. \end{array} \tag{3}$$

where $\triangle(\mathbf{y})$ denotes putting the vector $\mathbf{y}$ on the main diagonal of a square matrix. Here

$X \in \mathbb{R}^{n\times p}$, $\mathbf{y} \in \{+1,-1\}^n$, n is the number of data points (targets), and p is the number of features. Since by Micchelli et al.[20]

$$\|\mathbf{w}\|_1 = \min_{\gamma\geq 0} \frac{1}{2}\sum_j \left(\frac{w_j^2}{\gamma_j} + \gamma_j\right) = \min_{\gamma\geq 0} \frac{1}{2}(\mathbf{w}^T\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} + \boldsymbol{\gamma}^T\mathbf{1}),$$

so (3) becomes

$$\begin{array}{rc} \min\limits_{\mathbf{w},\mathbf{b},\boldsymbol{\xi},\boldsymbol{\gamma}} & \frac{\beta}{2}(\mathbf{w}^T\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} + \boldsymbol{\gamma}^T\mathbf{1}) + \mathbf{1}^T\boldsymbol{\xi} \\ \textbf{s.t.}: & \boldsymbol{\xi} \geq \mathbf{1} - \triangle(\mathbf{y})(X\mathbf{w} - b\mathbf{1}), \\ & \boldsymbol{\xi} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}. \end{array} \tag{4}$$

By introducing Lagrangian multipliers $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\mu} \geq \mathbf{0}$, (4) becomes

$$\begin{array}{rc} \min\limits_{\mathbf{w},\mathbf{b},\boldsymbol{\xi},\boldsymbol{\gamma}} \max\limits_{\boldsymbol{\lambda},\boldsymbol{\mu}} & \frac{\beta}{2}(\mathbf{w}^T\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} + \boldsymbol{\gamma}^T\mathbf{1}) + \mathbf{1}^T\boldsymbol{\xi} + \boldsymbol{\lambda}^T(\mathbf{1} - \triangle(\mathbf{y})(X\mathbf{w} - b\mathbf{1}) - \boldsymbol{\xi}) - \boldsymbol{\mu}^T\boldsymbol{\xi} \\ \textbf{s.t.}: & \boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}. \end{array} \tag{5}$$

Let the objective function of (5) be L_1, and let $\frac{\partial L_1}{\partial \boldsymbol{\xi}} = 0$, we get $\boldsymbol{\lambda} = \mathbf{1} - \boldsymbol{\mu}$. Therefore, since $\boldsymbol{\mu} \geq \mathbf{0}$, we conclude that $\boldsymbol{\lambda} \leq \mathbf{1}$, hence $\mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}$. By substitution, (5) becomes

$$\begin{array}{rc} \min\limits_{\mathbf{w},\mathbf{b},\boldsymbol{\gamma}} \max\limits_{\boldsymbol{\lambda}} & \frac{\beta}{2}(\mathbf{w}^T\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} + \boldsymbol{\gamma}^T\mathbf{1}) + \boldsymbol{\lambda}^T\mathbf{1} - \boldsymbol{\lambda}^T\triangle(\mathbf{y})X\mathbf{w} + b\boldsymbol{\lambda}^T\triangle(\mathbf{y})\mathbf{1} \\ \textbf{s.t.}: & \mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}, \\ & \boldsymbol{\gamma} \geq \mathbf{0}. \end{array} \tag{6}$$

Let the objective function of (6) be L_2, and let $\frac{\partial L_2}{\partial b} = 0$. We get $\boldsymbol{\lambda}^T\mathbf{y} = 0$, so (6) becomes

$$\begin{array}{rc} \min\limits_{\mathbf{w},\boldsymbol{\gamma}} \max\limits_{\boldsymbol{\lambda}} & \frac{\beta}{2}(\mathbf{w}^T\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} + \boldsymbol{\gamma}^T\mathbf{1}) + \boldsymbol{\lambda}^T\mathbf{1} - \boldsymbol{\lambda}^T\triangle(\mathbf{y})X\mathbf{w} \\ \textbf{s.t.}: & \mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}, \\ & \boldsymbol{\lambda}^T\mathbf{y} = 0, \\ & \boldsymbol{\gamma} \geq \mathbf{0}. \end{array} \tag{7}$$

