Amino Acids (2016) 48:2533–2547
DOI 10.1007/s00726-016-2274-4

ORIGINAL ARTICLE

# TargetCrys: protein crystallization prediction by fusing multi-view features with two-layered SVM

**Jun Hu[1] · Ke Han[1] · Yang Li[1] · Jing-Yu Yang[1] · Hong-Bin Shen[2] · Dong-Jun Yu[1]**

Received: 30 July 2015 / Accepted: 7 June 2016 / Published online: 14 June 2016


**Abstract** The accurate prediction of whether a protein will crystallize plays a crucial role in improving the success rate of protein crystallization projects. A common critical problem in the development of machine-learning-based protein crystallization predictors is how to effectively utilize protein features extracted from different views. In this study, we aimed to improve the efficiency of fusing multi-view protein features by proposing a new two-layered SVM (2L-SVM) which switches the feature-level fusion problem to a decision-level fusion problem: the SVMs in the 1st layer of the 2L-SVM are trained on each of the multi-view feature sets; then, the outputs of the 1st layer SVMs, which are the "intermediate" decisions made based on the respective feature sets, are further ensembled by a 2nd layer SVM. Based on the proposed 2L-SVM, we implemented a sequence-based protein crystallization predictor called TargetCrys. Experimental results on several benchmark datasets demonstrated the efficacy of the proposed 2L-SVM for fusing multi-view features. We also compared TargetCrys with existing sequence-based protein crystallization predictors and demonstrated that the proposed TargetCrys outperformed most of the existing predictors and is competitive with the state-of-the-art predictors. The TargetCrys webserver and datasets used in this study are freely available for academic use at: http://csbio.njust.edu.cn/bioinf/TargetCrys.

**Keywords** Protein crystallization prediction · Multi-view feature fusion · Support vector machine · Machine learning

Handling Editor: S. C. E. Tosatto.

✉ Dong-Jun Yu
njyudj@njust.edu.cn; njyudj@126.com

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing 210094, China

[2] Department of Automation, Shanghai Jiao Tong University, Dongchuan Road 800, Shanghai 200240, China

## Introduction

It has become widely accepted that a close relationship exists between protein structure and function (Tramontano and Cozzetto 2004; Gromiha 2010; Zhang 2014). Knowing the accurate structure of a protein is vitally important for determining its functionalities and interactions with other biological molecules (Chou 2004; Roy and Zhang 2012; Hu et al. 2014a). Due to the rapid development of technological breakthroughs and the explosion in gene sequence data (Rung and Brazma 2013), a large volume of protein sequences without determined structures have been accumulated. The large gap between protein sequences and structures has inspired the launch of structural genomics initiatives (SGI) (Todd et al. 2005) that aim to describe the tertiary structures of every protein encoded by a given genome and narrow the gap between sequences and structures. X-ray crystallography (Mizianty et al. 2014), Nuclear Magnetic Resonance (NMR) spectroscopy (Jackman 2012), and electron microscopy (Bradshaw et al. 2012) are the three most commonly used methods for determining the structures of proteins. Representing one of the most popular experimental approaches, the X-ray crystallography method has been used to obtain approximately 80–90 % of the deposited protein structures in the Protein Data Bank (PDB) (Berman et al. 2000; Singh et al. 2011; Mizianty and Kurgan 2012). However, the major challenge of the X-ray crystallography method is that not all of the proteins used for determining structures are crystallizable,

resulting in a low success rate (approximately 2–10 %) for protein crystallization projects (Service R 2005); thus, a great deal of time and resources are wasted on these non-crystallizable proteins (Jahandideh and Mahdavi 2012).

Correctly determining whether a protein will crystallize greatly improves the success rate of crystallization projects, because obtaining diffraction-quality protein crystals is a crucial step for the X-ray crystallography method. Considerable attention has been paid for developing useful models that can be used to either support or directly predict protein crystallization (Rupp and Wang 2004). For example, Kantardjieff et al. (2004; Kantardjieff and Rupp 2004) performed pioneer work to support protein crystallization projects by calculating the isoelectric point (pI) from a primary sequence and then using the suggested optimal pH ranges for crystallization screening; Overton and Barton developed a normalized scale for SG target ranking called the OB-Score (Overton and Barton 2006) based on the pI and hydrophobicity features. Researchers also found that other protein characteristics such as the presence of transmembrane helices, low-complexity regions, intrinsic protein disorders, and coiled-coil regions could be used for predicting crystallization propensity (Rodrigues and Hubbard 2003; Price Ii et al. 2009; Zucker et al. 2010). These pioneering works have laid a solid foundation for designing more sophisticated predictors for protein crystallization from protein sequences.

