Semi-supervised Transductive Hot Spot Predictor Working on Multiple Assumptions

Jim Jing-Yan Wang[1], Islam Khaleel Almasri[1], Yuexiang Shi[2], Xin Gao[1, 3, *]
[1]*Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST),Thuwal 23955-6900, Saudi Arabia*
[2]*Information Engineering School, Xiangtan University, Xiangtan 411105, China*
[3]*Computational Bioscience Research Center, King Abdullah University of Science and Technology (KAUST),Thuwal 23955-6900, Saudi Arabia*
[*]*All correspondence should be addressed to Xin Gao, Email: xin.gao@kaust.edu.sa*

Abstract: Protein-protein interactions are critically dependent on just a few residues ("hot spots") at the interfaces. Hot spots make a dominant contribution to the binding free energy and if mutated they can disrupt the interaction. As mutagenesis studies require significant experimental efforts, there exists a need for accurate and reliable computational hot spot prediction methods. Compared to the supervised hot spot prediction algorithms, the semi-supervised prediction methods can take into consideration both the labeled and unlabeled residues in the dataset during the prediction procedure. The transductive support vector machine has been utilized for this task and demonstrated a better prediction performance. To the best of our knowledge, however, none of the transductive semi-supervised algorithms takes all the three semi-supervised assumptions, i.e., smoothness, cluster and manifold assumptions, together into account during learning. In this paper, we propose a novel semi-supervised method for hot spot residue prediction, by considering all the three semi-supervised assumptions using nonlinear models. Our algorithm, IterPropMCS, works in an iterative manner. In each iteration, the algorithm first propagates the labels of the labeled residues to the unlabeled ones, along the shortest path between them on a graph, assuming that they lie on a nonlinear manifold. Then it selects the most confident residues as the labeled ones for the next iteration, according to the cluster and smoothness criteria, which is implemented by a nonlinear density estimator. Experiments on a benchmark dataset, using protein structure-based features, demonstrate that our approach is effective in predicting hot spots and compares favorably to other available methods. The results also show that our method outperforms the state-of-the-art transductive learning methods.

Keywords: Hot Spot Prediction, Semi-supervised Learning, Multiple Semi-supervised Assumptions, Nonlinear Manifold, Nonlinear Density Estimator

1. INTRODUCTION

It is well known that a few key hot spot residues make a dominant contribution to the binding free energy of protein-protein interactions [1]. In the case of mutation,

they can disrupt the interactions. These residues are critical in understanding the principles of protein-protein interactions. Experimental methods such as alanine scanning mutagenesis [2] cannot improve the integral performance well because they are time consuming and expensive. Therefore, computational methods are needed to efficiently and effectively predict hot spots in protein interfaces [3].

There have been a large number of studies on hot spot prediction problem. The existing methods can mainly be classified into three categories, i.e., energy-based methods [4-7], molecular dynamics-based methods [8-10], and machine learning methods [3,11-17]. Although energy-based methods and molecular dynamics-based methods demonstrated their power on some tasks, machine learning methods appeared to have better accuracy and scalability. In [13], Darnell et al. presented two knowledge-based models that improved the ability to predict hot spots: K-FADE used shape specificity features calculated by the fast atomic density evaluation (FADE) program, and K-CON used biochemical contact features. In [14], Guney et al. presented a new database of computational hot spots in protein interfaces, HotSprint. HotSprint contained data from 35,776 protein interfaces from the multi-chain structures in Protein Data Bank (PDB). The conserved residues in interfaces with certain buried accessible solvent area (ASA) and complex ASA thresholds were flagged as computational hot spots. In 2009, Cho et al. presented several new features and showed that they were more effective than the conventional features [12]. By combining the proposed features with the conventional features, they developed a predictive model for hot spot prediction. In the same year, Lise et al. presented a method to identify hot spot residues by combining advantages of machine learning approaches and energy-based approaches [15]. The basic energetic terms that contributed to hot spot interactions, i.e. Van der Waals potentials, solvation energy, hydrogen bonds and Coulomb electrostatics, were considered. They treated them as the input features and applied machine learning algorithms such as support vector machine (SVM) and Gaussian processes (GP) to optimally combine and integrate them, based on a set of training examples of alanine mutations. Later, Tuncbag et al. described an intuitive method to determine computational hot spots based on

conservation, accessible solvent accessibility and statistical pairwise residue potentials of the interface residues [3]. Meanwhile, combination of these features was examined in a comprehensive way to study their effects in hot spot detection. In [18], Li et al. proposed the geometrically centered region (GCR) based on tripartite graphs, by exploring the role of immobilized water and capturing the compactness in contact and the far distance from bulk solvent, for hot spot prediction. In [16], Xia et al. introduced an efficient approach that used SVMs to predict hot spot residues in protein interfaces. In [17], Tuncbag et al. presented a web server, HotPoint, which predicted hot spots in protein interfaces using an empirical model. The empirical model incorporated a few simple rules consisting of occlusion from solvent and total knowledge-based pair potentials of residues. In [19], Li et al. proposed a "double water exclusion" hypothesis to refine the O-ring theory, and modeled a water-free hot spot using a biclique pattern defined as two maximal groups of residues from two chains in a protein. Assi et al. later described a computational approach to predict hot spot residues in protein interfaces [11]. The method, called presaging critical residues in protein interfaces (PCRPi), integrated different metrics into one single probabilistic measure by using Bayesian networks.

The key components of machine learning-based hot spot prediction methods are the residue representation (how to represent the residues as feature vectors) and the machine learning algorithm (how to detect hot spots). The majority of the aforementioned methods focused on discovering discriminative features. From machine learning point of view, the hot spot prediction problem is a binary classification problem. The traditional way is to solve the problem by supervised learning methods. That is, a classifier is trained on the labeled residues only (training set) and then applied to the unlabeled residues (test set). However, supervised hot spot prediction methods usually suffer from the lack of sufficient labeled residues since revealing hot spots by mutagenesis experiments is time consuming and expensive [1]. On the other hand, a large number of unlabeled residues are available. To leverage both the labeled and the unlabeled data, semi-supervised learning methods are proposed. Such methods can be classified into two categories, i.e., transductive

learning and inductive learning.

