Constructing Query-Driven Dynamic Machine Learning Model With Application to Protein-Ligand Binding Sites Prediction

Dong-Jun Yu*, *Member, IEEE*, Jun Hu, Qian-Mu Li, Zhen-Min Tang, Jing-Yu Yang, and Hong-Bin Shen*

***Abstract*—We are facing an era with annotated biological data rapidly and continuously generated. How to effectively incorporate new annotated data into the learning step is crucial for enhancing the performance of a bioinformatics prediction model. Although machine-learning-based methods have been extensively used for dealing with various biological problems, existing approaches usually train static prediction models based on fixed training datasets. The static approaches are found having several disadvantages such as low scalability and impractical when training dataset is huge. In view of this, we propose a dynamic learning framework for constructing query-driven prediction models. The key difference between the proposed framework and the existing approaches is that the training set for the machine learning algorithm of the proposed framework is dynamically generated according to the query input, as opposed to training a general model regardless of queries in traditional static methods. Accordingly, a query-driven predictor based on the smaller set of data specifically selected from the entire annotated base dataset will be applied on the query. The new way for constructing the dynamic model enables us capable of updating the annotated base dataset flexibly and using the most relevant core subset as the training set makes the constructed model having better generalization ability on the query, showing "part could be better than all" phenomenon. According to the new framework, we have implemented a dynamic protein-ligand binding sites predictor called OSML (On-site model for ligand binding sites prediction). Computer experiments on 10 different ligand types of three hierarchically organized levels show that OSML outperforms most existing predictors. The results indicate that the current dynamic framework is a promising future direction for bridging the gap between the rapidly accumulated annotated biological data and the effective machine-learning-based predictors. OSML web server and datasets are freely available at: http://www.csbio.sjtu.edu.cn/bioinf/OSML/ for academic use.**

***Index Terms*—Dynamic learning framework, machine learning, OSML, query-driven prediction model.**

Manuscript received December 02, 2013; revised December 18, 2014; accepted January 17, 2015. Date of current version February 27, 2015. This work was supported by the National Natural Science Foundation of China (No. 61373062, 91130033, 61175024, 61222306, and 61233011), the Natural Science Foundation of Jiangsu (No. BK20141403), Jiangsu Postdoctoral Science Foundation (No. 1201027C), China Postdoctoral Science Foundation (No. 2013M530260, 2014T70526), "The Six Top Talents" of Jiangsu Province (No. 2013-XXRJ-022), the Fundamental Research Funds for the Central Universities (No. 30920130111010). *Asterisk indicates corresponding authors.*

*D.-J. Yu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing, China, 210094, and also with the Changshu Institute, Nanjing University of Science and Technology, Changshu 215513, China (e-mail: njyudj@njust.edu.cn).

J. Hu, Q.-M. Li, Z.-M. Tang, and J.-Y. Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing, China, 210094.

*H.-B. Shen is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: hbshen@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNB.2015.2394328

I. Introduction

With the development of advanced sequencing technology and concerted genome projects, new annotated biological data are rapidly accumulated. How to effectively learn the known knowledge to unveil the structure and function for those unannotated data is a difficult but useful problem. Many machine-learning-based methods have been extensively used for dealing with various biological problems such as protein structure and function prediction [1]–[4], protein-ligand interactions prediction [5], [6], and protein-protein interactions prediction [7]–[9], protein classification [10]–[12], and so on.

Although much progress has been made in applying machine-learning-based methods to bioinformatics problems, challenges remain. One of the existing problems is that most of the existing methods tend to train a *static* prediction model on fixed dataset (termed as *static method*) thus will potentially incur several disadvantages deserved to be seriously considered:

A. Low Scalability

The static method trains prediction model on fixed dataset thus cannot well accommodate the knowledge buried in the new data. When new annotated data are available, the trained static model has to be retrained on the original dataset plus the new annotated data. As new data emerge continuously, this retraining process will be repeated continuously and thus is inefficient and with high computation complexity. On the other hand, if the existing models are not updated with the new data, their performance can be significantly lower than the newly developed predictors. This disadvantage of static method is termed as "low scalability" in this article.

The capability of utilizing new data is a critically important issue in designing prediction models since "more data usually beats better algorithms": adding more independent data usually beats out designing ever-better algorithms to analyze an existing dataset [13].

B. Impractical When Dataset Is Huge

When the size of dataset is huge enough, training and optimizing a static model on entire dataset is often impractical, either for lack of memory storage or for the long training and optimizing time needed.

The above-mentioned two points motivate researchers to develop new strategies to circumvent the disadvantages of the traditional static methods. Incremental learning model is a promising route to solve the problems of the traditional static methods listed above. An incremental learning model, which is often trained on an initial small or moderate dataset, can

be *incrementally* refined by appropriately incorporating new annotated data rather than by retraining the model from the very beginning as done by traditional static methods. As a result, many incremental learning methods have been proposed and successfully applied to different kinds of biological applications [14], [15].

In this study, we attempt to deal with the problems of the traditional static methods from a quite different view by proposing a new query-driven dynamic learning framework. Under the proposed framework, the prediction models are dynamically generated according to the query data as opposed to training a common fixed static model for all query data in traditional methods. More specifically, for a given query data, we first try to identify a query-driven smaller set of data from the entire dataset; then, the data contained in the smaller set rather than the entire dataset are used for dynamically constructing a prediction model; finally, the generated query-driven prediction model is used to perform prediction for the query data.

The proposed dynamic learning framework is a potentially promising way to solve above-mentioned disadvantages of the traditional static methods.

First, the proposed framework can expediently accommodate new annotated data thus possess better scalability: when new annotated data are available, just adding them into the existing dataset and no other additional work need to be done. Whether the new data will be used for prediction depends on the query data input.

Second, it is practical to perform prediction even when the dataset is huge. Different from static methods which train prediction models on the entire dataset, the proposed framework dynamically train a compact prediction model with smaller query-driven dataset. This is especially useful when the dataset is huge where training a static prediction model with the entire dataset is inapplicable.

To demonstrate the efficacy of the proposed dynamic learning framework, we implemented the framework for an exemplar protein-ligand binding sites prediction problem. Protein-ligand interactions are ubiquitous and indispensable for biological activities and play important roles in virtually all biological processes [16]–[18]. Experimentally identifying protein-ligand binding sites is lab-intensive and time-consuming. In addition, with advanced sequencing technology and concerted genome projects, large volumes of biological sequences without being functionally annotated have been accumulated, thus developing computational methods for predicting protein-ligand binding sites solely from amino acid sequences would be especially useful. Considerable effort has been made and many sequence-based predictors for predicting protein-ligand binding sites have emerged during the past decade [5], [6], [19]. However, most existing predictors, to the best of our knowledge, are static model based thus inevitably have those disadvantages as described above. In view of this, a new predictor called OSML (**O**n-**s**ite **m**odel for **l**igand binding sites prediction), which is an implementation of the proposed dynamic learning framework, is constructed in this paper.

We have compared the OSML with other popular sequence-based protein-ligand binding sites predictors on benchmark datasets. The computer experiments show that OSML achieves high performances and outperforms most existing protein-ligand binding sites predictors while still possessing the valuable characteristics inherited from the proposed dynamic learning framework. Currently, the OSML takes the benchmark datasets used in this study as *base dataset* for generating query-driven dataset, and can perform predictions for 10 different types of ligands.

II. Proposed Dynamic Learning Framework and Implementation of OSML

A. Proposed Dynamic Learning Framework

Let q and D be the query data and the base dataset, respectively. The traditional method usually trains a fixed static prediction model, denoted as $StaticModel$, on the entire base dataset D and then performs predictions for q with the trained $StaticModel$. Several disadvantages of static method have been discussed in Section I. In this section, we will try to present a query-driven dynamic learning framework to circumvent these disadvantages.

The proposed framework is a two-stage scheme: in the first stage, query-driven prediction model is dynamically constructed and optimized; then, prediction for the query data is performed in the second stage. The procedure of the proposed dynamic learning framework is described as follows:

1) Stage I. Dynamic Model Construction and Optimization: For the query data q, a smaller query-driven dataset, denoted as $D_{q-driven}$, is identified from the entire base dataset :

$$D_{q-driven} \leftarrow IdentifyQueryDrivenDataset(q, D). \quad (1)$$

The query-driven dataset $D_{q-driven}$ is dynamically generated for further constructing prediction model. Obviously, the identified query-driven dataset depends on the base dataset D, the query data q, and the method used for identification. The method used for identifying query-driven dataset differs for different applications. In the experimental section, we will demonstrate how to identify query-driven dataset for protein-ligand binding sites prediction problem.

Next step is to extract effective feature vector for each sample contained in $D_{q-driven}$ and the obtained feature vectors constitute the set of labeled training vectors, denoted as $F_{q-driven}$:

$$F_{q-driven} \leftarrow FeatureExtraction(D_{q-driven}). \quad (2)$$

Clearly, how to extract effective feature vectors is also problem-dependent.

After obtaining training vectors, we can further initialize and optimize a prediction model by applying an appropriate machine learning algorithm on $F_{q-driven}$ as follows:

$$InitialModel \leftarrow IntializeModel(F_{q-driven}) \quad (3)$$

$$(DModel, P_{DModel}) \leftarrow OptimizeModel(InitialModel, F_{q-driven}) \quad (4)$$

where $DModel$ and P_{DModel} are the dynamically trained prediction model and the set of optimized model parameters, respectively.