Since the success of SECRET (Smialowski et al. 2006), which performs predictions based on sequence-derived features using support vector machine and a Naïve Bayes classifier, there has been a surge of interest in the development of statistical- and machine-learning-based methods to further improve the performance of protein crystallization propensity predictions. Kurgan's group has published a series of achievements on predicting protein crystallization propensities from protein sequences and sequence-derived features, and several predictors such as CRYSTALP (Chen et al. 2007), CRYSTALP2 (Kurgan et al. 2009), MetaCrys (Mizianty and Kurgan 2009), PCCpred (Mizianty and Kurgan 2011), and CRYSpred (Mizianty and Kurgan 2012) have been developed. Other popular predictors include PXS (Price Ii et al. 2009), XtalPred (Slabinski et al. 2007), ParCrys (Overton et al. 2008), MetaPPCP (Mizianty and Kurgan 2009), SVMCRYS (Kandaswamy et al. 2010), MCSG-Z score (Babnigg and Joachimiak 2010), RFCRYS (Jahandideh and Mahdavi 2012), SCMCRYS (Charoenkwan et al. 2013), and PPCinter (Gao et al. 2014). Among these listed methods, several methods such as OB-Score (Overton and Barton 2006) and XtalPred (Slabinski et al. 2007) have been applied by structural genomics centers to help improve the success rate of crystallization projects. Table 1 chronologically summarizes recent progress in sequence-based protein crystallization predictions.

By observing Table 1, several observations can be made:

First, amino acid composition (AAC) is the most frequently used feature. Among the 10 listed predictors, 8 utilize AAC as the main feature source. Apart from AAC, many physicochemical or structural properties are also often used for predicting protein crystallization. Interestingly, to the best of our knowledge, none of the existing predictors utilizes sequence evolutionary information, which has been demonstrated to be very useful for many other protein attribute prediction problems, to perform protein crystallization predictions. Consequently, questions remain concerning the reasons researchers fail to consider sequence evolutionary information and its usefulness for protein crystallization predictions.

Second, for most of the listed predictors the features extracted from multi-views[1] including AAC, dipeptide composition (DPC), tripeptide composition (TPC), and pseudo amino acid composition (PseAAC) are used to fulfill the protein crystallization predictions. However, almost all of these predictors utilize the combined feature by simply serially combining protein features extracted from different views. Previous studies have shown that serially combining different features, sometimes but not definitely, lead to the improvement of prediction accuracy compared with a single view feature (Chen et al. 2008; Yu et al. 2013a). The underlying reason is that the combination of features will simultaneously increase the information redundancy, that could, in turn, deteriorate the prediction accuracy (Kohavi and John 1997). Here, information redundancy is the information (or knowledge) which is non-unique in a feature vector (Foulonneau 2007). When there exists redundant information in feature vectors, the traditional learning algorithms (such as, SVM in this paper) often cannot obtain the best model (Dieckmann and Rieskamp 2007).

Third, although a great deal of progress has been made in predicting protein crystallization and state-of-the-art predictors can achieve high prediction accuracies, further improvement in prediction performances will accelerate the progress of X-ray crystallography-based protein structure determinations.

Motivated by the above-mentioned issues, in this study we evaluated the following: (1) investigating whether and to what extent sequence evolutionary information can be used to perform protein crystallization predictions; (2) developing a new two-layered SVM (2L-SVM) method for protein multi-view feature fusion; and (3) implementing a sequence-based protein crystallization predictor based on

[1] In this study, multi-view features mean the features extracted from different sources, such as amino acids composition, protein evolutionary profile, and so on.

**Table 1** Summarization of some popular sequence-based protein crystallization predictors

| Method | Features | Classifier | Single/ensemble | Year |
|---|---|---|---|---|
| OB-score (Overton and Barton 2006) | Physicochemical properties | Single threshold | Single | 2006 |
| XtalPred (Slabinski et al. 2007) | AAC, Physicochemical properties, and secondary Structure | Logarithm method | Single | 2007 |
| ParCrys (Overton et al. 2008) | AAC, physicochemical properties, and low complexity region | Parzen window density estimator | Single | 2008 |
| CRYSTALP2 (Kurgan et al. 2009) | AAC, DPC, TPC, physicochemical properties, and *p*-collocated amino acid pair composition | Gaussian RBFN | Single | 2009 |
| MetaPPCP (Mizianty and Kurgan 2009) | AAC, DPC, TPC, physicochemical properties, secondary structure, low complexity region, and *p*-collocated amino acid pair composition | Single threshold, logarithm method, Parzen window density estimator, Gaussian radial basis function network, and logistic model tree | Ensemble | 2009 |
| SVMCrys (Kandaswamy et al. 2010) | AAC, TPC, physicochemical properties, and Secondary Structure | SVM | Single | 2010 |
| PPCpred (Mizianty and Kurgan 2011) | AAC, physicochemical properties, secondary structure, disorder, and solvent accessibility | SVM | Ensemble | 2011 |
| RFCRYS (Jahandideh and Mahdavi 2012) | AAC, DPC, TPC, Physicochemical properties, sequence length, and PseAAC | Random forest | Ensemble | 2012 |
| SCMCRYS (Charoenkwan et al. 2013) | *p*-collocated amino acid pair composition | Scoring card method | Ensemble | 2013 |
| PPCinter (Gao et al. 2014) | AAC, physicochemical properties, secondary structure, disorder, and protein–protein interaction information | SVM | Single | 2014 |
| TargetCrys | AAC, DPC, TPC, PseAAC, and PsePSSM | 2L-SVM | Ensemble | This paper |