Let the objective function of (7) be L_3, and let $\frac{\partial L_3}{\partial \mathbf{w}} = 0$, we get $\beta\triangle(\boldsymbol{\gamma})^{-1}\mathbf{w} - X^T\triangle(\mathbf{y})\lambda = \mathbf{0}$, so that $\mathbf{w} = \frac{1}{\beta}\triangle(\boldsymbol{\gamma})X^T\triangle(\mathbf{y})\boldsymbol{\lambda}$. By substitution, (7) becomes

$$\begin{array}{rc} \min\limits_{\boldsymbol{\gamma}} \max\limits_{\boldsymbol{\lambda}} & \boldsymbol{\lambda}^T\mathbf{1} - \frac{1}{2\beta}\boldsymbol{\lambda}^T\triangle(\mathbf{y})X\triangle(\boldsymbol{\gamma})X^T\triangle(\mathbf{y})\boldsymbol{\lambda} + \frac{\beta}{2}\boldsymbol{\gamma}^T\mathbf{1} \\ \textbf{s.t.}: & \mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}, \\ & \boldsymbol{\lambda}^T\mathbf{y} = 0, \\ & \boldsymbol{\gamma} \geq \mathbf{0}. \end{array} \tag{8}$$

Note that $\boldsymbol{\gamma}$ is the feature selection vector. Crucially, this problem is convex in $\boldsymbol{\gamma}$ and has no local minima,[21] hence it provides an optimal form of feature selection that can be efficiently obtained in conjunction with SVM training. Because a drug may bind to different regions of different proteins, i.e., different regions on different targets can bind to the same drug, each positive data point may correspond to a different set of important features (substrings). Therefore, the nature of this drug-target classification problem is essentially a multi-instance classification problem. To address this, we consider two ideas:

Idea (a) Use a radial basis function (RBF) kernel (Gaussian kernel), rather than a linear kernel since this addresses the multi-instance classification problem more effectively after implicitly mapping data points to an infinite dimensional space. After Gaussian kernelization, the original linear kernel matrix $K = X\triangle(\boldsymbol{\gamma})X^T$ becomes $K'_{ij} = e^{\frac{-1}{2\sigma^2}(\mathbf{x}_i-\mathbf{x}_j)^T\triangle(\boldsymbol{\gamma})(\mathbf{x}_i-\mathbf{x}_j)}$.

Idea (b) Because each positive data point may correspond to a unique set of important features, in principle each positive example $\mathbf{x}_i$ should employ its own feature selection vector $\boldsymbol{\gamma}_i^+$ while all negative examples should share a same vector $\boldsymbol{\gamma}^-$. So we get $K''_{ij} = e^{\frac{-1}{2\sigma^2}\|\boldsymbol{\gamma}_i\odot\mathbf{x}_i-\boldsymbol{\gamma}_j\odot\mathbf{x}_j\|^2}$ for all i and j, where $\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_i^+$ if $y_i = +1$, and $\boldsymbol{\gamma}_i = \boldsymbol{\gamma}^-$ if $y_i = -1$. Here $\odot$ stands for element-wise multiplication.

Idea (a) can be easily applied to (8) at the sacrifice of convexity, while applying Idea (b) to (8) will introduce too many extra coefficients which makes the model computationally expensive. To circumvent these issues, we introduce an efficient approach to re-weight the features. Intuitively, we wish to down-weight the features that are false positive indicator of binding, i.e. features that have a high score/value at some negative training examples (not bind). This motivation is similar to the case in multi-instance learning, where false positive indicators call for more strict control than true positive indicators. Towards this end, we introduce a p-dimensional weight vector $\mathbf{c}$ corresponding to the p features, and re-scale the feature matrix X by $\tilde{X} = X\triangle(\mathbf{c})$. A simple formula of $\mathbf{c}$ that concretizes our intuition is $c_j = \frac{1}{n}\sum_i a_{ij}$, where $a_{ij} = 1$ if $x_{ij} \leq 1-\epsilon$ and $y_i = 1$, and $a_{ij} = 0$ otherwise. Here ϵ is a small positive number, and all elements in X are assumed to have been normalized to $[0, 1]$. Therefore by replacing X with $\tilde{X}$ in (8), we encourage using features that indicate less false positive, and formally we obtain