Transductive learning predicts the labels for the test data by considering both labeled and unlabeled data. No decision functions are explicitly derived. There are a variety of transductive learning algorithms, such as linear neighborhood propagation (LNP) [20] and consistency method [21].

Semi-supervised inductive learning (SSIL) methods, on the other hand, try to induce a decision boundary that has a low classification error rate on the entire sample space. Some SSIL algorithms have been developed recently, such as RegBoost [22] and SemiBoost [23]. A more detailed review on semi-supervised learning can be found in [24].

Until now, limited work has been done on developing semi-supervised learning methods for hot spot prediction. Lise et al. explored the idea of introducing unlabeled data in the training set [15]. Transductive SVM was developed to work in the semi-supervised learning setting and take advantages of the additional information embedded in unlabeled data. However, the existing transductive learning methods suffer from the issue of not satisfying the assumptions for semi-supervised learning. The power of semi-supervised learning comes from the following three semi-supervised assumptions (SSAs):

1. *Manifold* assumes that the high-dimensional data lie on a low-dimensional nonlinear manifold. Properties of the manifold ensure more accurate density estimation or more appropriate similarity measures [22].
2. *Cluster* assumes that if points are located in the same cluster, they are likely to share the same class label [22].
3. *Smoothness* assumes that for a pair of points close to each other and lying in a high-density region, they are more likely to share the same class label. The decision boundary, on the other hand, is likely to lie in a low data-density region [22]. Therefore, we can predict the label of a point lying in a high-density region more affirmatively than a point lying in a low-density region.

Although a number of transductive learning methods have been proposed recently,

such as label propagation [25], linear neighborhood propagation [20], local spline regression [26], transductive SVM [27] and label consistency [21], none of them takes all three semi-supervised assumptions together into consideration, to the best of our knowledge. In this paper, we propose a novel transductive semi-supervised learning algorithm, IterPropMCS, for hot spot prediction, which considers all the three semi-supervised assumptions for the first time. We assume that the residues lie on a nonlinear manifold, and the nonlinear manifold can be recovered by a nearest graph. IterPropMCS is a graph-based algorithm. Data are stored in the form of graphs, and class information is propagated from the labeled nodes to unlabeled ones. IterPropMCS contains a novel propagation strategy, shortest path propagation (SSP), based on the manifold assumption. Moreover, based on the cluster and smoothness assumptions, IterPropMCS has a scoring function to re-select the most confident nodes labeled by SSP. The cluster and smoothness assumptions are implemented by a nonlinear density estimator. The algorithm works in an iterative manner. In each iteration, labels from the “source” residues are first propagated to the unlabeled “target” residues; a new set of “source” residues is then selected using the scoring function.

2. MATERIALS AND METHODS

In this section, we propose a novel algorithm, IterPropMCS, that **Iter**atively **prop**agates labels on a **M**anifold under the supervision of **C**luster & **S**moothness criteria. Our algorithm predicts labels for the unannotated residues by calling a manifold propagation block (M-Block) and a cluster & smoothness criteria block (C&S-Block) iteratively. We will first introduce the M-Block and the C&S-Block, then describe the final algorithm, and finally introduce the dataset used to evaluate the algorithm.

Shortest Path Propagation on Manifold (M-Block)

Suppose there are n residues represented as feature vectors $\chi = \{x_i\}_{i \in N}$, $N = \{1,2,\cdots,n\}$ and l of them $\{x_i\}_{i \in L}$ are labeled as $\{\bar{y}_i\}_{i \in L}, L = \{1,2,\cdots,l\}$ and $\bar{y}_i \in \{+1,-1\}$, where $\bar{y}_i = +1$ if residue x_i is a hot spot, otherwise $\bar{y}_i = -1$. The task is to assign the labels $\{y_i\}_{i \in U}$ to the remaining unlabeled samples $\{x_i\}_{i \in U}$ with $U = \{l+1, l+2, \cdots, n\}$. Let $u = n - l$ be the number of unlabeled samples. We organize the residue set in the form of a graph $G = \{V, E, W\}$. The node set V corresponds to the n residues, $V = \chi$. E is the edge set, and $E = \{e_{ij}\}_{i, j \in N, j \in N_i}$, where N_i is the set of all the neighboring residues (k nearest neighbors) of x_i. $W = \{w_{ij}\}_{i, j \in N}$ with w_{ij} equal to the weight of edge e_{ij}. In a straightforward manner, we define a weight between two residues as a Gaussian function.

We first assume the residues' feature vectors $\{x_i\}_{i \in N}$ lay on an underlying nonlinear manifold, where only the geodesic distances or the shortest paths reflect the true low-dimensional geometry of the manifold. Based on this assumption, we try to propagate the labels from the labeled residues $\{x_i\}, i \in L$ to the unlabeled ones $\{x_j\}, j \in U$, along the shortest paths between them. We denote the shortest path between residue i and j as $SP(i, j)$, which contains the nodes along the path. Then we consider all the labeled nodes $\{x_i\}, i \in L$ in the graph. We set $\pi_j = \bigcup_{i \in L} SP(i, j)$ as the nodes between x_j and the labeled ones $\{x_i\}_{i \in L}$. Formally,

$$\pi_j = \{k \mid k \in SP(i, j), i \in L\} \tag{1}$$

Our idea is to propagate the labels $\bar{y}_i, i \in L$, along these shortest paths between

the labeled nodes and the unlabeled ones $j \in U$ as

$$y_j = \sum_{k \in \pi_j} p_{jk} y_k \tag{2}$$

where p_{jk} is the normalized similarity measure between nodes j and k, defined by a kernel function of shortest path distance,

$$p_{jk} = \begin{cases} \dfrac{K^{sp}(x_j, x_k)}{\sum_{k \in \pi_j} K^{sp}(x_j, x_k)}, if\ k \in \pi_j \\ 0, else \end{cases} \tag{3}$$

where $K^{sp}(x_j, x_k) = exp(\dfrac{-d^{sp}(x_j, x_k)^2}{\sigma_{jk}^2})$ is the shortest path dissimilarity-based kernel, and $d^{sp}(x_j, x_k)$ is the shortest path distance between x_j and x_k.