*AAC* amino acid composition, *DPC* dipeptide composition, *TPC* tripeptide composition, *PseAAC* pseudo amino acid composition, *PsePSSM* pseudo position specific scoring matrix

**Table 2** Statistics of the two benchmark datasets

| Training subset | | | Independent validation subset | | |
|---|---|---|---|---|---|
| Dataset | No. of positive sequences[a] | No. of negative sequences[b] | Dataset | No. of positive sequences[a] | No. of negative sequences[b] |
| TRAIN3587 | 1204 | 2383 | TEST3585 | 1204 | 2381 |
| TRAIN1500 | 756 | 744 | TEST500 | 244 | 256 |

[a] Positive sequences: crystallizable proteins

[b] Negative sequences: non-crystallizable proteins

the proposed 2L-SVM that can either outperform or complement existing methods.

## Materials and methods

### Benchmark datasets

We used two benchmark datasets that have been used in previous studies to evaluate the efficacy of the proposed method. The first benchmark dataset constructed by Mizianty and Kurgan (2011) consists of a training subset (TRAIN3587) and a corresponding independent validation subset (TEST3585). TRAIN3587 contains 1204 crystallizable and 2383 non-crystallizable proteins, while TEST3585 is composed of 1204 crystallizable and 2381 non-crystallizable proteins.

Two points have been carefully considered by Mizianty and Kurgan for constructing TRAIN3587 and TEST3585 (Mizianty and Kurgan 2011):

1. How to label a protein sequence? A protein is labeled as non-crystallizable if its completed stop status falls into any of the following statuses: 'sequencing failed', 'cloning failed', 'expression failed', 'purification failed', 'crystallization failed', and 'poor diffraction'; while a protein is labeled as crystallizable if its completed stop status is 'crystal structure'.
2. How to reduce the redundancy of dataset? They observed that even relatively minor changes in protein sequence (e.g., point mutations and usage of C- and N-terminus tag to ease purification) may change the final outcome of the crystallization. Thus, applying <25 % sequence identity constraint on constructing benchmark dataset, which is widely used in other protein attribute prediction problems, will remove the difficulty of prediction instead, leading to over-optimistic results. Considering this, Mizianty and Kurgan (2011) constructed the benchmark datasets (i.e., TRAIN3587 and TEST3585), where similar sequences (>25 % sequence identity) within each class, rather than whole dataset, were removed. Those similar sequences between classes were not removed.

For comprehensive comparison with the existing predictors, we also employed the second benchmark dataset which consists of a training subset, denoted as TRAIN1500, and one corresponding independent validation subset, denoted as TEST500. TRAIN1500 and TEST 500 were created by Kurgan et al. (2009). TRAIN1500 contains 756 crystallizable and 744 non-crystallizable protein sequences, whereas TEST500 consists of 244 crystallizable and 256 non-crystallizable protein sequences.

Table 2 summarizes the detailed statistics of the two benchmark datasets described above. Details for constructing these datasets can be found in (Overton et al. 2008; Kurgan et al. 2009; Mizianty and Kurgan 2011).

### Multi-view feature representation

From the point of view of pattern recognition, protein crystallization prediction is a typical classification problem. Thus, the question of how to represent protein sequences with discriminative features is one of the most crucial aspects for designing a classification model with high accuracy. Many attempts have been made to find better feature representations of protein sequences for protein crystallization predictions. In this section, several typical features including AAC, DPC, TPC, and PseAAC that have been demonstrated to be, especially, useful for protein crystallization prediction are briefly described. Additionally, the pseudo position specific scoring matrix feature, denoted as PsePSSM, will also be introduced because we will investigate whether protein evolutionary information can be used for protein crystallization predictions.