$$\begin{array}{rl} \min\limits_{\boldsymbol{\gamma}} \max\limits_{\boldsymbol{\lambda}} & \boldsymbol{\lambda}^T\mathbf{1} - \frac{1}{2\beta}\boldsymbol{\lambda}^T\triangle(\mathbf{y})K'\triangle(\mathbf{y})\boldsymbol{\lambda} + \frac{\beta}{2}\boldsymbol{\gamma}^T\mathbf{1} \\ \textbf{s.t.}: & \mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}, \\ & \boldsymbol{\lambda}^T\mathbf{y} = 0, \\ & \boldsymbol{\gamma} \geq \mathbf{0}, \end{array} \tag{9}$$

where $K'_{ij} = \exp\left(\frac{-1}{2\sigma^2}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T\triangle(\boldsymbol{\gamma})(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)\right)$.

We solve (9) by using a combination of L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno Bounded Optimization) and gradient decent method over $\boldsymbol{\gamma}$. After optimization, we get solutions for $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$. $\boldsymbol{\gamma}$ serves as a useful feature selector, with $\gamma_j > \epsilon$ indicating the j's features should be selected and otherwise not. $\boldsymbol{\lambda}$ can be used to construct the hyperplane in the SVM and to predict new data points. Given a test data point (target) $\mathbf{x}_c$, we can predict its label (binding to the drug or not) based on the sign of the classifier's output:

$$y_c = \sum_{i=1}^{n} \lambda_i y_i \exp\left(\frac{-1}{2\sigma^2}(\tilde{\mathbf{x}}_c - \tilde{\mathbf{x}}_i)^T\triangle(\boldsymbol{\gamma})(\tilde{\mathbf{x}}_c - \tilde{\mathbf{x}}_i)\right) - b. \tag{10}$$

As a key step for solving (9), we need the partial derivative of the objective function in (9) (denoted as L_4) with respect to the k's feature selector γ_k:

$$\frac{\partial L_4}{\partial \gamma_k} = \frac{1}{2\beta} \sum_{ij} \lambda_i \lambda_j y_i y_j \frac{\partial K'_{ij}}{\partial \gamma_k} + \frac{\beta}{2},$$

where

$$\frac{\partial K'_{ij}}{\partial \gamma_k} = K'_{ij} \Big[\frac{-1}{2\sigma^2} (\tilde{x}_{ik} - \tilde{x}_{jk})^2 \Big] = \exp \left(\frac{-1}{2\sigma^2} (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T \triangle(\gamma) (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T \right) \Big[\frac{-1}{2\sigma^2} (\tilde{x}_{ik} - \tilde{x}_{jk})^2 \Big].$$

3. Experimental Results and Discussion

3.1. *Methods under Comparison*

We compared our method with the state-of-the-art method proposed by Bleakley et al.[17] In particular, we focused on target-side prediction of their method to make the two approaches comparable. Bleakley et al.[17] used the normalized Smith-Waterman score in (1) to evaluate the similarity between two target sequences. In the context of SVM classification, they used this target-target similarity matrix as the kernel matrix, i.e. the kernel matrix was fixed in their method. Based on this similarity measure, nearest neighbor (NN) classifiers can also be constructed as a baseline. We refer to Bleakley et al.'s approach as BLM_SVM and BLM_NN respectively. On the other hand, our methods include:

- SS_L1-SVM: L1-SVM with *withSS* feature (the main model of this paper),
- SS_L1-SVM: the classic L2 norm SVM with *withSS* feature,
- SS_NN_FS: nearest neighbor classifier based on the features selected by SS_L1-SVM,
- SS_NN_noFS: nearest neighbor classifier based on all *withSS* features.

3.2. *Experiment Settings*

The framework of our experiment is similar to Bleakley et al.[17] Specifically, we enumerated all pairs of drug d_i and protein t in the whole data set. For each (d_i, t) pair, we treated t as the single test example while the remaining proteins were used as training examples. To learn an L1 and L2 SVM, we chose the hyper-parameters (e.g. β and ϵ) by using three-fold cross validation on the training set, making sure that all the three folders contain at least one target that binds to the drug (i.e., at least one positive example). After the classification model was learned, we applied it to protein t in a way like (10), and obtained a score y_{it} that could be subsequently used to compute useful performance measures (see Section 3.4). All y_{it} calculated by ranging over all drugs d_i and target t in the data set constituted a drug-by-target score table.