Intuitively, Equation (2) means that the label y_j can be constructed by a linear combination of the nodes in π_j along the shortest paths between itself and the labeled nodes. We denote the entire label vector $y = \begin{bmatrix} y_l \\ y_u \end{bmatrix}$, among which $y_l = [y_1, \cdots, y_l]^{\mathrm{T}}$ is the sub-vector of the labeled ones, while $y_u = [y_{l+1}, \cdots, y_n]^{\mathrm{T}}$ is the label vector to be predicted. Hence, we estimate the label y_l by minimizing

$$\begin{aligned} E &= \sum_{j \in N} E_j = \sum_{j \in N} \left(\sum\nolimits_{k \in \pi_j} p_{jk} y_k - y_j \right)^2 \\ &= y^{\mathrm{T}} (I - P)^{\mathrm{T}} (I - P) y = y^{\mathrm{T}} Q y \end{aligned} \tag{4}$$

Here $P = [p_{jk}]_{j,k \in N}$ and $Q = (I - P)^{\mathrm{T}} (I - P)$. Q is the precision matrix and is called the biharmonic matrix. Apparently, we should try to find a solution y_u that restricts

$y_l = \bar{y}_l$ to make $E = 0$, since $E \geq 0$. Instead of solving this minimization problem directly, labeling is achieved by solving a biharmonic equation,

$$\begin{aligned} & Qy = 0 \\ & s.t.\ y_l = \bar{y}_l \end{aligned} \tag{5}$$

The constraints represent Dirichlet boundary conditions. By decomposing the matrix according to the partition of labeled and unlabeled interface residues, we have

$$\begin{aligned} & Q_{ul}\bar{y}_l + Q_{uu}\tilde{y}_u = 0 \\ & \tilde{y}_u = -Q_{uu}^{-1}Q_{ul}\bar{y}_l \\ & y_u = sign(\tilde{y}_u) \end{aligned} \tag{6}$$

where $Q_{uu} = [q_{ij}]_{i,j \in U}$ is a submatrix of Q and Q_{ul} is similarly defined. In this way, we design the Manifold propagation block (M-block), which predicts the labels $y_j, j \in U$ by propagating from the labeled ones $y_i, i \in L$. In fact, the labeled nodes can be called the “source” nodes since the label information flows to the unlabeled nodes along the shortest paths. Note that the shortest path propagation is related but completely different from the low-density separation used in [28]. The former is used to propagate the labels only whereas the latter was used to learn classifiers.

Cluster and Smoothness Criteria (C&S-Block)

Based on the other two assumptions, i.e., cluster and smoothness, we build a criterion to assign a confidence, F_j, to each unlabeled residue x_j. The confidence score F_j is then used to select the most confident residues to form the labeled residue set for the next iteration. In this section, we will first propose two scoring functions, F_j^{clu} and F_j^{smo}, to reflect cluster and smoothness assumptions,

respectively, and then combine them into F_j.

• **Cluster Criterion** We assume that the entire residue set N is clustered to K clusters $C = \{1,2,\cdots,K\}$, and for a residue x_j, we set an indicator $\phi_j \in C$ to signify which cluster it belongs to. Given a cluster c, the confidence for a residue x_j belonging to cluster c can be calculated as:

$$F_j^{clu}(c) = \sum_{i:\phi_i=c} y_j y_i = y_j \sum_{i:\phi_i=c} y_i \tag{7}$$

As is claimed by the cluster assumption, if x_j belongs to cluster c, it should share the same label with other samples in c, $\{i:\phi_i = c\}$, resulting that $y_j = y_i$; thus, $y_j y_i = 1$. Otherwise, $y_j y_i = -1$ since $y_i, y_j \in \{+1,-1\}$. $F_j^{clu}(c)$ is in fact a counter of residues in c sharing the same label with x_j. Consequently, we can use it as an indicator of whether x_j belongs to cluster c. In the clustering procedure, we also use it to assign x_j to a new cluster ϕ_j^{new}

$$\phi_j^{new} = \underset{c}{maxarg}\, F_j^{clu}(c) \tag{8}$$

Moreover, we can use $F_j^{clu}(\phi_j)$ as a confidence of whether x_j conforms to the cluster assumption, so we define it as

$$F_j^{clu} = y_j \sum_{i:\phi_i=\phi_j} y_i = \sum_{i:\phi_i=\phi_j} f_j^{clu}(i) \tag{9}$$

where $f_j^{clu}(i) = y_j y_i$ is the i-th component of the function.

• **Smoothness Criterion** According to the smoothness assumption, labels of interface residues in high density regions have higher confidence than the ones in low

density regions. This feature inspires a direct confidence measure F_j^{smo} by computing the density of x_j.

To model the local density of the interface residues contained in the training set, we use a nonlinear density estimator --- the Gaussian kernel-based kernel density estimator (KDE) based on the residues $x_i(\phi_i = \phi_j)$ in the same cluster as x_j,

$$\begin{aligned} F_j^{smo} &= p(x_j) \\ &= \frac{1}{Zn_{\phi_j}} \sum_{i:\phi_i=\phi_j} exp(-\beta \| x_i - x_j \|^2) \\ &= \sum_{i:\phi_i=\phi_j} f_j^{smo}(i) \end{aligned} \tag{10}$$

where Z is a normalization factor restricting $p(x)$ to be a proper density, β is a parameter of KDE determined experimentally, n_{ϕ_j} is the number of residues in cluster ϕ_j, and $f_j^{smo}(i) = \frac{1}{Zn_{\phi_j}} exp(-\beta \| x_i - x_j \|^2)$ is the i-th component of the function.