2) Stage II. Perform Prediction: Let F_q be the set of extracted feature vectors of samples contained in the query data q.

$$F_q \leftarrow FeatureExtraction(q). \tag{5}$$

Then, the query-driven prediction model obtained in stage I can be used to perform prediction task as follows:

$$Result \leftarrow Predict(F_q, DModel, P_{Model}) \tag{6}$$

where $Result$ is the prediction result for query data q.

The entire dynamic learning framework is summarized in Algorithm 1.

Algorithm 1: Dynamic Learning Framework

Input: q – the query data; D – the base dataset.

Output: *Result* – prediction result for q.

Procedure:

Stage I Dynamic model construction and optimization

1 $D_{q-driven} \leftarrow IdentifyQueryDrivenDataset(q, D)$;

2 $F_{q-driven} \leftarrow FeatureExtraction(D_{q-driven})$;

3 $InitialModel \leftarrow IntializeModel(F_{q-driven})$;

4

$(DModel, P_{DModel})$
$\leftarrow OptimizeModel(InitialModel, F_{q-driven})$;

Stage II Perform prediction

5 $F_q \leftarrow FeatureExtraction(q)$;

6 $Result \leftarrow Predict(F_q, DModel, P_{Model})$;

7 Return *Result*;

B. OSML: An On-Site Model for Protein-Ligand Binding Sites Prediction

By revisiting Algorithm 1, we can find that three steps are critical for implementing the dynamic learning framework for a specific application: a) identify a smaller query-driven dataset from base dataset (1), b) extract effective feature vectors (2), and c) choose a machine-learning method for constructing and optimizing prediction model [(3) and (4)].

The above-mentioned three steps are problem-dependent. When implementing the proposed framework for a specific application, the user should carefully select appropriate methods for identifying query-driven dataset, extracting feature vectors, and training prediction model.

As an example, in this section we will demonstrate how to implement a new predictor, called OSML, for protein-ligand binding sites prediction under the proposed dynamic learning framework. Three critical steps of the framework are implemented as follows:

1) Identify a Smaller Query-Driven Dataset: Since it has been widely accepted in bioinformatics fields that similar protein sequences will have similar functions, we thus identify query-driven dataset for a query sequence by utilizing PSI-BLAST [20], which is an effective sequence alignment tool for measuring the similarity of different sequences. More specifically, we search the query sequence q against the entire base dataset D through three iterations with T_E as E-value cutoff, and the sequences appeared in the last iteration constitute the smaller query-driven dataset $D_{q-driven}$.

Clearly, the cardinality of $D_{q-driven}$, i.e., the number of sequences contained in $D_{q-driven}$, will depends on the value of T_E: a bigger value of T_E will obtain a $D_{q-driven}$ with higher cardinality; while a smaller value of T_E will lead to a $D_{q-driven}$ with lower cardinality. In this study, T_E was uniformly set to be 0.002 for all the experiments.

2) Extract Effective Feature Vectors: Previous [5], [6], [21] and our recent work [22] have demonstrated that protein evolutionary information and predicted protein secondary structure are two effective feature sources for protein-ligand binding sites prediction. In view of this, these two feature sources will also be taken in this study. Details for feature extraction will be further described in Section III.

3) Construct and Optimize Prediction Model: As to the model construction and optimization, many machine-learning methods such as support vector machine (SVM) [23], [24], random forest [25], hidden Markov model [26], Bayesian classifier, etc., can be applied to the proposed dynamic learning framework. Nevertheless, SVM is finally chosen for constructing the prediction model in our web server implementation after our preliminary local tests. LIBSVM [24], which is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/, was used. Radial basis function is chosen as the kernel function. The other two parameters, i.e., the regularization parameter γ and the kernel width parameter σ, are optimized based on five-fold cross-validation using a grid search strategy in the LIBSVM software.

The dynamically trained query-driven SVM predicts the binding propensity of each residue in the query sequence; then, a predefined threshold T is used to determine the final predicted binding residues: those residues with binding propensities above the threshold T are marked as ligand-binding. The threshold T is optimized by maximizing the value of *MCC* (*Matthews correlation coefficient*) of predictions over five-fold cross-validation on the training datasets (cross-validation datasets) listed in Table I.

Till now, binding residues have been predicted from the query sequence using the dynamically trained prediction model. In fact, it will be more useful for biologists and users if the predictor can tell which residues actually form binding sites (pockets) for protein-ligand interaction, especially in the situation where there exists more than one binding sites (pockets) in one protein sequence. Considering this, recently we have developed a spatial clustering algorithm for further clustering the predicted binding residues to the binding sites (pockets), which will also be used in the implementation the OSML server. Details for applying spatial clustering algorithm can be found in [19]. Note that the spatial clustering procedure will not affect the performance of the binding residues prediction of OSML.

TABLE I
COMPOSITION OF THE CROSS-VALIDATION DATASETS AND THE INDEPENDENT VALIDATION DATASETS FOR THE 10 TYPES OF LIGANDS

Ligand Category	Ligand Type	Cross-Validation Dataset (Training Dataset)		Independent Validation Dataset		Total No. of Sequences
		No. of Sequences	(numP, numN)*	No. of Sequences	(numP, numN)*	
Nucleotide	ATP	221	(3021, 72334)	66	(912, 21929)	287
	ADP	296	(3833, 98740)	63	(926, 25226)	359
	AMP	145	(1603, 44401)	45	(574, 14895)	190
	GTP	54	(745, 21205)	9	(117, 2682)	63
	GDP	82	(1101, 26244)	29	(437, 7846)	111
Metal Ion	Ca^{2+}	965	(4914, 287801)	251	(1753, 82763)	1216
	Mg^{2+}	1138	(3860, 350716)	294	(1773, 95746)	1432
	Mn^{2+}	335	(1496, 112312)	83	(423, 24888)	418
	Fe^{3+}	173	(818, 50453)	41	(208, 13397)	214
	Zn^{2+}	1168	(4705, 315235)	249	(1345, 67693)	1417

* Figures numP, numN in 2-tuple (numP, numN) represent the numbers of positive (binding residues) and negative (non-binding residues) samples, respectively.

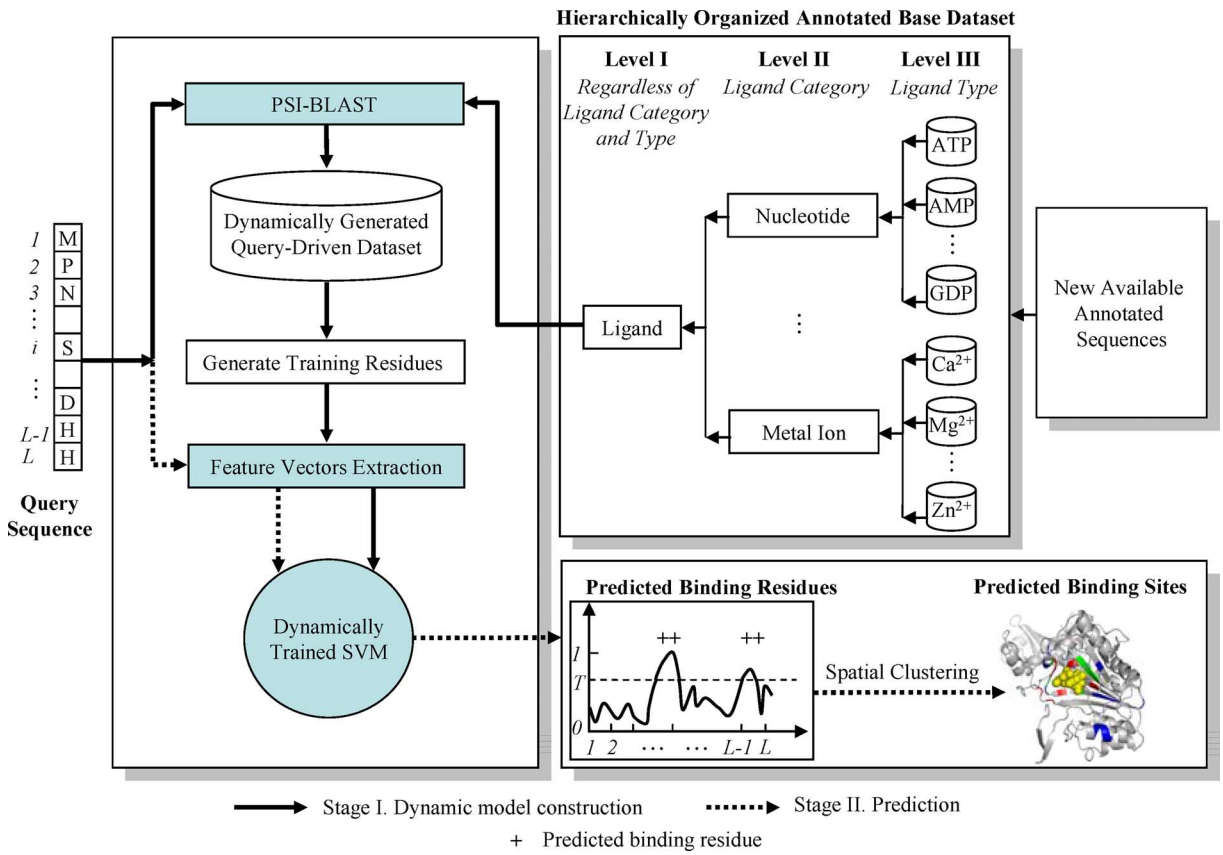


Fig. 1. Flowchart of OSML: an on-site implemented predictor for protein-ligand binding sites prediction.