We set the minimum length of a feature sub-sequence to 5 after trying all lengths from 4 to 12, noting that a too small cutoff generated excessively many features while a too big cutoff generated an insufficient number of features.

3.3. *Datasets*

We used drug-target interaction information from Bleakley et al.,[17] which was collected from the KEGG BRITE,[2] BRENDA,[22] SuperTarget[23] and DrugBank[24] databases. In particular, we

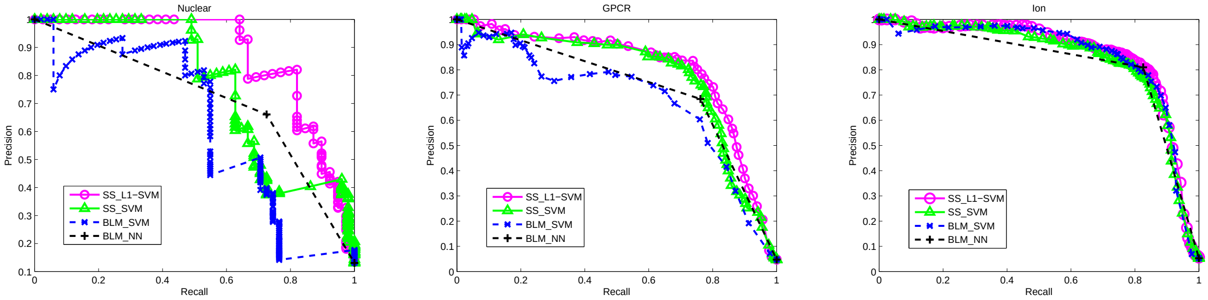

Fig. 1. The precision-recall curves of the four methods SS_L1-SVM, SS_SVM, BLM_SVM, BLM_NN, SS_NN_FS, SS_NN_noFS on three data set. The results are based on training data with drug interacting with at least 2 targets.

used three data sets—nuclear receptors, GPCRS, and ion channel—which have 54, 223, 210 drugs, 26, 95, 204 targets, and 90, 635, 1476 interactions, respectively. The three data sets used in this article are identical to those used in the state-of-the-art study,[17] which facilitates a fair comparison between the two methods. Since we only focused on target-side prediction, we did not require any drug structural or pharmacological information to obtain drug-drug similarity information. The amino acid sequences of the target proteins were obtained from the KEGG GENES database.[2]

3.4. *Classification results*

We used five measurements to evaluate the quality of drug-target prediction: Area under the Precision-Recall Curve (AUPR), Area under the ROC Curve (AUC), F-Measure, Precision (or Specification), and Recall (or Sensitivity). With the prediction score table y_{it} available from Section 3.2, these performance measures were all computed in a micro-average fashion. That is, given a cutoff point, all y_{it} could be converted into a binary label via thresholding (i.e., binding or not). By comparing these labels with the ground truth over the whole drug-by-target score table, we derived the number of false positive and false negative, which led to Precision, Recall, and F-Measure. The AUPR and AUC were derived by increasing the cutoffs with a fine step size, which led to thousands of points in the precision-recall curve. Of the five measurements, AUPR, AUC, and F-Measure are more robust than the others.

We only demonstrate the results based on *withSS* feature because the *withoutSS* feature set did not result in as good performance. Tables 1, 2, and 3 demonstrate the effectiveness of the different drug-target prediction methods over the five evaluation quantities. The F-Measure, Precision, and Recall scores reported in these tables were obtained at the cutoff point where F-Measure was maximized for respective methods. Figure 1 demonstrates the precision-recall curves of SS_L1-SVM and SS_SVM compared to BLM_SVM, BLM_NN, SS_NN_FS, and SS_NN_noFS on three data sets, namely Nuclear, GPCR, and Ion Channel from left to right.

Based on these evaluation, the SVM approaches that use *withSS* feature set (i.e., SS_L1-SVM and SS_SVM) outperform the current state-of-the-art methods BLM_SVM and BLM_NN. Moreover, the L1 norm feature selection method SS_L1-SVM is more effective than the traditional SVM method; it uses only 72.85%, 85.02%, and 62.86% of the original features

Table 1. Evaluations of classification quality on Nuclear data set. The F-Measure, Precision, and Recall scores were obtained at the cutoff point where F-Measure was maximized for respective prediction methods.