The cluster criterion, F_j^{clu}, is used to measure whether labeling x_j as y_j conforms to the cluster assumption, whereas the smoothness criterion, F_j^{smo}, prefers residues in high density regions that are far from the boundary. We propose a novel confidence score function, F_j, that combines both F_j^{clu} and F_j^{smo}, which is defined as

$$\begin{aligned} F_j &= \sum_{i:\phi_i=\phi_j} f_j^{clu}(i) \times f_j^{smo}(i) \\ &= \frac{1}{Zn_{\phi_j}} y_j \sum_{i:\phi_i=\phi_j} y_i \times exp(-\beta \| x_i - x_j \|^2) \end{aligned} \tag{11}$$

As we can see from (9) and (10), Both F_j^{clu} and F_j^{smo} are calculated as linear combination of many components. While the final F_j is defined as the linear

combination of the products of these components, as in (11). We could say that different assumptions have different significance over different components. The cluster assumption of i-th component $f_j^{clu}(i)$ is weighted by its smoothness assumption $f_j^{smo}(i)$, while the smoothness assumption of i-th component $f_j^{smo}(i)$ is weighted by its cluster assumption $f_j^{clu}(i)$. This score F_j will be large only when x_j satisfies both the cluster and smoothness assumptions.

After we predict label y_j for each sample x_j, they are sorted according to F_j. The unlabeled residues that have high confidence scores are considered as labeled residues for the next iteration, together with the originally labeled ones. That is

$$L' = L \bigcup \{ j \mid x_j \in \mathsf{X}, F_j \geq \theta \} \tag{12}$$

where θ is a threshold. Then L is the set of the originally labeled residues and L' is the set of labeled residues for the next iteration. L' will then be used to predict the other unlabeled samples $y_k, k \in U'$, using the M-block.

Moreover, we also update the cluster ϕ_j using $F_j(c) = \frac{1}{Zn_c} y_j \sum_{i:\phi_i = c} y_i exp(-\beta \| x_i - x_j \|)$ as:

$$\phi_j^{new} = \underset{c}{maxarg}\, F_j(c) \tag{13}$$

An unlabeled residue x_j is assigned to a new cluster ϕ_j^{new} that has the highest confidence.

IterPropMCS Algorithm

In our final algorithm, the M-Block learns the label $y_u^{(t)}$ from the “source” interface residues (which are the labeled ones) $y_l^{(t)}$, based on the manifold

assumption. Then we optimize the performance of this block using C&S-Block to re-select the best "source" interface residues $y_l^{(t+1)}$. The selection is based on the confidence score F_j, which is obtained by considering the cluster and smoothness assumptions at the same time. The selected residues will be combined with the original labeled residues to form a new "source" set which is then fed back to the M-Block. The algorithm works in an iterative manner, which runs the manifold propagation, and cluster and smoothness criteria alternately until convergence. The description of IterPropMCS algorithm is given in Algorithm 1.

Algorithm 1 Iterative Hot Spot Prediction Algorithm: IterPropMCS.

Require: n interface residues $x_i, i \in N$, l labels $\bar{y}_i \in L$;

Require: Cluster number K;

Construct the neighboring graph G and compute the shortest path information $SP(i, j)$ and $d^{sp}(x_i, x_j)$ for each pair of residues (i, j);

Initialize label source set $L^{(0)} = L$ and the target set $U^{(0)} = U$;

Initialize the cluster information $\phi_i^{(0)}, i \in N$ by a consensus clustering algorithm, such as k-means;

for $t = 0,1,\cdots,T$ **do**

M block: Decide the nodes set $\pi_j^{(t)} = \{i \mid i \in SP(k, j), k \in L^{(t)}\}$ for each x_j, and compute the manifold similarity matrix items $P^{(t)} = [p_{jk}^{(t)}]$ as in (3);

Predict the labels of the target samples $y_j, j \in U^{(t)}$ from the source samples $y_i, i \in L^{(t)}$, by solving (6) based on $P^{(t)}$;

If $y_j^{(t)} = y_j^{(t-1)}$ for each $x_j \in U$ **then**

break;

C&S block: Estimate the C&S confidence score $F_j^{(t)}$ for each x_j as in (11) based on $y_j^{(t)}, j \in N$;

Update the source set $L^{(t+1)}$ by (12) based on $F_j^{(t)}$;

Update cluster information $\phi_j^{(t+1)}$ as in (13) based on $y_j^{(t)}, j \in N$;

Update the source label $y_i^{(t+1)} = y_i^{(t)}$ for $i \in L^{(t+1)}$.

end for

Output the learned labels $y_j, j \in U$.

Dataset and Setup

To conduct the experiments, we use the residue dataset for the hot spot prediction task as χ, which is proposed in [3]. In this dataset, there are totally 262 residues, 112 of which are labeled as “hot spot” residues, and the rest 150 of which are labeled as “non-hot spot” residues. To represent each of the residue, we have used three different

feature sets, including the one proposed by Darnell et al. [13, 29], the one used by Tuncbag et al. [3, 17] and the one employed by Lise et al. [15]. The three feature sets are merged together to form a novel feature vector x_i for the i-th residue, which is named as **com**bined **s**tructure and **e**nergy-based features (ComSE).

To evaluate the performance of the proposed semi-supervised predictor, we conduct the 10-fold cross validation to the entire dataset. The dataset is split into 10 independent folds randomly, and each fold is used as test set in turns, while other 9 folds will be used as training set. The training set will be treated as labeled residue set in our IterPropMCS algorithm, while the test set will be the unlabeled residue. Moreover, we use $sensitivity$, $specificity$, $accuracy$ and F_1 score as the performance measures of the prediction results. These measures are defined as follows:

$$\begin{aligned} sensitivity &= \frac{TP}{TP+FN} \\ specificity &= \frac{TN}{FP+TN} \\ accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ F_1 &= \frac{2TP}{2TP+FP+FN} \end{aligned} \tag{14}$$

where TP is the number of true positives defined as the hot spots predicted correctly, FN is the number of false negatives defined as the hot spots predicted wrongly as non-hot spots, TN is the number of true negatives defined as the non-hot spots predicted correctly, and FP is the number of false positives defined as the non-hot spots predicted wrongly as hot spots.