#### *Amino acid composition feature*

Amino acid composition (AAC) is a traditional sequence-based feature representation method that has been widely used in many protein attribute prediction problems including protein crystallization prediction (Chen et al. 2007; Overton et al. 2008; Jahandideh and Mahdavi 2012). Let $AA_1$, $AA_2$, ..., $AA_{19}$, and $AA_{20}$, be the 20 ordered native amino acid types (i.e., $A$, $C$, ..., $W$, $Y$), $n_i$ be the number of occurrence of the $AA_i$ in a protein sequence, and $L$ be the protein sequence length. Then, the AAC feature

of the protein is a 20-D vector that can be formulated as follows:

$$\mathrm{AAC} = \left(\frac{n_1}{L}, \frac{n_2}{L}, \ldots, \frac{n_{19}}{L}, \frac{n_{20}}{L}\right)^T \quad (1)$$

where $T$ represents the transpose of the vector.

*Dipeptide composition feature*

Dipeptide composition (DPC) is based on the frequency of a contiguous collocation of amino acid pairs. DPC can provide more information about a protein sequence than the traditional AAC because it reflects interactions between local (with respect to the sequence) amino acid pairs (Chen et al. 2007). Let $\mathrm{AA}_1\mathrm{AA}_1$, $\mathrm{AA}_1\mathrm{AA}_2$, …, $\mathrm{AA}_{20}\mathrm{AA}_{19}$, and $\mathrm{AA}_{20}\mathrm{AA}_{20}$ be the 400 possible amino acid pairs and $n_{i,j}$ be the number of occurrences of $\mathrm{AA}_i\mathrm{AA}_j$ ($1 \le i,j \le 20$) in a protein sequence; then, the DPC of the protein sequence is a 400-D vector that can be formulated as follows:

$$\mathrm{DPC} = \left(\frac{n_{1,1}}{L-1}, \frac{n_{1,2}}{L-1}, \ldots, \frac{n_{20,19}}{L-1}, \frac{n_{20,20}}{L-1}\right)^T \quad (2)$$

*Tripeptide composition feature*

Similar to the dipeptide composition, the tripeptide composition (TPC) is based on the frequency of a contiguous collocation of three amino acids (Kurgan et al. 2009; Ding et al. 2012). Let $\{\mathrm{AA}_i\mathrm{AA}_j\mathrm{AA}_k | 1 \le i,j,k \le 20\}$ be the set of all possible tripeptides. Clearly, the cardinality of the set is 8000. By scanning one sequence using a sliding window of three contiguous residues with each step, the 8000-D *TPC* vector can be formulated as follows (Ding et al. 2012):

$$\mathrm{TPC} = \left(\frac{n_{1,1,1}}{L-2}, \frac{n_{1,1,2}}{L-2}, \ldots, \frac{n_{20,20,19}}{L-2}, \frac{n_{20,20,20}}{L-2}\right)^T \quad (3)$$

where $n_{i,j,k}$ is the number of occurrences of $\mathrm{AA}_i\mathrm{AA}_j\mathrm{AA}_k$ ($1 \le i,j,k \le 20$) in the protein sequence.

*Pseudo amino acid composition feature*

The pseudo amino acid composition (PseAAC) (Chou 2001, 2005; Nanni et al. 2012), which can reflect both the composition information and sequence-order information of a protein sequence, is the extension of the traditional amino acid composition. In this study, the Type 2 PseAAC proposed by Chou (2005) is taken into consideration. The Type 2 PseAAC encodes a protein sequence into a $(20 + \zeta \cdot \lambda)$-D vector, among which the first 20 components are the traditional amino acid composition (AAC) and the remaining $\zeta \cdot \lambda$ components are scalar quantities reflecting the sequence-order information of the protein where $\zeta$ and $\lambda$ are the number of amino acid physiochemical characteristics and the rank of correlation, respectively. Six commonly used physiochemical characteristics (Yu et al. 2012) including hydrophobicity, hydrophilicity, side chain mass, pK of the $\alpha - \mathrm{COOH}$ group, pK of the $\alpha - \mathrm{NH_2}$ group, and the isoelectric point (pI) at 25 °C are used to construct the sequence-order information of the PseAAC (i.e., $\zeta = 6$). The PseAAC feature is generated via the server developed by Shen and Chou (2008) that is freely available at: http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/. Note that when generating PseAAC, the rank of correlation ($\lambda$) is set to be 8. Thus, the dimensionality of the generated PseAAC is $20 + 6 \times 8 = 68$.