Fig. 1 illustrates the flowchart of OSML.

Note that in OSML, the base dataset consisting of annotated ligand-binding protein sequences is *hierarchically* organized to facilitate the user to perform protein-ligand binding sites predictions on three different levels (refer to Fig. 1). More specifically, OSML provides three options, i.e., Level I (*Regardless of Ligand Category and Type*), Level II (*Ligand Category*), and Level III (*Ligand Type*), for user to designate on which level the query-driven dataset will be generated. For the convenience of the subsequent description, we call it a *Level I Test* if the query-driven dataset is generated on Level I of the base dataset. Similarly, *Level II Test* and *Level III Test* denote that the query-driven datasets are generated on Level II and III of the base dataset, respectively. More specifically, *Level I Test*, *Level II Test*, and *Level III Test* are defined as follows:

Level III Test: If Level III option is chosen, the user should also designate the *Ligand Type* (e.g., ATP or Ca^{2+}); then, OSML will generate query-driven dataset by performing PSI-BLAST with the subset, which consists of the corresponding annotated sequences that belongs to the designated

Ligand Type, as base dataset. If the user wants to perform ligand-specific prediction, e.g., ATP or Ca^{2+}, the *Level III Test* is suggested.

Level II Test: If Level II option is chosen, the user should also designate the *Ligand Category* (e.g., nucleotide or metal-ion); then, OSML will generate query-driven dataset by performing PSI-BLAST with the combined set, which consists of all the annotated sequences that belong to the designated *Ligand Category*, as base dataset. If the user wants to predict all possible protein-ligand binding residues for a specific ligand category, e.g., nucleotide or metal ions, *Level II Test* is an appropriate choice.

Level I Test: If Level I option is chosen, OSML will generate query-driven dataset by performing PSI-BLAST with the combined set, which consists of all the annotated sequences (i.e., the entire base dataset) regardless of the *Ligand Types* and *Ligand Categories* they belong to, as base dataset. If the user wants to perform *general-purpose* prediction, i.e., predicting all possible protein-ligand binding residues regardless of ligand type, we suggest that the *Level I Test* is taken.

It has not escaped from our notice that sometimes we may fail to identify sequences similar to the given query sequence from the base dataset, i.e., $D_{q-driven} = \phi$. In other words, there are no homologous sequences (the degree of homology can be tuned by setting different values of E-value for performing PSI-BLAST) existed in the base dataset. Two strategies can be taken to deal with this problem:

Strategy 1: In essence, OSML performs predictions based on the hypothesis that similar sequences will have similar functionalities. Since we can not find any similar sequences in the base dataset for the query sequence, all the residues in the query sequence are predicted as negative ones (non-binding residues).

Strategy 2: We train an optional static predictor to perform predictions for those query sequences that fail to identify query-driven datasets as follows: If the base dataset is small or moderate, we train a static model on the entire base dataset; if the base dataset is huge, we train a static model with part data, the amount of which depends on the computation capability of the computer used, randomly selected from the entire base dataset.

In this study, strategy 2 was taken based on the entire base dataset.

III. Materials and Methods

A. Benchmark Datasets

Well-constructed benchmark datasets including cross-validation dataset (training dataset), independent validation dataset, and blind test dataset are crucial for objectively evaluating the effectiveness of a prediction model. Many protein-ligand complexes have been deposited into the Protein Data Bank (PDB) [27]. However, not all ligands present in the PDB are biologically relevant, as small molecules are often used as additives for solving the protein structures [28], [29]. In view of this, effort has been made to filter out the biological protein-ligand complexes from the PDB and several purified ligand-protein interaction datasets have emerged, such as FireDB [29], LigASite [28], PDBbind [30], and BioLip [31], among which BioLip is the most recently released semi-manually curated database for biologically relevant protein-ligand interactions. BioLip was constructed by a four-step biological feature filtering procedure followed by careful manual verifications [31]: First, an automated four-step hierarchical procedure is used to verify the biological relevance of a ligand; After the automated procedure is completed, a careful manual check is performed to eliminate possible false positives, which can occur for entries with the commonly used crystallization additives. By doing so, it is believed with high confidence that the obtained ligand-protein complexes are real biologically relevant. Details for constructing BioLip can be found in [31].

In this study, we constructed cross-validation datasets and independent validation datasets based on the BioLip [31]. 10 different types of ligands, i.e., 5 types of metal ions and 5 types of nucleotides were considered in the present study. For each of the 10 types of the considered ligands, we constructed its cross-validation dataset and corresponding independent validation dataset as follows.

1) Cross-Validation Dataset: We extracted all the protein sequences, which interact with the given ligand and were released into PDB *before* 10 March 2010, from BioLip (downloaded on 12-Nov-2014); then the maximal pairwise sequence identity of the extracted protein sequences was culled to 40% with PISCES software [32] and the resulting sequences constitute the cross-validation dataset for that ligand;

Independent validation dataset: we extracted all the protein sequences that interact with the ligand and were deposited into PDB *after* 10 March 2010 from BioLip (downloaded on 12-Nov-2014); again, the maximal pairwise sequence identity of the extracted protein sequences was reduced to 40% and the resulting sequences constitute the validation dataset; moreover, if a given sequence in the validation dataset shares $> 40\%$ identity to a sequence in the training dataset, then we remove the sequence from the validation dataset. This assures that the sequences in validation dataset are independent of those in training dataset. Table I summarizes the detailed compositions of the cross-validation datasets and the corresponding independent validation datasets for the 10 types of ligands.

To further demonstrate the efficacy of the proposed OSML, the ninth community-wide critical assessment of techniques for protein structure prediction (CASP9) dataset was taken for blind test. CASP9 released 129 target protein sequences for blind test of protein structure and function prediction methods. Among the 129 sequences, 31 were used for evaluating the ligand binding sites predictions, where the predictors are asked to identify ligand binding residues in the sequences. As one sequence (Target ID: T0533) has been canceled on 26 May 2010, the remaining 30 sequences constitute the blind test set, which will be denoted as CASP9-Ligand in subsequent description.

B. Feature Extraction

It has been demonstrated that position specific scoring matrix (PSSM) derived features are prominent for protein attribute predictions and have been applied to many bioinformatics problems, such as protein function prediction [33], protein-ATP binding sites prediction [22], [34], protein secondary structure prediction [35], transmembrane helices prediction [36],

and subcellular localization prediction [37]–[39], and so on. Recently, PSSM-derived features have also been successfully utilized in protein-nucleotide binding [6], [19], [21] prediction. In view of this, we also take PSSM as one of the main feature sources.

A fundamental hypothesis for most of the sequence-based protein attribute predictors is that similar sequences will have similar functions. Previous studies have also shown that there exists close relationship between the protein structure and function. Many structural characteristics such as the secondary structure information [34], [40] have been intensively investigated for the identification of protein functional residues (e.g., protein-ligand binding residues). Appropriately utilizing protein structural information may help to improve the performance of protein-nucleotide binding prediction, which has also been empirically demonstrated in our recent work [22]. In view of this, protein secondary structure information predicted from protein sequence by PSIPRED [41] is used as another feature source for protein-ligand binding sites prediction.

The position specific scoring matrix (PSSM), which partially encodes the evolutionary information of a protein sequence, is obtained by using the PSI-BLAST [20] to search the Swiss-Prot database through three iterations with 0.002 as the E-value cutoff for multiple sequence alignment against the query sequence. The obtained original PSSM is further normalized with the logistic function, which has been demonstrated useful for improving protein-ATP binding prediction performance [34], as follows:

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{7}$$

where x is the score in original PSSM matrix. Then, a sliding window of size W is applied to each residue to extract its *evolution feature* as follows: for a residue at position i of the protein sequence, its *evolution feature* can be represented by a vector, which is composed of the PSSM matrix elements of the protein corresponding to a sequence segment of length W centered on i. In this study, we have tested different values of W and found that $W = 17$ is a better choice. Thus, the dimensionality of *evolution feature* vector of a residue is $17 \times 20 = 340$.

The predicted secondary structure information of a protein sequence is obtained by applying PSIPRED [41] software, which predicts the probabilities of belonging to three secondary structure classes [coil (C), helix (H), and strand (E)] for each residue in a protein sequence. More specifically, for a protein sequence with L residues, the PSIPRED outputs an $L \times 3$ probability matrix, which represents the predicted secondary structure information of the protein. Again, a sliding window of size 17 was used to extract *predicted protein secondary structure feature* of each reside and the dimensionality of the extracted feature is $17 \times 3 = 51$.

The final discriminative feature vector in this study is formed by combining the two types of features described above and the total dimensionality of the obtained feature vector is $340 + 51 = 391$.

IV. Results and Analysis

A. Evaluation Indexes

Sensitivity (*Sen*), *Specificity* (*Spe*), *Accuracy* (*Acc*), and the *Matthews correlation coefficient* (*MCC*) were used to evaluate the performance of the proposed method as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{11}$$

where TP, FP, TN, and FN denote true positive, false positive, true negative, and false negative, respectively.