3. RESULTS AND DISCUSSION

Comparison with Machine Learning Methods

To evaluate the performance of IterPropMCS, we compared our method against some popular machine learning methods, including Bayes network (BN), naive Bayes (NB), RBF network (RBF), decision tree (DT), and SVM. To measure the performance of the different classification strategies, we applied a 10-fold cross-validation on the dataset. Figure 1 shows the performance of different methods on the training and test sets. It is clear that IterPropMCS outperforms all the other classification techniques in terms of sensitivity, specificity, accuracy and F_1 score. To verify if the differences of performances between IterPropMCS and other compared machine learning methods are statistically significant, we also perform paired T-tests of the hypothesis that two the performance measures of IterPropMCS and a compared methods come from distributions with equal means. The P values of the T-tests are reported in Table 1. It could be observed that most of the performances improvement archived by IterPropMCS over other algorithms are statistically significant at the significance level of $\alpha = 0.05$.

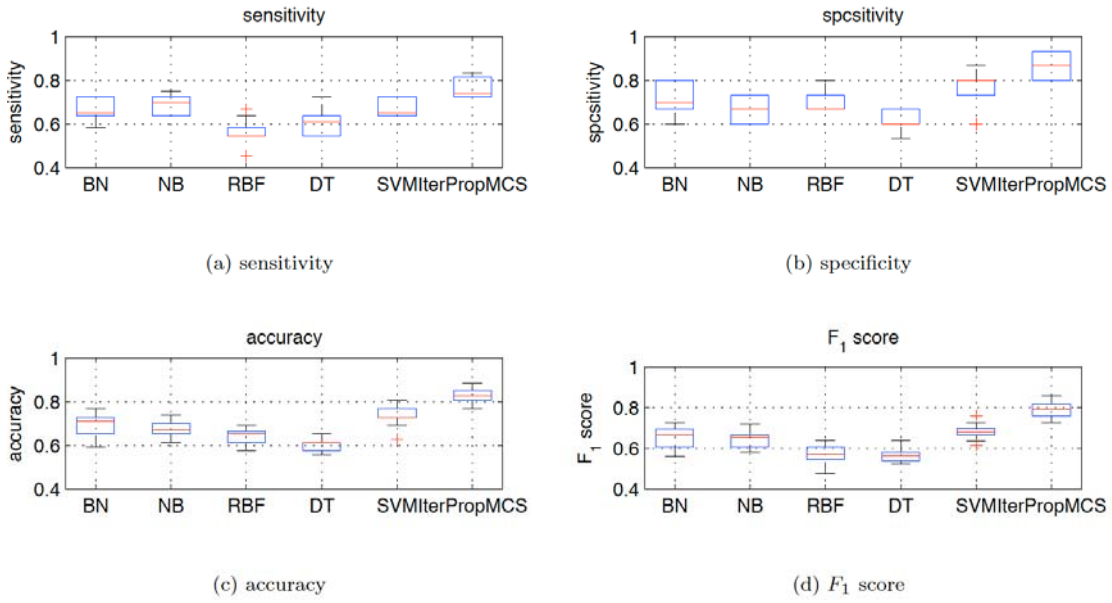

Figure 1: Boxplots of 10-fold cross validation performance of popular machine learning methods on hot spot prediction

Table 1: P values of paired T-test of the prediction performances of IterPropMCS against other machine learning methods.

Compared Methods	P-values of performance measures considered			
	sensitivity	specificity	F_1 score	accuracy
IterPropMCS vs. BN	0.0513	0.0002	0.0001	0.0002
IterPropMCS vs. NB	0.2295	0.0000	0.0001	0.0005
IterPropMCS vs. RBF	0.0001	0.0002	0.0000	0.0000
IterPropMCS vs. DT	0.0038	0.0000	0.0000	0.0000
IterPropMCS vs. SVM	0.0000	0.0390	0.0015	0.0001

Comparison with Transductive Learning Methods

We further compared IterPropMCS with several state-of-the-art transductive learning algorithms, including Gaussian random fields (GRF) [30], Laplacian regularized least square (LRLS) [31], label propagation (LP) [25], linear neighborhood propagation (LNP) [20] and consistency method (Con) [32]. As shown in Figure 2, IterPropMCS is consistently better than all the other transductive learning methods on all criteria. T-tests are also conducted to verify if the improvements of IterPropMCS over other algorithms are statistically significant. The P values are shown in Table 2. We could see that the performances differences between IterpropMCS and most other transductive learning algorithms are statistically significant at the significance level of $\alpha = 0.05$. This is another piece of strong evidence that by employing the three semi-supervised assumptions simultaneously, strong improvements can be achieved.