Performance comparison:	AUPR	AUC	F-Measure	Precision	Recall
SS_L1-SVM	**0.8756**	**0.9512**	**0.8205**	**0.8205**	0.8205
SS_SVM	0.7635	0.9277	0.7111	0.8205	0.6275
BLM_SVM	0.6163	0.8034	0.6353	0.7941	0.5294
BLM_NN	0.7111	0.8347	0.6916	0.6607	0.7255
SS_NN_FS	0.6985	0.8680	0.6415	0.5075	**0.8718**
SS_NN_noFS	0.6743	0.8459	0.6308	0.5190	0.8039

Table 2. Evaluations of classification quality on GPCR data set. The F-Measure, Precision, and Recall scores were obtained at the cutoff point where F-Measure was maximized for respective prediction methods.

Performance comparison:	AUPR	AUC	F-Measure	Precision	Recall
SS_L1-SVM	**0.8039**	**0.9603**	**0.7840**	**0.8360**	0.7381
SS_SVM	0.7720	0.9600	0.7607	0.8013	0.7240
BLM_SVM	0.6800	0.9435	0.6812	0.7152	0.6503
BLM_NN	0.7287	0.8721	0.7209	0.6842	0.7618
SS_NN_FS	0.7155	0.8878	0.6997	0.6219	**0.7996**
SS_NN_noFS	0.7219	0.8875	0.7081	0.6365	0.7977

Table 3. Evaluations of classification quality on Ion data set. The F-Measure, Precision, and Recall scores were obtained at the cutoff point where F-Measure was maximized for respective prediction methods.

Performance comparison:	AUPR	AUC	F-Measure	Precision	Recall
SS_L1-SVM	**0.8632**	0.9666	**0.8205**	**0.8260**	0.8151
SS_SVM	0.8450	**0.9690**	0.8045	0.8173	0.7921
BLM_SVM	0.8561	0.9568	0.8088	0.7785	**0.8416**
BLM_NN	0.8226	0.9075	0.8179	0.8101	0.8258
SS_NN_FS	0.7041	0.8542	0.6954	0.6647	0.7290
SS_NN_noFS	0.6702	0.8640	0.6497	0.5671	0.7606

in the Nuclear, GPCR, and Ion Channels datasets, respectively. The significant reduction in feature set size can not only make the classification more efficient and effective, it can also help biological practitioners to identify important features more accurately.

We further investigated the prediction result generated by the SS_L1-SVM method and the BLM_SVM method. At the prediction cutoff where both methods attained their own maximum F-Measure score, there were 8, 127, and 78 true positive interactions that SS_L1-SVM managed to identify but were missed by BLM_SVM. This was in comparison to 7, 16, 52 true positives that were identified by BLM_SVM but not by SS_L1-SVM. False positive is another important measurement of a method. On the three datasets Nuclear, GPCR, and Ion Channels, SS_L1-SVM generated 0, 73, and 139 false positive interactions, compared to 2, 85, 117 false positive interactions generated by BLM_SVM.

Some interesting case studies are in order. On the Nuclear dataset, the two nearest neighbors of the target protein RORB (KEGG Homo sapiens protein ID "hsa6096") under the normalized Smith-Waterman score are RORA ("hsa6095") and RORC ("hsa6097"), with scores 0.578 and 0.458 respectively. RORB and RORC share a common interacting drug *Tretinoin* (KEGG drug ID "D00094") while RORB and RORA do not. According to the BLM method, RORB will be predicted to have no interaction with *Tretinoin* because its nearest neighbor RORA does not interact with *Tretinoin*. On the contrary, our method can correctly identify the interaction between RORB and *Tretinoin* because the *withSS* feature set based method can discover two important substrings "EVVLVRMCRA-N" and "N-TV-FEGKYGGM" that exist in both RORB and RORC. Therefore, although the overall match score between RORB and RORC is not the highest, their feature vectors (with respect to the two feature substrings) are the most similar.