However, the above four evaluation indexes are threshold dependent. In addition, protein-ligand binding sites prediction is a typical imbalanced learning problem, where the number of samples in minority class (binding residues) is significantly less than that of samples in majority class (non-binding residues). Under the imbalanced learning scenario, over pursuing the overall accuracy is not appropriate and can be deceiving for evaluating the performance of a predictor. In general, people would expect that a predictor can provide high accuracy of the minority class (e.g., binding residues in this study) without severely jeopardizing the accuracy of the majority class (non-binding residues) [42]. In view of this, together with the fact that the *MCC* provides the overall measurement of the quality of the binary predictions, we thus reported the threshold-dependent evaluation indexes (e.g., *Sen*, *Spe*, *Acc*, and *MCC*) by choosing the threshold which maximizes the value of *MCC* of predictions. More specifically, for each type of ligands, we first identify the threshold that maximizes the value of *MCC* of the predictions on the training dataset over cross-validation; then, the identified threshold, rather than another optimized threshold, was then used as the default value to evaluate the performance of the proposed method on the corresponding independent validation dataset.

B. The Proposed Dynamic Model Achieves Better Performance Than the Traditional Static Model

In this section, we will empirically demonstrate that the proposed dynamic learning framework is superior to the traditional static mode method on protein-ligand binding sites prediction problem by performing *Level I Test* (refer to Section II-B). For the purpose of fair comparison, we implemented the proposed dynamic learning framework and a corresponding static prediction model both with support vector machine (SVM) as model engine. For the convenience of subsequent description, the implemented dynamic learning framework and the corresponding

TABLE II
PERFORMANCE COMPARISONS BETWEEN OSML AND STATICSVM ON THE CROSS-VALIDATION DATASETS OF THE 10 CONSIDERED LIGANDS OVER FIVE-FOLD CROSS-VALIDATION

Ligand Category	Ligand Type	Method	Threshold *	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
Nucleotide	ATP	OSML	0.73	52.3	98.2	96.3	0.524
		StaticSVM	0.64	52.8	97.9	96.0	0.506
	ADP	OSML	0.59	52.3	98.6	96.8	0.553
		StaticSVM	0.50	54.0	98.1	96.3	0.521
	AMP	OSML	0.93	43.5	98.0	95.8	0.435
		StaticSVM	0.51	46.4	97.2	95.1	0.413
	GTP	OSML	0.97	40.9	99.4	97.3	0.531
		StaticSVM	0.56	45.7	99.1	97.2	0.531
	GDP	OSML	0.86	58.1	99.0	97.2	0.634
		StaticSVM	0.43	58.2	98.9	97.1	0.622
Metal Ion	Ca^{2+}	OSML	0.65	23.0	98.8	97.2	0.238
		StaticSVM	0.67	25.8	97.9	96.5	0.213
	Mg^{2+}	OSML	0.88	33.8	98.7	97.6	0.308
		StaticSVM	0.80	42.0	97.7	96.8	0.303
	Mn^{2+}	OSML	0.86	44.3	98.2	97.3	0.351
		StaticSVM	0.76	46.5	98.2	97.3	0.366
	Fe^{3+}	OSML	0.92	52.9	99.0	98.2	0.511
		StaticSVM	0.81	48.0	98.9	97.9	0.458
	Zn^{2+}	OSML	0.62	35.2	99.0	97.8	0.357
		StaticSVM	0.70	40.7	98.3	97.2	0.336

* For each type of ligands, the threshold was identified by maximizing the value of *MCC* of predictions on the merged training dataset (the ratio between positive and negative samples was balanced to be 1:1) over five-fold cross-validation.

static prediction model are denoted as OSML and StaticSVM, respectively.

We carried out cross-validation evaluation experiments for comparison as follows:

First, cross-validation datasets (training datasets, refer to Table I) of the 10 types of considered ligands were merged to obtain a combined dataset;

Secondly, K-fold cross-validation was applied to evaluate the performances of OSML and StaticSVM: the combined dataset was randomly partitioned into K equally-sized, disjoint subsets; then, one subset was used for testing and remaining $K-1$ subsets were used for training; this practice continued until all the K subsets of the dataset were traversed over. Note that in each round of cross-validation, $K-1$ subsets were combined for training StaticSVM; while for OSML, $K-1$ subsets were combined as base dataset for generating query-driven dataset. In this study, we performed five-fold cross-validation, i.e., $K=5$.

Thirdly, performances of OSML and StaticSVM were calculated for each type of ligands respectively in the evaluation stage.

The performance comparisons between OSML and StaticSVM on the cross-validation datasets of the 10 considered ligands over five-fold cross-validation are listed in Table II.

We also performed independent validation tests on the 10 considered independent validation datasets and the results were listed in Table III. When performing independent validation tests, StaticSVM was trained on the combined dataset of the cross-validation datasets of the 10 ligands; while for OSML, the combined dataset was used as base dataset for generating query-driven smaller dataset.

Note that in both the cross-validation test and the independent validation test, positive samples were those binding residues regardless of the types of ligands they bind to, while the negative samples were those non-binding residues accordingly. On the one hand, the number of positive samples is significantly less than that of negative samples; on the one hand, the combined dataset contains large number of samples, thus performing cross-validation test on the combined dataset with all the samples is very time-consuming. In view of this, we thus used the random under-sampling technique to balance the numbers of positive and negative samples, and the ratio between positive and negative samples was balanced to be 1:1. Another point should be addressed is that changing the ratio between positive and negative samples may affect the prediction performances [34]. However, the purpose of the experiments in this section was to experimentally compare the relative performances between OSML and StaticSVM under a fair scenario rather than to pursuit their optimal performances. We thus only provided performance comparison results under the ratio 1:1 between the positive and negative samples.

By observing Tables II and III, we can find that the OSML almost consistently outperforms the StaticSVM except for few occasions in terms of *MCC* is concerned, which is the overall measurement of the quality of the binary predictions. Averaged improvements of 1.7% and 2.4% were achieved on cross-validation test and independent validation test, respectively.

TABLE III
PERFORMANCE COMPARISONS BETWEEN OSML AND STATICSVM ON THE INDEPENDENT VALIDATION DATASETS OF THE 10 CONSIDERED LIGANDS

Ligand Category	Ligand Type	Method	Threshold *	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
Nucleotide	ATP	OSML	0.73	44.0	99.0	96.8	0.525
		StaticSVM	0.64	42.7	99.0	96.7	0.509
	ADP	OSML	0.59	47.9	99.1	97.2	0.548
		StaticSVM	0.50	48.5	98.7	96.9	0.519
	AMP	OSML	0.93	42.7	98.0	95.9	0.417
		StaticSVM	0.51	40.8	97.9	95.7	0.396
	GTP	OSML	0.97	59.7	99.3	97.6	0.675
		StaticSVM	0.56	43.9	99.5	97.1	0.573
	GDP	OSML	0.86	50.9	99.2	96.6	0.613
		StaticSVM	0.43	48.9	99.0	96.3	0.581
Metal Ion	Ca^{2+}	OSML	0.65	21.5	99.5	97.9	0.318
		StaticSVM	0.67	24.2	99.2	97.6	0.299
	Mg^{2+}	OSML	0.88	22.5	99.0	97.6	0.249
		StaticSVM	0.80	21.9	99.0	97.6	0.238
	Mn^{2+}	OSML	0.86	38.6	99.0	98.0	0.382
		StaticSVM	0.76	44.1	98.6	97.6	0.381
	Fe^{3+}	OSML	0.92	39.7	99.5	98.6	0.459
		StaticSVM	0.81	41.2	99.3	98.4	0.435
	Zn^{2+}	OSML	0.62	34.2	99.0	97.7	0.358
		StaticSVM	0.70	39.6	98.6	97.4	0.365

* For each type of ligands, results were obtained by using the threshold, which was identified on the merged training dataset (the ratio between positive and negative samples was balanced to be 1:1) over five-fold cross-validation, as the default value.

As described in previous sections, StaticSVM trains a fixed static model with *all* of the available sequences in the entire base dataset, while OSML trains a dynamic model only with query-driven sequences selected from the entire base dataset according to the query input. Together with the results listed in Tables II and III, we argue that it might be better to train a prediction model with *part* rather than *all* of the available sequences for a specific query sequence ("**part could be better than all**"), which will be further demonstrated in the subsequent sections. This is one of the important reasons why we developed the dynamic learning framework.

C. Dynamic Learning Model can Automatically Learn the Knowledge Buried in New Data

In this section, the capability of learning knowledge buried in new data of the proposed method will be demonstrated by performing *Level III*, *Level II*, and *Level I Tests* (refer to Section II-B).

1) Level III Test: For each type of ligands, let S and T be the cross-validation dataset (training dataset) and independent validation dataset, respectively. We performed *Level III Test* as follows:

1) The dataset S was randomly partitioned into 5 equally-sized, disjoint subsets $(S_1, S_2, \ldots, S_5)$;
2) Then, the independent validation dataset T was tested on StaticSVM trained with S_1 and on OSML with S_1 as base dataset, respectively;
3) At the t-th step $(2 \leq t \leq 5)$, T was tested on StaticSVM retrained with $S_1 \cup S_2 \cdots \cup S_t$ and on OSML with $S_1 \cup S_2 \cdots \cup S_t$ as base dataset, respectively.