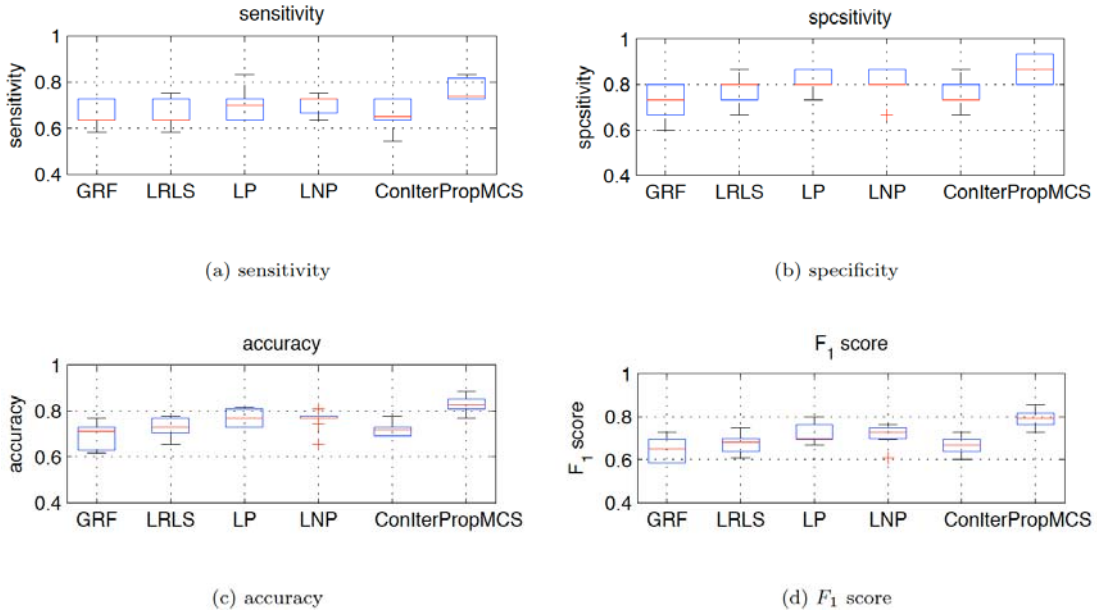

Figure 2: Boxplots of 10-fold cross validation performance of different transductive learning methods on hot spot prediction

Table 2: P values of paired T-test of the prediction performances of IterPropMCS against other transductive learning methods.

Compared Methods	P-values of performance measures considered			
	Sensitivity	Specificity	F_1 Score	ACC
IterPropMCS vs. LNP	0.0267	0.0100	0.0046	0.0051
IterPropMCS vs. LP	0.0032	0.1108	0.0152	0.0074
IterPropMCS vs. LRLS	0.0001	0.0128	0.0004	0.0001
IterPropMCS vs. Con	0.0002	0.0005	0.0000	0.0000
IterPropMCS vs. GRF	0.0044	0.0020	0.0017	0.0017

Comparison with Hot Spot Predictors

We then compared IterPropMCS with five state-of-the-art hot spot prediction methods, Robetta [5], including KFC [29], Tuncbag et al.'s method (Tuncbag) [3], Lise et al.'s method (Lisa) [15], and APIS [16]. Most of these methods are based on machine learning models that encode structural information. All learning-based methods were evaluated on the same data sets as IterPropMCS using 10-fold cross validation protocal. As shown in Figure 3, IterPropMCS consistently outperforms all the five hot spot predictors on sensitivity, F_1 score and accuracy. The sensitivity of

APIS almost catch up with IterPropMCS, but its sensitivity is much lower than IterPropMCS. The P values of T-tests of performance measures of IterPropMCS against its competitors are reported in Table 3. We could conclude that IterPropMCS outperforms the predictors statistically significantly at the significance level of $\alpha = 0.05$ for most of the prediction performances. The only exception is the differences of specificity between IterPropMCS and APIS.

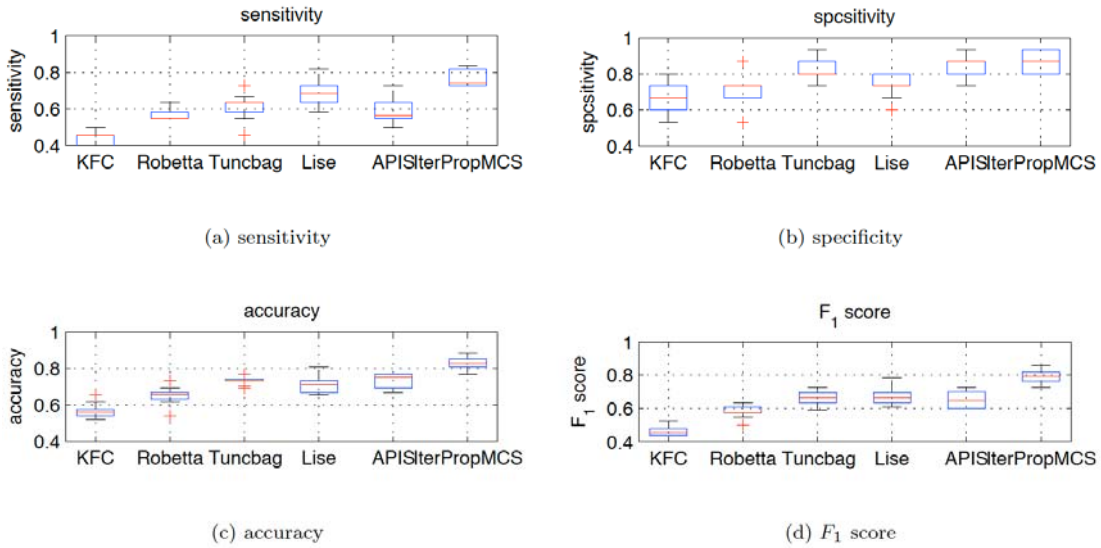

Figure 3: Boxplots of 10-fold cross validation performance of different hot spot predictors.

Table 3: P values of paired T-test of the prediction performances of IterPropMCS against other transductive learning methods.

Compared Methods	P-values of performance measures considered			
	Sensitivity	Specificity	F_1 Score	ACC
IterPropMCS vs. KFC	0.0000	0.0006	0.0000	0.0000
IterPropMCS vs. Robetta	0.0000	0.0032	0.0000	0.0000
IterPropMCS vs. Tuncbag	0.0001	0.0445	0.0000	0.0000
IterPropMCS vs. Lise	0.0037	0.0004	0.0004	0.0006
IterPropMCS vs. APIS	0.0001	0.3732	0.0006	0.0001

Comparison among Different Features

A number of existing work on hot spot prediction focuses on feature selection [3, 13,

15, 17, 29]. We tested the performance of IterPropMCS when working on different feature sets. Three different feature sets were combined, including the physical and chemical features proposed by Darnell et al. [13, 29], the structural features used by Tuncbag et al. [3] and the basic energetic terms employed by Lise et al. [15]. To study the importance of different feature sets, we conducted a leave-one-out process on the three feature sets. IterPropMCS was trained for three times, excluding one feature set at a time. The performances of predictors with different feature sets deleted are shown in Figure 4. In Figure 4, NO DEL denotes no feature deleted, while Darnell DEL denotes Darnell et al's feature set deleted, and so on. We found that exclusion of each feature led to certain level of decrease in performance, as shown in Figure 4. The P values of T-tests performed to performances of predictors using different features deleted are given in Table 4. As we can see from Table 4, among different features, excluding Darnell et al.'s features did not decrease the performance statistically significantly at the significance level of $\alpha = 0.05$, whereas excluding the Tuncbag et al. or Lise et al.'s features statistically significantly deteriorated the prediction performances.