On the GPCR dataset, the five nearest neighbors of the target protein CHRM1 (KEGG Homo sapiens protein ID "hsa1128") under the normalized Smith-Waterman scores are CHRM5 ("hsa1133"), CHRM3 ("hsa1131"), CHRM4 ("hsa1132"), CHRM2 ("hsa1129"), and HRH3 ("hsa11255"), with scores 0.4707, 0.4536, 0.4237, 0.4228, and 0.2446 respectively. Although CHRM1 is supposed to bind to drug *Metoclopramide* (KEGG drug ID "D00726"), none of its five nearest neighbors bind to this drug. In fact binding occurs only with the 6-th nearest neighbor, HRH2 ("hsa3274"), whose SW_{norm} score with respect to CHRM1 is 0.2137. Therefore, the BLM methods can hardly predict that CHRM1 binds to *Metoclopramide*. In contrast, our method can correctly predict this interaction because the important substrings such as "KRTPRRAA", "Y-AKRTP-RAA-MI-L-W", and "NYFL-SLA-AD" are present in both CHRM1 and several proteins that bind to *Metoclopramide*, e.g., HTR1A ("hsa3350"), HTR1B ("hsa3351"), HTR1D ("hsa3352"), HTR1E ("hsa3354"), HTR1F ("hsa3355"), HTR2A ("hsa3356"), HTR2B ("hsa3357"), HTR2C("hsa3358"), HTR4("hsa3360"), HTR5A("hsa3361"), and HTR6("hsa3362"), which are all considered as faraway neighbors according to the SW_{norm} scores.

The binding regions discovered by our computational model can also be found on the Ion dataset. To provide potential drug-target binding regions for further investigation, we produced all the important common substrings selected by the SS_L1-SVM method, which are made available online at "`http://www.cs.ualberta.ca/~ys3/drug_target`".

4. Conclusions

In this article, we proposed a novel drug-target interaction prediction method based on potential drug-target binding regions. According to the evaluation metrics, the proposed method significantly outperformed the current state-of-the-art methods. More importantly, it identified a number of drug-target interactions that were missed by previous methods. We believe that the poor recall of previous methods is due to the use of a target kernel matrix based on Smith-Waterman score: a low overall similarity between two protein sequences does not mean they do not share common drug binding regions. This drawback was overcome in our approach by collecting a large number of candidate binding regions (i.e., common substrings) that subsequently played the primary role in interaction prediction. In addition, the use of an explicit vector representation, as opposed to implicit similarity measure, enabled the easy use of non-linear kernel expansions that were not possible for fixed kernel methods like BLM.

Besides the kernels based on substring feature vector, we believe the techniques of string kernel proposed in[25] could be useful in this problem. One straightforward benefit of using the string kernel is that it will automatically consider all substrings of a given sequence pair. It can also provide more intuitive understanding of substring-based sequence similarities than using Gaussian kernel. However, to employ the string kernel, one needs to customize the feature selection function into the model and to extend the non-gapped matching in string kernels.

We presented a feature selection method based on L_1-norm SVM that could not only predict the binding relations more accurately, but also find important candidate binding regions (features). It integrated feature selection directly into L_1-norm SVM and kernelized the optimization model. A drawback was that the sparse regularization term tended to select only a single feature from the candidate set. This is a well known problem with L_1 based regularization.[26] To avoid this limitation, we will investigate a combination of L_1 and L_2 norm regularizers, known as the elastic net,[26] which is generally more effective at group feature selection. Another possible extension is to adopt the OSCAR model,[27] which appears even more effective. We also discovered that the inference problem of drug-target interaction—in some cases—can be considered as a multi-instance learning problem. So we proposed using multiple feature selection vectors for each positive training example in theory and applied the feature cost vector to address the multi-instance problem in practice. We hypothesize that more advanced machine learning methods specifically tailored for multi-instance classification can further improve the accuracy of drug-target interaction prediction. Moreover, considering that protein 3D structures carry the essential binding information and an increasing amount of protein 3D structure is being made available (e.g., PSI Nature Structural Biology Knowledgebase[28]), incorporating protein 3D information in the prediction model in addition to sequence information would lead to promising improvement.