TABLE IV
PERFORMANCES OF OSML AND STATICSVM FOR LEARNING THE NEW DATA ON LIGAND ATP

Ligand Type	Step	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
ATP	1	OSML	30.7	99.2	96.4	0.424
		StaticSVM	29.3	99.0	96.2	0.386
	2	OSML	35.5	99.2	96.6	0.465
		StaticSVM	34.1	99.1	96.5	0.445
	3	OSML	39.8	98.8	96.4	0.465
		StaticSVM	36.8	99.0	96.5	0.458
	4	OSML	40.6	99.1	96.7	0.503
		StaticSVM	40.3	98.9	96.5	0.482
	5	OSML	46.2	99.0	96.8	0.534
		StaticSVM	43.2	99.1	96.8	0.518

Table IV illustrated the performances of OSML and StaticSVM for learning the new data on ligand ATP, results of other 9 types of ligands can be found in Tables S1 and S2 in the supplementary material.

From Table IV, Table S1 and Table S2, several observations can be drawn as follows:

1) For each type of ligand, the performances of both the OSML and StaticSVM for predicting the independent validation dataset were consistently increasing with few occasions from step 1 to 5. Taking ligand ATP as an

TABLE V
PERFORMANCES OF OSML AND STATICSVM FOR LEARNING THE NEW DATA COMING FROM THE SAME LIGAND CATEGORY (NUCLEOTIDES)

Step	Training dataset	Method	*MCC*				
			ATP	ADP	AMP	GTP	GDP
1	$\bigcup_{t=1}^{1} S_t$	OSML	**0.534**	**0.467**	**0.410**	**0.523**	**0.413**
		StaticSVM	**0.518**	**0.420**	**0.402**	**0.500**	**0.420**
2	$\bigcup_{t=1}^{2} S_t$	OSML	0.551	0.603	0.387	0.486	0.428
		StaticSVM	0.529	0.566	0.372	0.462	0.417
3	$\bigcup_{t=1}^{3} S_t$	OSML	0.558	0.600	0.520	0.481	0.408
		StaticSVM	0.544	0.568	0.500	0.450	0.409
4	$\bigcup_{t=1}^{4} S_t$	OSML	0.552	0.596	0.522	0.683	0.529
		StaticSVM	0.541	0.566	0.499	0.640	0.513
5	$\bigcup_{t=1}^{5} S_t$	OSML	**0.551**	**0.591**	**0.528**	**0.674**	**0.612**
		StaticSVM	**0.544**	**0.567**	**0.501**	**0.622**	**0.582**

example (refer to Table IV), the values of *MCC* of OSML are 0.424, 0.465, 0.465, 0.503, and 0.534 from step 1 to 5, respectively; while the values of *MCC* of StaticSVM are 0.386, 0.445, 0.458, 0.482, and 0.518 from Step 1 to 5. In other words, both the OSML and StaticSVM can capture the additional knowledge from the new data thus enhance their generalization capabilities.

2) OSML almost always achieved better performances than StaticSVM at each of the 5 steps for all the 10 considered ligands. The argument "**part could be better than all**" was further demonstrated on the *Level III Test*.

2) Level II Test: Let S_1, S_2, S_3, S_4, and S_5 be the cross-validation dataset (training dataset, refer to Table I) of ATP, ADP, AMP, GTP, and GDP, respectively; while T_1, T_2, T_3, T_4, and T_5 be the independent validation dataset of ATP, ADP, AMP, GTP, and GDP, respectively. *Level II Test* was carried out as follows:

1) First, we tested T_1, T_2, T_3, T_4, and T_5 on StaticSVM trained with S_1 and on OSML with S_1 as base dataset, respectively;
2) At the t-th step ($2 \leq t \leq 5$), T_1, T_2, T_3, T_4, and T_5 were tested on StaticSVM retrained with $S_1 \cup S_2 \cdots \cup S_t$ and on OSML with $S_1 \cup S_2 \cdots \cup S_t$ as the base dataset, respectively.

Table V illustrates performances of OSML and StaticSVM for learning the new data coming from the same ligand category (nucleotide).

From the results listed in Table V, several observations can be drawn:

1) It was found that models trained on ATP can also perform well predictions for other four types of ligands, i.e., ADP, AMP, GDP, and GTP (refer to the results of step 1 listed in Table V). This observation consists with the results obtained by Firoz *et al.* [43].
2) By gradually incorporating the datasets of ADP, AMP, GTP, and GDP into the dataset of ATP, we can find that the *overall trends* of the prediction performances of both OSML and StaticSVM were increasing, although with slight fluctuations, denoting that the new data coming from the same ligand *category* may potentially help to improve prediction performance. For example, by comparing the results of Step 1 and 5 (highlighted with bold fonts) listed in Table V, we can find that the improvements for ATP, ADP, AMP, GTP, and GDP of OSML are 1.7%, 12.4%, 11.8%, 15.1%, and 19.9%, respectively; while that of StaticSVM are 2.6%, 14.7%, 9.9%, 12.2%, and 16.2%, respectively.
   Clearly, ATP, ADP, AMP, GTP, and GDP are different ligand *types*. However, they belong to the same ligand *category* and the interactions between proteins and ligands belonging to the same category possess similar sizes, roles and distributions. We believe this is one of the important reasons that accounts for the two observations described above.
3) From the results listed in Table V, again we found that OSML consistently outperforms StaticSVM at each of the 5 steps. The argument "**part could be better than all**" was further demonstrated on *Level II Test*.

3) Level I Test: In this section, we performed experiments on *Level I Test* as follows:

First, we trained a StaticSVM with all the training datasets of the five nucleotides, i.e., ATP, ADP, AMP, GTP, and GDP; then, we tested all the 10 independent validation datasets (refer to Table I) on the trained StaticSVM. Accordingly, we tested all the 10 independent validation datasets on the proposed OSML by taking the training datasets of the five nucleotides as base dataset.

Second, we retrained a StaticSVM will all the training datasets of the five nucleotides and the five metal irons; then, all the 10 independent validation datasets were on the trained StaticSVM. After that, we tested all the 10 independent validation datasets on the proposed OSML by taking the training datasets of the five nucleotides and the five metal irons as base dataset.

TABLE VI
PERFORMANCES OF OSML AND STATICSVM FOR LEARNING THE NEW DATA COMING FROM DIFFERENT LIGAND CATEGORIES

Training Dataset	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
Nucleotides	OSML	27.9	98.3	97.0	0.252
	StaticSVM	24.2	98.3	96.9	0.213
Nucleotides + Metal Irons	OSML	35.2	98.9	97.5	0.377
	StaticSVM	34.6	98.3	97.1	0.297

Table VI summarizes the averaged performance comparison of OSML and StaticSVM for learning the new data coming from different ligand categories. From Table VI, we can find that on the one hand, both the OSML and the StaticSVM can effectively utilize the new knowledge buried in the data of the newly added ligands (Metal irons); on the other hand, OSML again performed better than StaticSVM. These observations were consistent with the observations made in *Level III* and *Level II tests*.

D. Comparison With Existing Protein-Ligand Binding Sites Predictors

In this section, we will compare the proposed method with other popular protein-ligand binding sites predictors via independent validation test and blind test.

1) Independent Validation Test: Results on *Level III* of the proposed method were taken for comparison on independent validation test. According to the definition of *Level III Test*, the

proposed method will generate query-driven dataset with the base dataset consisting of sequences that all interact with the designated *Ligand Type* (refer to Section II-B). In other words, the proposed method is *ligand-specific* under *Level III Test*.

Previous studies have demonstrated that performances of *general-purpose* predictors are generally inferior to that of ligand-specific predictors [5], [44]. Thus, the comparison results will be in favor of the proposed method (i.e., the performance of the proposed method tend to be overestimated) if we compare the proposed method with other *general-purpose* predictors. In view of this, we thus compared the proposed method with other *ligand-specific* predictors rather than *general-purpose* predictors. More specifically, for each type of ligands considered in this study, we compared the proposed method with three other ligand-specific predictors.

Ligand-specific predictors taken for comparison differ for different types of ligands. In this study, TargetS [44], NsitePred [5], and SVMPred [6] were chosen as ligand-specific predictors for comparison for the five types of nucleotide ligands; while TargetS [44], FunFOLD [45], and CHED [46] were taken as ligand-specific predictors for comparison for the five types of metal ion ligands; Note that FunFOLD was initially designed for general-purpose binding sites prediction. In this study, we retrained the FunFOLD on the training dataset of each type of the five metal ion ligands and used it as ligand-specific predictor.

Next, we briefly describe the compared methods. In SVMPred [6], features derived from the predicted protein secondary structure, the relative solvent accessibility (RSA), the dihedral angles, and the PSSM are combined to form the discriminative feature of a residue; the SVM is used as prediction engine. NsitePred [5] was developed based on SVMPred by adding more features and an effective feature selection procedure. TargetS [44] utilizes the features derived from PSSM, predicted protein secondary structure, and ligand-specific binding propensity; further, an improved AdaBoost Classifier ensemble scheme based on random under-sampling was taken to deal with the class imbalance. FunFOLD [45], which uses an automatic approach for cluster identification and residue selection, is an improved automated method for the prediction of ligand binding residues using 3D models of proteins. CHED [46] is a predictor specifically designed for predicting metal-binding sites from apo protein structures.