*Pseudo position specific scoring matrix feature*

As described in the introduction, none of the existing predictors utilizes sequence evolutionary information to perform protein crystallization predictions. Considering this, we investigated whether and to what extent protein sequence evolutionary information can be used to perform protein crystallization predictions. In this study, the pseudo position specific scoring matrix feature (PsePSSM) (Chou and Shen 2007; Yu et al. 2013a, b, c) is used to extract evolutionary information from a protein sequence as follows:

For a protein sequence $\mathbf{P}$ with $L$ amino acid residues, we first obtain its position specific scoring matrix (PSSM), which is a $L \times 20$ matrix, using PSI-BLAST (Schaffer et al. 2001) to search the Swiss-Prot database against the sequence through three iterations with 0.001 as the $E$ value cut-off; then, each element (score) in the obtained PSSM is normalized to the range [0,1] with the logistic function $f(x) = 1/(1 + \mathrm{e}^{-x})$, where $x$ represents the original score in the PSSM matrix. Let $\mathbf{P}_{pssm} = \left(p_{i,j}\right)_{L\times 20}$ be the normalized PSSM of a protein sequence; then the PsePSSM can be constructed with the following two steps (Yu et al. 2012, 2013a):

Step 1: calculating the PSSM composition

The PSSM composition of a protein sequence is a 20-D feature vector as defined:

$$\mathbf{F}_{\overline{\mathrm{PSSM}}} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_{20})^T \quad (4)$$

where $\bar{p}_j = \frac{1}{L}\sum_{i=1}^{L} p_{i,j}$.

Step 2: computing correlation factors

We calculate the $g$-tier correlation factor for the $j$th ($1 \le j \le 20$) column of $\mathbf{P}_{\mathrm{pssm}} = \left(p_{i,j}\right)_{L\times 20}$, denoted as $\theta_j^g$, by coupling the $g$-most contiguous PSSM scores along the protein sequence as follows:

$$\theta_j^g = \frac{1}{L-g}\sum_{i=1}^{L-g}\left(p_{i,j} - p_{i+g,j}\right)^2 \quad (5)$$

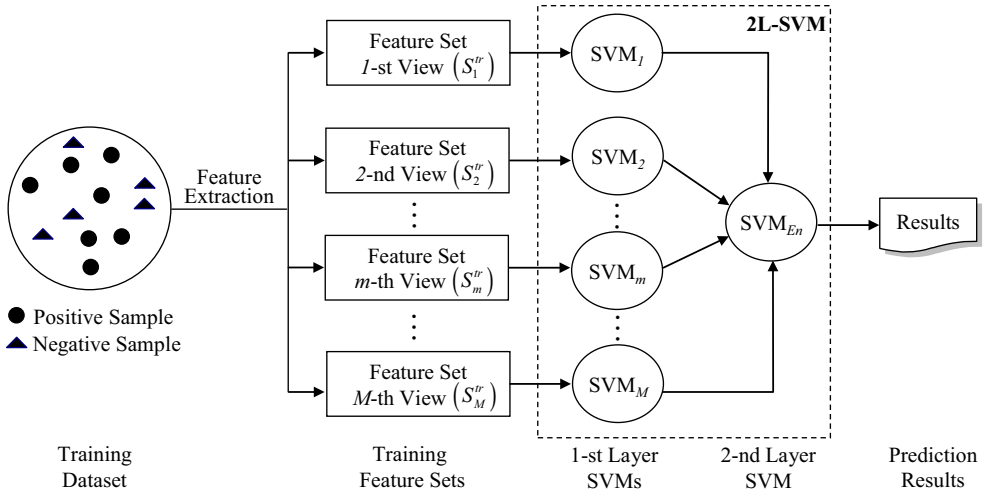


**Fig. 1** Schematic diagram of the proposed 2L-SVM for multi-view feature fusion

Let $\theta^g = \left(\theta_1^g, \theta_2^g, \cdots, \theta_{20}^g\right)^T$ be the $g$-tier correlation factor vector composed of $\theta_j^g$ $(1 \le j \le 20)$ and $G$ $(G < L)$ be the maximum of $g$ $(g = 1, 2, \ldots, G)$; then, the PsePSSM of a protein sequence, denoted as $\mathbf{x}_{\text{PsePSSM}}^G$, is the combination of its PSSM composition and the $G$ correlation factor vectors as follows:

$$\mathbf{x}_{\text{PSSM\_ACT}}^G = \begin{pmatrix} \mathbf{F}_{\overline{\text{PSSM}}} \\ \theta^1 \\ \vdots \\ \theta^g \\ \vdots \\ \theta^G \end{pmatrix} \tag{6}$$

Considering that the minimal length of the protein sequences in the benchmark datasets is 9 and the fact that there are no theoretical justifications on determining the optimal value of $G$, we empirically tested different values of $G$ from 1 to 8 on the training dataset (i.e., TRAIN3587) and found that the optimal value of $G$ is 8. Accordingly, the dimensionality of the obtained PsePSSM is $20 + 8 \times 20 = 180$.