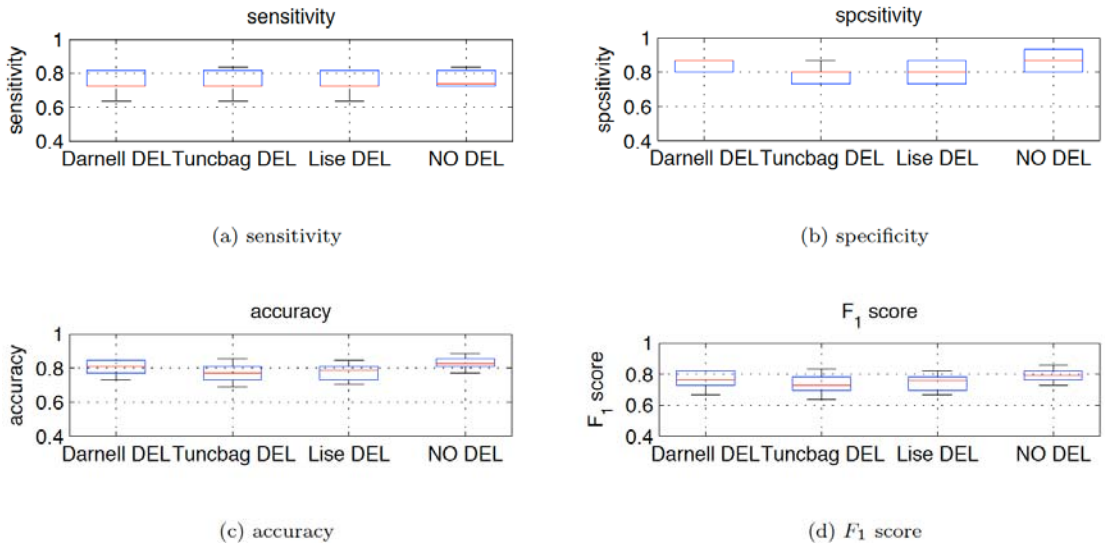

Figure 4: Boxplots of 10-fold cross validation performances with different feature sets deleted

Table 4: P values of paired T-test of the prediction performances with different feature sets deleted.

Compared Features	P-values of performance measures considered			
	Sensitivity	Specificity	F_1 Score	ACC
NO DEL vs. Darnell DEL	0.2963	0.2695	0.1952	0.1895
NO DEL vs. Tuncbag DEL	0.3249	0.0174	0.0167	0.0251
NO DEL vs. Lise DEL	0.4146	0.0319	0.0330	0.0545

Case Study

As a case study, we investigated performance of different predictors on Nidogen-1 g2/perlecan ig3 complex, a basement membrane protein with PDB ID 1GL4, chain A. There are five experimentally verified hot spot residues in this protein [33], i.e., the ASP residue of position 427, the HIS residue of position 429, the TYR residue of position 431, the GLU residue of position 616 and the ARG residue of position 620. For the sake of simplicity, we denoted the five residues as D427, H429, Y431, E616 and R620, respectively. Moreover, the ARG residue of position 403 and the TYR residue of position 440 are experimentally determined to be non-hot spots, which are denoted as R403 and Y440. For the five hot spots, each of the KFC, APIS and the Tuncbag et al.'s method predicted three correctly. IterPropMCS, on the other hand, predicted four hot spots correctly. IterPropMCS only failed to predict D427, which has also been predicted as a non-hot spot by all the other predictors. For the two non-hot spots, R403 was predicted correctly by all the predictors. However, Y440 was only predicted correctly by IterPropMCS and the Tuncbag et al.'s method. In Figure 5, we have shown where the prediction get improved, i.e., which residues in the complex are correctly predicted by IterPropMCS, but wrongly predicted by other methods. In Figure 5, the correctly predicted hot spot are marked with green color, while the

non-hot spot red color.

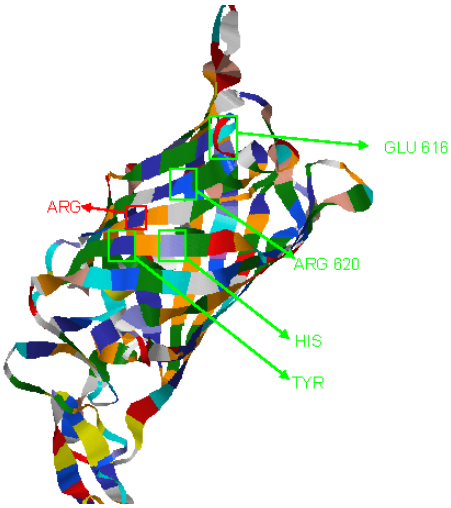

Figure 5: Illumination of hot spot and non-hot spot residues correctly predicted by IterPropMCS

4. CONCLUSIONS

We have proposed a novel semi-supervised learning algorithm, IterPropMCS, to predict hot spot residues based on structural features. To the best of our knowledge, this is the first transductive learning algorithm that takes all three semi-supervised assumptions into consideration, i.e., manifold, cluster and smoothness. Experimental results demonstrate that IterPropMCS outperforms the state-of-the-art methods on the hot spot prediction.

CONFLICT OF INTEREST

The authors declare no conflict of interest in the publication of this manuscript.

ACKNOWLEDGEMENTS

This work was partially supported by grants from the "Twelve-Five" National Plan of Scientific and Technological Support Project (sub-project) (Grant No. 2012BAK06B04), the Open Fund of Hunan Provincial Education Department Innovation Platform (Grant No. 11K069), the construct program of the key discipline

in human province and King Abdullah University of Science and Technology (KAUST).