For each type of ligands, results of the proposed method were obtained by testing the independent validation dataset on the dynamically generated model with the cross-validation dataset (training dataset, refer to Table I) of that ligand as base dataset; while prediction results of other ligand-specific predictors were obtained by feeding sequences or 3D structures in the independent validation dataset to their corresponding web servers. Considering SVMPred does not provide a web server, we thus locally implemented it based on our cross-validation datasets and tested it with the corresponding independent validation datasets.

Table VII illustrates the performance comparisons of the proposed method with TargetS [44], NsitePred [5], and SVMPred [6] on the independent validation datasets of the five nucleotide ligands. Comparison results for metal ion category, i.e., Ca^{2+}, Mg^{2+}, Mn^{2+}, Fe^{3+}, Zn^{2+} are listed in Table S3 in the supplementary materials.

TABLE VII
PERFORMANCE COMPARISONS OF THE PROPOSED METHOD WITH TARGETS [44], NSITEPRED [5], AND SVMPRED [6] ON THE INDEPENDENT VALIDATION DATASETS OF THE FIVE NUCLEOTIDE LIGANDS

Ligand Type	Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
ATP	OSML (Threshold=0.50)*	46.2	99.0	96.8	0.534
	TargetS	53.6	98.1	96.4	0.522
	NsitePred	55.6	97.3	95.6	0.482
	SVMPred	54.3	96.9	95.2	0.454
ADP	OSML (Threshold=0.50)*	47.6	99.2	97.3	0.555
	TargetS	54.5	98.9	97.4	0.583
	NsitePred	54.7	97.5	96.0	0.474
	SVMPred	49.3	97.3	95.6	0.422
AMP	OSML (Threshold=0.50)*	33.7	99.3	96.8	0.457
	TargetS	45.7	98.3	96.4	0.465
	NsitePred	43.0	97.8	95.8	0.407
	SVMPred	42.8	96.8	94.8	0.356
GTP	OSML (Threshold=0.50)*	63.2	99.1	97.6	0.684
	TargetS	64.6	98.4	97.0	0.625
	NsitePred	64.9	96.5	95.2	0.516
	SVMPred	60.1	93.6	92.2	0.383
GDP	OSML (Threshold=0.43)*	49.1	99.4	96.7	0.618
	TargetS	57.2	98.4	96.2	0.596
	NsitePred	56.1	97.9	95.6	0.553
	SVMPred	51.9	97.4	95.0	0.496

* For each type of ligands, results were obtained by using the threshold, which was identified on the corresponding training dataset over five-fold cross-validation, as the default value.

By observing Table VII, we can find that the proposed OSML significantly outperformed SVMPred [6] and is also superior to NsitePred [5]. Taken NsitePred as an example, which is one of the most recently developed *ligand-specific* protein-ligand predictors, improvements of 5.2%, 8.1%, 5.0%, 16.8%, and 6.5% on *MCC* were observed for ATP, ADP, AMP, GTP, and GDP, respectively, by the proposed OSML. On the other hand, OSML slightly outperformed TargetS on ATP, GTP, and GDP, but was somewhat inferior to TargetS on ADP and AMP. In summary, OSML achieved comparable performances to TargetS [44]. Similar phenomenon can be observed in Table S3.

Results on independent validation test listed in Table VII and Table S3 demonstrate that the proposed method, although it trains a query-driven model with only part of the entire base dataset, can still achieve comparable or even better performance to the state-of-the-art sequence-based protein-ligand binding sites predictors while still possessing the valuable characteristics inherited from the proposed dynamic learning framework.

2) Blind Test on CASP9-Ligand Dataset: In this section, we will compare the proposed OSML with other predictors by performing blind test on CASP9-Ligand dataset. Critical assessment of protein structure prediction (CASP) has become the internationally recognized contest in protein structure and function prediction from sequence. The function prediction category (FN) was introduced since the 6th CASP. The predictors participated in CASP FN represent the state-of-the-art methods for protein function prediction. In CASP9, 33 groups submitted predictions in the CASP9 function prediction category (FN) [47]. Among the 33 participating groups, 18 were registered as "human predictors" and 15 as "server predictors". Considering

TABLE VIII
PERFORMANCE COMPARISONS BETWEEN OSML AND THE 15 SERVER PREDICTORS ON CASP9-LIGAND DATASET

Predictor_ID	Predictor Name	No. of Predicted Sequences	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
FN339	I-TASSER_FUNCTION	30	67.3	98.5	97.3	0.643
FN315	FIRESTAR	30	64.5	98.6	97.3	0.638
FN236	GWS	30	48.5	98.7	96.8	0.528
FN425	INTFOLD-FN	30	41.4	99.1	96.9	0.506
-	OSML (Threshold = 0.920) ▲	30	54.1	97.5	95.8	0.481
FN452	SEOK-SERVER (GalaxyWEB)	30	54.1	97.4	95.7	0.476
FN303	FINDSITE-DBDT	27	79.1	92.2	91.7	0.460
FN017	3DLIGANDSITE1	26	50.3	97.5	95.7	0.457
FN207	ATOME2_CBS	30	53.5	97.1	95.4	0.454
FN415	3DLIGANDSITE2	26	49.7	97.4	95.5	0.443
FN453	HHPREDA	30	34.4	99.1	96.6	0.443
FN102	BILAB-ENABLE	30	71.3	92.9	92	0.421
FN057	3DLIGANDSITE3	27	43.3	97.6	95.5	0.406
FN072	3DLIGANDSITE4	27	45.1	97.3	95.2	0.401
FN132	MN-FOLD	30	84.2	73.4	73.8	0.247
FN193	MASON	29	83.0	67.0	67.7	0.202

* Performances of the 15 server predictors on CASP9-Ligand dataset were calculated according to the CASP9 official results available at http://www.predictioncenter.org/download_area/CASP9/predictions/FN.tar.gz.
▲Results were obtained by using the threshold, which was identified on the newly constructed training dataset over five-fold cross-validation, as the default value.

this, we compared the proposed OSML with the 15 server predictors according to the official results, which are available at http://www.predictioncenter.org/download_area/CASP9/predictions/FN.tar.gz.

The sequences in CASP9-Ligand dataset interact with many different types of ligands, the number of which is far beyond that (only 5 nucleotides and 5 metal ions, refer to Table I) covered by the current OSML. Thus, it is unfair for OSML to directly compare the 15 server predictors with an OSML trained only with the sequences interacting with nucleotides and metal ions on CASP9-Ligand dataset. In principle, any functionally annotated proteins can be used as training samples to train a predictor for performing CASP test, as long as the sequences in CASP do not appear in the training dataset of the predictor. Considering this, we thus re-constructed the training dataset of OSML for performing CASP test as follows:

We extracted all the protein sequences, which interact with different types of ligands (i.e., Nucleotides, SO4, IMD, GOL, PEG, PLP, CSA, LLP, ANP, FAD, EDO, COA, TLA, and STE in this computer experiment) and were released into PDB *before* 10 March 2010 (i.e., before CASP9 started), from BioLip and the extracted protein sequences constitute the training dataset of OSML for performing blind test on CASP9-Ligand dataset.

We performed blind test on CASP9-Ligand dataset as follows: first, we identified the threshold that maximizes the value of *MCC* of the predictions on the extracted training dataset over five-fold cross-validation; then, based on the identified threshold, performance of OSML on CASP9-Ligand dataset was calculated. Table VIII summarizes the performance comparisons between OSML and the 15 server predictors on CASP9-Ligand dataset.

From Table VIII, we can find that the OSML achieved relatively higher prediction performance ($MCC = 0.481$) for the 30 blind test proteins and acted as the sixth best performer among all the 16 listed predictors. Currently, the OSML were trained with only limited types of ligands. However, by observing Table VIII, we can find that OSML still achieved satisfactory performance. We believe that the prediction performance of OSML will be further improved by using more data as its base dataset.

3) Performance Comparison on Benchmark Dataset That has Been Used by Other Predictors: Except for the independent validation test and blind test performed above, we will try to further demonstrate the efficacy of the proposed method by comparing it with other predictors based on the same benchmark dataset that has been used by the compared predictors. Taking protein-nucleotide binding sites prediction as an example, Dataset 1 and Dataset 2 constructed by Chen *et al.* [5] were taken as the benchmark datasets. The Dataset 1 consists of 227, 321, 140, 56, and 105 sequences that bind to ATP, ADP, AMP, GTP, and GDP, respectively, and the maximal pairwise sequence identity of the sequences among each type of the five nucleotides was less than 40%. Dataset 2, which is the independent validation dataset of Dataset 1, consists of 17, 25, 18, 6, and 9 chains that bind to ATP, ADP, AMP, GTP, and GDP, respectively. The maximal pairwise sequence identity of the sequences among each type of the five nucleotides in Dataset 2 was less than 40%. In addition, any sequence in Dataset 2 shares less than 40% identity with a sequence in Dataset 1. Details for constructing Dataset 1 and 2 can be found in [5].