### Multi-view feature fusion with 2L-SVM

In this section, we will describe how to fuse multi-view features with the proposed 2L-SVM. Figure 1 illustrates a schematic diagram of the 2L-SVM. As shown in Fig. 1, the proposed 2L-SVM consists of two layers: the 1st layer contains $M$ SVMs, denoted as $\text{SVM}_1$, $\text{SVM}_2$, …, $\text{SVM}_M$, that are trained with the feature sets extracted from $M$ different views; each of the trained 1st layer SVMs learns the discriminative knowledge buried in the corresponding feature set. The 2nd layer has an SVM, denoted as $\text{SVM}_{\text{En}}$, to ensemble the outputs of the 1st layer SVMs and perform the final decision.

Once the 2L-SVM has been trained, the prediction for a query sample can be performed as follows: first, features from $M$ views of the query sample are extracted; then, the $M$ features are fed to corresponding trained 1st layer SVMs; the outputs of the $M$ 1st layer SVMs constitute a so-called "intermediate" $M$-dimensional feature vector, which will be further fed to the trained 2nd layer SVM (i.e., $\text{SVM}_{\text{En}}$) to perform the final decision.

Next, we describe how to train a 2L-SVM with a given training dataset.

Let $(x_i; y_i)$ be the $i$th labeled sample in a given training set $D = \{(x_i; y_i)\}_{i=1}^N$, where $x_i$ is the $i$th sample, $y_i$ is its corresponding class label, and $N$ is the number of samples (e.g., protein sequences in this study). Suppose that we can extract multi-view features for each sample contained in $D$. Let $\left(\mathbf{x}_{i,m}; y_i\right)$ be the data pair representation of the $i$th sample $(1 \le i \le N)$ from the $m$th view, where $\mathbf{x}_{i,m} \in R^{d_m}$ is the feature vector of the $i$th sample extracted from the $m$th $(1 \le m \le M)$ view, $y_i \in \{+1, -1\}$ is the class label, $d_m$ is the dimensionality of the feature, $M$ is the number of views from which protein features are extracted, and $+1$ and $-1$ are the labels of the positive class and negative class, respectively. From the $m$th view, a corresponding training feature set, denoted as $S_m^{\text{tr}} = \left\{\left(\mathbf{x}_{i,m}; y_i\right)\right\}_{i=1}^N$, can be constructed.

Based on the training feature sets extracted from $M$ views (i.e., $S_1^{\text{tr}}$, $S_2^{\text{tr}}$, …, and $S_M^{\text{tr}}$) we can train a 2L-SVM with two consecutive steps as follows:

Step 1: Train the 1st layer SVMs

On each of the $M$ training feature sets, we train a corresponding 1st layer SVM with the standard training algorithm (Vapnik 1998; Chang and Lin 2011). In this study, LIBSVM (version libsvm-3.18) (Chang and Lin 2011), which is freely downloadable at http://www.csie.ntu.edu.tw/~cjlin/libsvm, is used to train SVM models. Unless otherwise stated, all of the SVM models trained in this study take the radial basis function (RBF) as the kernel function, which has been verified to be a better choice in previous studies (Chauhan et al. 2009; Chen et al. 2011; Yu et al. 2013c); the regularization parameter ($c$) and the kernel width parameter ($\gamma$) of RBF were optimized based on ten-fold cross-validation using the grid search strategy provided in LIBSVM (Chang and Lin 2011) on the corresponding training dataset. Detailed values of the parameters used in this study can be found in "Appendix A".

Step 2: Train the 2nd layer SVM

In this step, the key problem is how to generate the feature set for training the 2nd layer SVM (i.e., $\text{SVM}_{\text{En}}$). We propose a strategy to solve this problem as follows.

The proposed strategy generates an "intermediate" $M$-dimensional feature vector from each sample for training the 2nd layer SVM using an "indirect method". Specifically, for each sample $i$ in the training set $D$ we obtain the $m$th component of the "intermediate" feature vector, denoted as $\tilde{o}_{i,m}$, by feeding the $m$th view feature of sample $i$ (i.e., $\mathbf{x}_{i,m}$) into an $\text{SVM}_m^*$ rather than into the trained $\text{SVM}_m$. Let $(\tilde{\mathbf{o}}_i; y_i)$ be the data pair generated from the $i$th sample; then the $N$ data pairs, denoted as $\{(\tilde{\mathbf{o}}_i; y_i)\}_{i=1}^{N}$, constitute the feature set for training the 2nd layer SVM. Importantly, here the data pair $(\mathbf{x}_{i,m}; y_i)$ should not be included in the feature set for the training $\text{SVM}_m^*$.