REFERENCES

[1] Walter P, Ulucan O, Metzger J, Helms V (2012) Bioinformatics of Protein-Protein Interfaces and Small Molecule Effectors. Current Bioinformatics 7: 159-172.

[2] Sen J, Yan T, Wang J, Rong L, Tao L, et al. (2010) Alanine scanning mutagenesis of HIV-1 gp41 heptad repeat 1: Insight into the gp120-gp41 interaction. Biochemistry 49: 5057-5065.

[3] Tuncbag N, Gursoy A, Keskin O (2009) Identification of computational hot spots in protein interfaces: combining solvent accessibility and inter-residue potentials improves the accuracy. Bioinformatics 25: 1513-1520.

[4] Guerois R, Nielsen J, Serrano L (2002) Predict-ing changes in the stability of proteins and pro-tein complexes: A study of more than 1000 mutations. Journal of Molecular Biology 320: 369-387.

[5] Kortemme T, Baker D (2002) A simple physical model for binding energy hot spots in protein-protein complexes. Proceedings of the National Academy of Sciences 99: 14116-14121.

[6] Kortemme T, Kim D, Baker D (2004) Computational alanine scanning of protein-protein in-terfaces. Science's STKE 2004: pl2.

[7] Guharoy M, Chakrabarti P (2009) Empirical estimation of the energetic contribution of indi-vidual interface residues in structures of protein-protein complexes. Journal of Computer-aided Molecular Design 23: 645-654.

[8] Gonzalez-Ruiz D, Gohlke H (2006) Targeting protein-protein interactions with small molecules: Challenges and perspectives for computational binding epitope detection and ligand finding. Current Medical Chemistry 13: 2607-2625.

[9] Huo S, Massova I, Kollman P (2002) Computational alanine scanning of the 1 : 1 human growth hormone-receptor complex. Journal of Computational Chemistry 23: 15-27.

[10] Rajamani D, Thiel S, Vajda S, Camacho C (2004) Anchor residues in protein-protein inter-actions. Proceedings of the National Academy of Sciences 101: 11287-11292.

[11] Assi S, Tanaka T, Rabbitts T, Fernandez-Fuentes N (2010) PCRPi: Presaging critical residues in protein interfaces, a new computational tool to chart hot spots in protein inter-faces. Nucleic Acids Research 38: e86.

[12] Cho K, Kim D, Lee D (2009) A feature-based approach to modeling protein-protein interaction hot spots. Nucleic Acids Research 37: 2672-2687.

[13] Darnell S, Page D, Mitchell J (2007) An auto-mated decision-tree approach to predicting pro-tein interaction hot spots. Proteins: Sturcture Function and Bioinformatics 68: 813-823.

[14] Guney E, Tuncbag N, Keskin O, Gursoy A (2008) HotSprint: database of computational hot spots in protein interfaces. Nucleic Acids Research 36: D662-D666.

[15] Lise S, Archambeau C, Pontil M, Jones D (2009) Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. BMC Bioinformatics 10: 365.

[16] Xia J, Zhao X, Song J, Huang D (2010) APIS: accurate prediction of hot spots in protein inter-faces by combining protrusion index with solvent accessibility. BMC Bioinformatics 11: 174.

[17] Tuncbag N, Keskin O, Gursoy A (2010) Hot-Point: hot spot prediction server for protein 13 interfaces. Nucleic Acids Research 38: W402-W406.

[18] Li Z, Li J (2010) Geometrically centered region: A “wet” model of protein binding hot spots not excluding water molecules. Proteins: Structure, Function, and Bioinformatics 78: 3304–3316.

[19] Li J, Liu Q (2009) ‘Double water exclusion’: a hypothesis refining the O-ring theory for the hot spots at protein interfaces. Bioinformatics 25: 743-750.

[20] J Wang, F Wang, C Zhang, Shen H, L Quan (2009) Linear neighborhood propagation and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 31: 1600-15.

[21] Chapelle O, Weston J, Sch¨olkopf B (2002) Cluster kernels for semi-supervised learning. Ad-vances in Neural Information Processing Systems 15: 585-592.

[22] Chen K, Wang S (2011) Semi-supervised learning via regularized boosting working on multi-ple semi-supervised assumptions. IEEE Trans-actions on Pattern Analysis and Machine Intel-ligence 33: 129-143.

[23] Mallapragada P, Jin R, Jain A, Liu Y (2009) SemiBoost: boosting for semi-supervised learn-ing. IEEE Transactions on Pattern Analysis and Machine Intelligence 31: 2000-2014.

[24] Zhu X (2005) Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

[25] Badrinarayanan V, Galasso F, Cipolla R (2010) Label propagation in video sequences. In: Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR2010). pp. 3265-3272.

[26] Xiang S, Nie F, Zhang C (2010) Semi-supervised classification via local spline regression. IEEE Transactions on Pattern Analysis and Machine Intelligence 32: 2039-2053.

[27] X Zhang, S Zhong An improved path-based transductive support vector machines algorithm for blind steganalysis classification. In: Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI2009).

[28] Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. In: AI & Statistics 2005.

[29] Darnell S, LeGault L, Mitchell J (2008) KFC Server: interactive forecasting of protein inter-action hot spots. Nucleic Acids Research 36: W265-W269.

[30] Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using gaussian fields and Harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning (ICML2010).

[31] Belkin M, Niyogi P, Sindhwani V (2006) Mani-fold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7: 2399-2434.

[32] Zhou D, Bousquet O, Lal T, Weston J, Zhou BS (2003) Learning with local and global consis-tency. In: Proceedings of the 17th Annual Conference on Neural Information Processing Sys-tems (NIPS2003),. pp. 321-328.

[33] Kvansakul M, Hopf M, Ries A, Timpl R, Hohenester E (2001) Structural basis for the high-affinity interaction of nidogen-1 with immunoglobulin-like domain 3 of perlecan. EMBO Journal 20: 5342-5346 global consistency. pages 321-328, 2003.