As performed in [5], we compared the proposed OSML with TargetS [44], NsitePred [5], SVMPred [6], and Rate4site [48] by using Datasets 1 and 2 as training dataset and independent validation dataset, respectively. We tested the independent validation Dataset 2 on OSML with Dataset 1 as base dataset for generating query-specific dataset. While the other four predic-

TABLE IX
PERFORMANCE COMPARISON OF THE PROPOSED OSML WITH OTHER PROTEIN-NUCLEOTIDE BINDING SITES PREDICTORS ON THE BENCHMARK DATASET 1 AND 2 CONSTRUCTED IN [5]

Ligand Type	Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
ATP	OSML (Threshold = 0.42)*	58.5	98.8	97.4	**0.595**
	TargetS	59.7	97.8	96.5	0.525
	NsitePred ▲	46.0	98.5	96.7	0.476
	SVMPred ▲	36.7	99.1	96.9	0.451
	Rate4site ▲	46.4	86.2	84.9	0.167
ADP	OSML (Threshold = 0.58)*	56.3	98.8	97.2	**0.585**
	TargetS	52.8	98.6	96.9	0.540
	NsitePred ▲	47.4	98.7	96.8	0.512
	SVMPred ▲	38.8	99.3	97.1	0.500
	Rate4site ▲	52.1	82.3	81.2	0.166
AMP	OSML (Threshold = 0.45)*	42.0	99.2	96.8	**0.522**
	TargetS	42.0	99.0	96.7	0.507
	NsitePred ▲	42.3	98.7	96.9	0.501
	SVMPred ▲	33.5	99.4	96.7	0.478
	Rate4site ▲	52.0	82.4	81.1	0.175
GTP	OSML (Threshold = 0.98)*	53.7	99.9	97.7	**0.709**
	TargetS	58.2	99.4	97.4	0.683
	NsitePred ▲	60.4	98.8	96.9	0.640
	SVMPred ▲	48.5	99.3	96.9	0.602
	Rate4site ▲	53.1	81.7	80.6	0.168
GDP	OSML (Threshold = 0.94)*	37.2	99.7	97.4	0.547
	TargetS	41.5	98.8	96.7	0.471
	NsitePred ▲	58.5	98.5	97.0	**0.576**
	SVMPred ▲	51.1	98.8	97.1	0.533
	Rate4site ▲	54.5	79.3	78.1	0.173

* For each type of ligands, results were obtained by using the threshold, which was identified on the corresponding training subset in Dataset 1 over five-fold cross-validation, as the default value.
▲ Data excerpted from [5].

tors, i.e., TargetS [44], NsitePred, SVMPred, and Rate4site were trained on Dataset 1 and tested with Dataset 2. The comparison results were listed in Table IX.

By observing Table IX, we found that the proposed OSML performed the best for 4 out of the 5 considered ligands. An averaged improvement of 2.5% on *MCC* was observed if compared with the second best performer, i.e., TargetS. Results listed in Table IX further demonstrated the efficacy of the proposed method.

E. Computational Efficiency Comparisons With Traditional Static Method

To observe the difference of computational efficiency between the proposed method and the traditional static method, we carried out experiments on the five nucleotides as follows: for each type of the five nucleotides, we trained StaticSVM on its training dataset and tested sequences in the corresponding independent validation dataset, while for OSML, the independent validation dataset was test on OSML with the corresponding training dataset as base dataset. Note that the Strategy 2 (refer to Section II-B) was applied to OSML. From the results, several observations can be made as follows:

First, the training time of OSML is equal to that of StaticSVM because OSML also needs to train an optional global static model to perform predictions for those query sequences that fail to identify query-driven datasets when the Strategy 2 was applied.

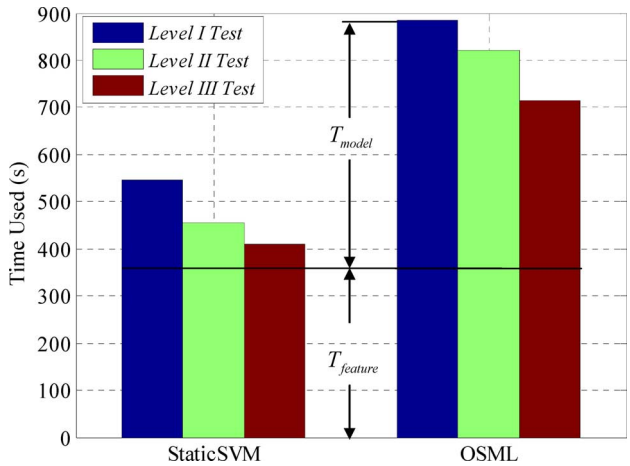


Fig. 2. Prediction efficiency comparisons between the proposed OSML and StaticSVM on *Level I*, *Level II*, and *Level III* Tests.

Second, the OSML model is much compact than StaticSVM. Our statistics show that the averaged number of support vectors of OSML is only about one-fourth of that of StaticSVM.

In essence, training a SVM on a given dataset is to find a set of support vectors under prescribed training parameters. Importantly, the number of the support vectors is far less than that of the original training feature vectors. On the other hand, the dimensionality of the support vectors is equal to that of the original training feature vectors. Generally speaking, the number of support vectors tends to increase with increment of training data size and data complexity. The StaticSVM trains a SVM with the whole dataset, while the OSML trains a SVM only with part data that has close relationship with the target sequence. In other words, the data size and the data complexity for training OSML are all smaller than that for training StaticSVM, thus it is reasonable that the number of support vectors of OSML is fewer that of StaticSVM.

Third, the prediction efficiency of the proposed OSML is inferior to that of the traditional static method. We quantitatively compared the prediction efficiency between the proposed OSML and StaticSVM as follows:

We calculated the averaged prediction times of OSML and StaticSVM under the inputs of the 10 independent validation datasets, on *Level I*, *Level II*, and *Level III Tests*, respectively. Note that the prediction time for a query sequence is the sum of the time used for extracting protein features, denoted as $T_{feature}$, and the time used by the model to perform prediction, denoted as T_{model}. Fig. 2 illustrates the prediction efficiency comparisons between the proposed OSML and StaticSVM.

From Fig. 2, we can find that the averaged $T_{feature}$ of StaticSVM is equal to that of OSML. The underlying reason is that both the StaticSVM and OSML need to extract PSSM feature and PSS feature for query sequences. On the other hand, the averaged T_{model} of StaticSVM is quite smaller than that of OSML throughout the *Level I*, *Level II*, and *Level III Tests*. The reason of this phenomenon is that OSML has to identify

query-driven dataset, dynamically train a SVM model, and perform prediction for a given query sequence, while the StaticSVM directly performs predictions for any query sequences with the pre-trained global static SVM model. In summary, we find that the averaged prediction time ($T_{feature} + T_{model}$) of OSML is about 1.5~2 times that of StaticSVM.

Nevertheless, the proposed OSML still possesses potential advantages over the StaticSVM: first, the proposed OSML can achieve better performances than the StaticSVM, which has been demonstrated in previously sections; second, when dataset is huge enough, StaticSVM can not work since training a global static model is impractical. However, OSML can still perform predictions if Strategy 1 is applied; finally, the proposed OSML has better scalability: when new annotated data are available, just adding them into the existing dataset and no other additional work need to be done.

V. Conclusions

In this study, we have proposed a new dynamic learning framework to deal with the difficulties faced by the traditional static methods. Different from the traditional static methods that train fixed static prediction models for all the queries, the proposed framework dynamically generates query-driven prediction model according to the query input thus several characteristics such as better scalability, better specificity, and better applicability, can be obtained. To demonstrate the effectiveness of the proposed dynamic learning framework, we implemented the proposed framework for protein-ligand binding sites prediction on data collected from a most recently released protein-ligand interaction database BioLip. Computer experiments on cross-validation test, blind test, and independent validation test demonstrated the efficacy of the proposed dynamic learning framework, which is flexible to suite for other biological prediction problems. The implemented predictor, called OSML, can currently perform protein-ligand binding sites prediction for 10 different ligand types on three different levels. In our future work, we will regularly update OSML by incorporating new ligand types and new annotated sequences into the base dataset, which will not affect the learning process, for further improving its prediction capability.

References

[1] B. Al-Lazikani *et al.*, "Protein structure prediction," *Curr. Opin. Chem. Biol.*, vol. 5, no. 1, pp. 51–56, Feb. 2001.

[2] E. Faraggi *et al.*, "SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles," *J. Comput. Chem.*, vol. 33, no. 3, pp. 259–67, Jan. 2012.

[3] K. Chen and L. Kurgan, "PFRES: Protein fold classification by using evolutionary information and predicted secondary structure," *Bioinformatics*, vol. 23, no. 21, pp. 2843–2850, 2007.

[4] C. Savojardo, P. Fariselli, and R. Casadio, "BETAWARE: A machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes," *Bioinformatics*, vol. 29, no. 4, pp. 504–5, Feb. 2013.

[5] K. Chen, M. J. Mizianty, and L. Kurgan, "Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors," *Bioinformatics*, vol. 28, no. 3, pp. 331–41, Feb. 2012.

[6] K. Chen, M. J. Mizianty, and L. Kurgan, "ATPsite: Sequence-based prediction of ATP-binding residues," *Proteome Sci.*, vol. 9, no. Suppl 1, p. S4, 2011.

[7] P. Fariselli *et al.*, "Prediction of protein–protein interaction sites in heterocomplexes with neural networks," *Eur. J. Biochem.*, vol. 269, no. 5, pp. 1356–61, Mar. 2002.

[8] I. Ezkurdia *et al.*, "Progress and challenges in predicting protein-protein interaction sites," *Brief Bioinform*, vol. 10, no. 3, pp. 233–46, May 2009.

[9] L. Nanni and A. Lumini, "An ensemble of K-local hyperplanes for predicting protein-protein interactions," *Bioinformatics*, vol. 22, no. 10, pp. 1207–10, May 2006.