Then, how to construct the $\text{SVM}_m^*$ is a crucial problem. A feasible solution, called solution 1, is described as follows:

To avoid the underlying over-fitting problem, for each sample $i$, we use an SVM that is trained on the feature set $\tilde{S}_m^{\text{tr}} = S_m^{\text{tr}} - \{(\mathbf{x}_{i,m}; y_i)\}$ rather than $S_m^{\text{tr}}$, denoted as $\text{SVM}_m^*$, to generate $\tilde{o}_{i,m}$ under the input of the $m$th view feature of sample $i$ (i.e., $\mathbf{x}_{i,m}$). By doing so, we can guarantee that the data pair $(\mathbf{x}_{i,m}, y_i)$ has not been used to train its corresponding $\text{SVM}_m^*$. However, this solution is time-consuming and may fail to work, especially, when the dataset is huge. The underlying reason is that the number of SVMs that need to be trained in this solution is $M \times N$, where $M$ and $N$ are the number of views from which features are extracted and the number of samples, respectively. Using the TRAIN1500 dataset in this study as an example, the number of SVMs that need to be trained is 7500 ($5 \times 1500$) if we extract features from five views, which is extremely time-consuming.

To reduce the computational workload, we propose an another more efficient solution, called solution 2, for computing $\text{SVM}_m^*$.

In solution 2 we first partition $S_m^{\text{tr}}$ into $K$ disjoint and equal-sized subsets, denoted as $S_{m,1}, S_{m,2}, \ldots, S_{m,k}, \ldots, S_{m,K}, (1 \le m \le M)$; then, we train an SVM, denoted as $\text{SVM}_{m,k}$ on each feature set defined as $S_m^{\text{tr}} - S_{m,k}$, $(1 \le m \le M, (1 \le k \le K)$ and obtain $M \times K$ trained SVM candidates, denoted as $\{\text{SVM}_{m,k}\}_{m=1,k=1}^{m=M,k=K}$. Next, we can generate the feature set from $D$ for training the 2nd layer SVM. Again, we will take the $m$th view as an example. For each sample $i$ in the training set $D$, we select an $\text{SVM}_{m,k}$ that has been trained on feature set $S_m^{\text{tr}} - S_{m,k}$ with the constraint of $(\mathbf{x}_{i,m}; y_i) \in S_{m,k}$ from the pre-trained SVM set $\{\text{SVM}_{m,k}\}_{k=1}^{k=K}$ as $\text{SVM}_m^*$, where $\mathbf{x}_{i,m}$ and $y_i$ are the $m$th view feature and the label of sample $i$, respectively, and the symbol '$-$' is the set difference operation in the classical set theory; then, we obtain the $m$th component of the "intermediate" feature vector $\tilde{o}_{i,m}$ by feeding the $m$th view feature of sample $i$ (i.e., $\mathbf{x}_{i,m}$) into the $\text{SVM}_m^*$. Note that another sample $i'$ ($i' \neq i$) may also use the same $\text{SVM}_{m,k}$ as its $\text{SVM}_m^*$ for generating the $m$th component of the "intermediate" feature vector, as long as samples $i'$ and $i$ have been partitioned into the same subset (i.e., $S_{m,k}$).

By applying solution 2 we can also avoid the over-fitting problem; however, the number of SVMs that need to be trained is significantly reduced to $M \times K$, where $M$ and $K$ are the number of views from which features are extracted and the number of partitions, respectively. For example, supposing that we also extract features from five views and set $K = 7$, then the number of SVMs that need to be trained on the TRAIN1500 dataset is only 35 ($5 \times 7$), which is substantially smaller than 7500 ($5 \times 1500$).

Figure 2 summarizes the detailed procedure of the algorithm for training a 2L-SVM on a given dataset.

Note that when we set $K$ to be the number of samples in the training dataset, the computational workload of solution 2 will be equal to that of the solution 1. In fact, choosing the value of $K$ is a tradeoff between the performance improvement and the time consumption. Considering this, in practice we suggest that the users choose a relatively small value of $K$ (less than 20); the reason for this suggestion will be further discussed in the section "Choosing the value of $K$ in 2L-SVM". In this study, unless otherwise stated we set $K = 7$ in all of the experiments that utilize 2L-SVM.