References

1. C. M. Dobson, Chemical space and biology, *Nature* **432**, 824 (2004).
2. M. Kanehisa, S. Goto, M. Hattori and et al., From genomics to chemical genomics: New developments in kegg, *Nucleic Acids Res.* **34**, D354 (2006).
3. B. R. Stockwell, Chemical genetics: Ligand-based discovery of gene function, *Nat. Rev. Genet.* **1**, 116 (2000).
4. D. L. Wheeler, T. Barrett, D. A. Benson and et al., Database resources of the national center for biotechnology information, *Nucleic Acids Res.* **34**, D173 (2006).
5. M. Kuhn, D. Szklarczyk, A. Franceschini, C. von Mering, L. J. Jensen and P. Bork, Stitch 3: zooming in on protein-chemical interactions, *Nucleic Acids Res.* **40**, 876 (2012).
6. G. Jones, P. Willett, R. C. Glen and et al., Development and validation for a genetic algorithm for flexible docking, *J. Mol. Biol.* **267**, 727 (1997).
7. G. M. Morris, D. S. Goodsell, R. S. Halliday and et al., Automated docking using a lamarckian

genetic algorithm and empirical binding free energy function, *J. Comput. Chem.* **19**, 1639 (1998).

8. B. K. Shoichet, D. L. Bodian and I. D. Kuntz, Molecular docking using shape descriptors, *J. Comput. Chem.* **13**, 380 (1992).
9. S. J. Haggarty, K. M. Koeller, J. C. Wong and et al., Multidimensional chemical genetic analysis of diversity-oriented synthesis-derived deacetylase inhibitors using cell-based assays, *Chem. Biol.* **10**, 383 (2003).
10. F. G. Kuruvilla, A. F. Shamji, S. M. Sternson and et al., Dissecting glucose signalling with diversity-oriented synthesis and small-molecule microarrays, *Nature* **416**, 653 (2002).
11. M. J. Keiser, B. L. Roth, B. N. Armbruster and et al., Relating protein pharmacology by ligand chemistry, *Nat. Biotech.* **25**, 197 (2007).
12. M. Campillos, M. Kuhn, A. C. Gavin and et al., Drug target identification using side-effect similarity, *Science* **321**, 263 (2008).
13. Y. Yamanishi, M. Araki, A. Gutteridge and et al., Prediction of drug-target interaction networks from the integration of chemical and genomic spaces, *Bioinformatics* **24**, i232 (2008).
14. L. Perlman, A. Gottlieb, N. Atias, E. Ruppin and R. Sharan, Combining drug and gene similarity measures for drug-target elucidation, *J. Comput. Biol.* **18(2)**, 133 (2011).
15. N. Nagamine and Y. Sakakibara, Statistical prediction of protein-chemical interactions based on chemical structure and mass spectrometry data, *Bioinformatics* **23**, 2004 (2007).
16. L. Jacob and J. P. Vert, Protein-ligand interaction prediction: An improved chemogenomics approach, *Bioinformatics* **24**, 2149 (2008).
17. K. Bleakley and Y. Yamanishi, Supervised prediction of drug-target interactions using bipartite local models, *Bioinformatics* **25**, 2397 (2009).
18. Y. Yamanishi, M. Kotera, M. Kanehisa and et al., Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework, *Bioinformatics* **26**, i246 (2010).
19. T. F. Smith and M. Waterman, Identification of common molecular subsequences, *J. Mol. Biol* **147**, 195 (1981).
20. C. Micchelli and M. Pontil, Learning the kernel function via regularization, *J. Mach. Learn. Res.* **6**, 1099 (2006).
21. G. Lanckriet, M. Cristianini, P. Bartlett and et al., Learning the kernel matrix with semi-definite programming, *J. Mach. Learn. Res.* **5**, 27 (2004).
22. I. Schomburg, A. Chang, C. Ebeling and et al., Brenda, the enzyme database: Updates and major new developments, *Nucleic Acids Res.* **32**, D431 (2004).
23. S. Gunther, M. Kuhn, M. Dunkel and et al., Supertarget and matador: Resources for exploring drug-target relationships, *Nucleic Acids Res.* **36**, D919 (2008).
24. D. S. Wishart, C. Knox, A. C. Guo and et al., Drugbank: A knowledgebase for drugs, drug actions and drug targets, *Nucleic Acids Res.* **36**, D901 (2007).
25. S. V. N. Vishwanathan and A. J. Smola, Fast kernels for string and tree matching, *NIPS* **15**, 569 (2002).
26. H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society B* **67**, 301 (2005).
27. W. Zhong and J. Kwok, Efficient sparse modeling with automatic feature grouping, *ICML* **28**, 9 (2011).
28. H. M. Berman, Psi nature structural biology knowledgebase (`www.sbkb.org/kb/index.html`) (2012).