[10] L. Nanni, A. Lumini, and S. Brahnam, "An empirical study on the matrix-based protein representations and their combination with sequence-based approaches," *Amino Acids*, vol. 44, no. 3, pp. 887–901, Mar. 2013.

[11] L. Nanni, S. Brahnam, and A. Lumini, "Wavelet images and Chou's pseudo amino acid composition for protein classification," *Amino Acids*, vol. 43, no. 2, pp. 657–65, Aug. 2012.

[12] L. Nanni and A. Lumini, "A genetic approach for building different alphabets for peptide and protein classification," *BMC Bioinformatics*, vol. 9, p. 45, 2008.

[13] A. Rajaraman, "More data usually beats better algorithms," *Datawocky*, 2008.

[14] H. B. He *et al.*, "Incremental learning from stream data," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.

[15] Z. L. Wang *et al.*, "An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 4, pp. 691–699, Jul. 2012.

[16] B. Alberts, *Molecular Biology of the Cell*, 5th ed. New York: Garland Sci., 2008.

[17] M. Gao and J. Skolnick, "The distribution of ligand-binding pockets around protein-protein interfaces suggests a general mechanism for pocket formation," *Proc. Natl. Acad. Sci. USA*, vol. 109, no. 10, pp. 3784–9, Mar. 2012.

[18] H. Kokubo, T. Tanaka, and Y. Okamoto, "Ab initio prediction of protein-ligand binding structures by replica-exchange umbrella sampling simulations," *J. Comput. Chem.*, vol. 32, no. 13, pp. 2810–21, Oct. 2011.

[19] D. J. Yu *et al.*, "TargetATPsite: A template-free method for ATP-binding sites prediction with residue evolution image sparse representation and classifier ensemble," *Jour. Comput. Chem.*, vol. 34, no. 11, pp. 974–985, Apr. 2013.

[20] A. A. Schaffer, "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements," *Nucleic Acids Res.*, vol. 29, pp. 2994–3005, 2001.

[21] J. S. Chauhan, N. K. Mishra, and G. P. Raghava, "Identification of ATP binding residues of a protein from its primary sequence," *BMC Bioinformat.*, vol. 10, p. 434, 2009.

[22] D. J. Yu *et al.*, "Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling," *Neurocomputing*, vol. 104, pp. 180–190, 2013.

[23] *Statistical Learning Theory*. New York: Wiley-Interscience, 1998.

[24] R. E. Fan, P. H. Chen, and C. J. Lin, "Working set selection using second order information for training SVM," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, 2005.

[25] L. Breiman, "RandomForests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[26] S. Mukherjee and S. Mitra, "Hidden Markov Models, grammars, and biology: A tutorial," *J. Bioinform. Comput. Biol.*, vol. 3, no. 2, pp. 491–526, Apr. 2005.

[27] P. W. Rose *et al.*, "The RCSB Protein Data Bank: Redesigned web site and web services," *Nucleic Acids Res.*, vol. 39, pp. D392–401, Jan. 2011, Database issue.

[28] B. H. Dessailly *et al.*, "LigASite–A database of biologically relevant binding sites in proteins with known apo-structures," *Nucleic Acids Res.*, vol. 36, pp. D667–73, Jan. 2008, Database issue.

[29] G. Lopez, A. Valencia, and M. Tress, "FireDB–A database of functionally important residues from proteins of known structure," *Nucleic Acids Res.*, vol. 35, pp. D219–23, Jan. 2007, Database issue.

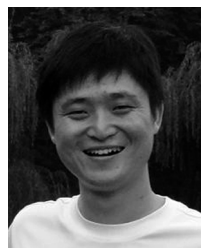
[30] R. Wang *et al.*, "The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures," *J. Med. Chem.*, vol. 47, no. 12, pp. 2977–80, Jun. 2004.

[31] J. Yang, A. Roy, and Y. Zhang, "BioLiP: A semi-manually curated database for biologically relevant ligand-protein interactions," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D1096–103, Jan. 2013.

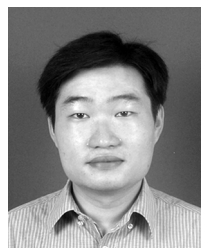
[32] G. Wang and R. L. Dunbrack Jr., "PISCES: A protein sequence culling server," *Bioinformatics*, vol. 19, no. 12, pp. 1589–91, Aug. 2003.

[33] J. C. Jeong, X. Lin, and X. W. Chen, "On position-specific scoring matrix for protein function prediction," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 8, no. 2, pp. 308–315, Mar.–Apr. 2011.
[34] Y. N. Zhang *et al.*, "Predicting protein-ATP binding sites from primary sequence through fusing bi-profile sampling of multi-view features," *BMC Bioinformatics*, vol. 13, p. 118, 2012.
[35] M. H. Zangooei and S. Jalili, "Protein secondary structure prediction using DWKF based on SVR-NSGAII," *Neurocomputing*, 2012.
[36] D. J. Yu, H. B. Shen, and J. Y. Yang, "SOMPNN: An efficient non-parametric model for predicting transmembrane helices," *Amino Acids*, vol. 42, no. 6, pp. 2195–205, Jun. 2012.
[37] A. Pierleoni, P. L. Martelli, and R. Casadio, "MemLoci: Predicting subcellular localization of membrane proteins in eukaryotes," *Bioinformatics*, vol. 27, no. 9, pp. 1224–30, May 2011.
[38] H. B. Shen and K. C. Chou, "A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0," *Anal. Biochem.*, vol. 394, no. 2, pp. 269–74, Nov. 2009.
[39] M. W. Mak, J. Guo, and S. Y. Kung, "PairProSVM: Protein subcellular localization based on local pairwise profile alignment and SVM," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 5, no. 3, pp. 416–22, Jul.–Sep. 2008.
[40] K. Chen, M. J. Mizianty, and L. Kurgan, "ATPsite: Sequence-based prediction of ATP-binding residues," *Proteome Sci.*, vol. 9, no. Suppl 1, p. S4, 2011.
[41] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, no. 2, pp. 195–202, Sep. 1999.
[42] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
[43] A. Firoz *et al.*, "Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates," *BMC Biochem.*, vol. 12, p. 20, 2011.
[44] D. Yu *et al.*, "Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 10, no. 4, pp. 994–1008, 2013.
[45] D. B. Roche, S. J. Tetchner, and L. J. McGuffin, "FunFOLD: An improved automated method for the prediction of ligand binding residues using 3D models of proteins," *BMC Bioinformat.*, vol. 12, p. 160, 2011.
[46] M. Babor *et al.*, "Prediction of transition metal-binding sites from apo protein structures," *Proteins*, vol. 70, no. 1, pp. 208–17, Jan. 2008.
[47] T. Schmidt *et al.*, "Assessment of ligand – binding residue predictions in CASP9," *Proteins: Structure, Function, Bioinformat.*, vol. 79, no. S10, pp. 126–136, 2011.
[48] T. Pupko *et al.*, "Rate4Site: An algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues," *Bioinformatics*, vol. 18, no. Suppl 1, pp. S71–7, 2002.

Dong-Jun Yu (M'12) received the B.S. degree in computer science and the M.S. degree in artificial intelligence from Jiangsu University of Science and Technology, China, in 1997 and 2000, respectively, and the Ph.D. degree in pattern analysis and machine intelligence from Nanjing University of Science and Technology, China, in 2003. In 2008, he acted as an academic visitor at University of York, U.K. He is currently a professor in the School of Computer Science and Engineering of Nanjing University of Science and Technology. His current interests include bioinformatics, pattern recognition, and data mining. He is a member of CCF.



Jun Hu received his B.S. degree in computer science from Anhui Normal University, China, in 2011. Currently, he is working toward the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include bioinformatics, data mining, and pattern recognition.



Qian-Mu Li received the B.S. degree in computer science and the Ph.D. degree in computer science from Nanjing University of Science and Technology, China, in 2002 and 2006, respectively. He is currently a professor in the School of Computer Science and Engineering of Nanjing University of Science and Technology, China. His current interests include data mining and information security. He is a member of CCF.

Zhen-Min Tang received the B.S. degree and M.S. degree in Computer Science from Nanjing University of Science and Technology (NUST), Nanjing, China. He is now a Professor in the School of Computer Science and Engineering at NUST. He is the author of over 40 scientific papers in pattern recognition, image processing, and artificial intelligence. His current interests are in the areas of pattern recognition, image processing, artificial intelligence, and expert system.

Jing-Yu Yang received the B.S. degree in Computer Science from NUST, Nanjing, China. From 1982 to 1984 he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL, USA. From 1993 to 1994 he was a visiting professor at the Department of Computer Science, Missouri University. And in 1998, he acted as a visiting professor at Concordia University, Canada. He is currently a professor and Chairman in the department of Computer Science at NUST. He is the author of over 150 scientific papers in computer vision, pattern recognition, and artificial intelligence. He has won more than 20 provincial awards and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.

Hong-Bin Shen received his Ph.D. degree from Shanghai Jiaotong University, China, in 2007. He was a postdoctoral research fellow of Harvard Medical School from 2007 to 2008. Currently, he is a professor of Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University. His research interests include data mining, pattern recognition, and bioinformatics. Dr. Shen has published more than 60 papers and constructed 20 bioinformatics severs in these areas and he serves the editorial members of several international journals.