## Architecture of the proposed method

In this study, we design and implement a new sequence-based web server called TargetCrys for protein crystallization prediction by applying the ensembling multi-view protein features with proposed 2L-SVM. Figure 3 illustrates the workflow of the proposed TargetCrys.

| | |
|---|---|
| **Input** | $D=\{(x_i;y_i)\}_{i=1}^{N}$ - Training set; $M$ - the number of views; $K$ - the number of partitions |
| **Output** | $\{\text{SVM}_1,\text{SVM}_2,\cdots,\text{SVM}_M,\text{SVM}_{En}\}$ - Trained 2L-SVM, where $\{\text{SVM}_1,\text{SVM}_2,\cdots,\text{SVM}_M\}$ is the set of the trained 1-st layer SVMs, and $\text{SVM}_{En}$ is the trained 2-nd layer SVM. |
| | **Initialization:** Extract feature sets of $D$ from $M$ views |
| 1 | Extract feature sets of $D$ from $M$ views:<br>$S_1^{tr}\leftarrow\{(\mathbf{x}_{i,1};y_i)\}_{i=1}^{N},\ S_2^{tr}\leftarrow\{(\mathbf{x}_{i,2};y_i)\}_{i=1}^{N},\ \cdots,S_m^{tr}\leftarrow\{(\mathbf{x}_{i,m};y_i)\}_{i=1}^{N},\ \cdots,S_M^{tr}\leftarrow\{(\mathbf{x}_{i,M};y_i)\}_{i=1}^{N}$<br>where $\mathbf{x}_{i,m}$ is the *m*-th view feature of sample *i*; |
| | **Step 1:** Train the 1-st layer SVMs |
| 2 | Train a SVM on each view of the extracted feature sets:<br>$\text{SVM}_m\leftarrow\text{Train}(S_m^{tr}),\quad 1\le m\le M$;<br>Then, $\{\text{SVM}_1,\text{SVM}_2,\cdots,\text{SVM}_M\}$ is the set of trained 1-st layer SVMs; |
| | **Step 2:** Train the 2-nd layer SVM |
| | // Pre-train $M\times K$ SVM candidates for generating the feature set for training 2-layer SVM |
| 3 | Partition each of the $M$ view feature sets into $K$ disjoint and equal-sized subsets. Taking the *m*-th view feature set $S_m^{tr}$ as an example, we partition it into $K$ subsets as follows:<br>$\{S_{m,1},S_{m,2},\cdots,S_{m,k},\cdots,S_{m,K}\}\leftarrow\text{partition}(S_m^{tr})$;<br>Then, we totally obtain $M\times K$ subsets, denoted as $\{S_{m,k}\}_{m=1,k=1}^{m=M,k=K}$; |
| 4 | Train a SVM, denoted as $\text{SVM}_{m,k}$ on each feature set defined as $S_m^{tr}-S_{m,k}$, $1\le m\le M$, $1\le k\le K$; then, we obtain $M\times K$ trained SVM candidates, denoted as $\{\text{SVM}_{m,k}\}_{m=1,k=1}^{m=M,k=K}$; |
| | // Generate the feature set from $D$ for training the 2-nd layer SVM |
| 5 | Let $O$ be the feature set generated from $D$ for training the 2-nd layer SVM;<br>Initialize $O$ as an empty set: $O\leftarrow\Phi$; |
| 6 | For each sample *i* in $D$ |
| 7 | Initialize the "intermediate" feature vector obtained from sample *i* as an empty vector: $\tilde{\mathbf{o}}_i\leftarrow(\ )^T$; |
| 8 | For each view *m* |
| 9 | Select a $\text{SVM}_{m,k}$, which has been pre-trained on feature set $S_m^{tr}-S_{m,k}$ with the constraint of $(\mathbf{x}_{i,m};y_i)\in S_{m,k}$, from the trained SVMs set $\{\text{SVM}_{m,k}\}_{k=1}^{k=K}$ as $\text{SVM}_m^*$; |
| 10 | Feed $\mathbf{x}_{i,m}$ to $\text{SVM}_m^*$ and obtain the output $\tilde{o}_{i,m}$; |
| 11 | Augment $\tilde{\mathbf{o}}_i$ with $\tilde{o}_{i,m}$ as follows:<br>$\tilde{\mathbf{o}}_i\leftarrow(\tilde{\mathbf{o}}_i,\tilde{o}_{i,m})^T$; |
| 12 | End For |
| 13 | $O\leftarrow O\cup(\tilde{\mathbf{o}}_i;y_i)$; // $\tilde{\mathbf{o}}_i$ is the "intermediate" feature vector obtained from sample *i* |
| 14 | End For |
| | // Train the 2-nd layer SVM on $O$ |
| 15 | $\text{SVM}_{En}\leftarrow\text{Train}(O)$ |
| 16 | Return $\{\text{SVM}_1,\text{SVM}_2,\cdots,\text{SVM}_M,\text{SVM}_{En}\}$ |

**Fig. 2** Algorithm for training a 2L-SVM on a given dataset

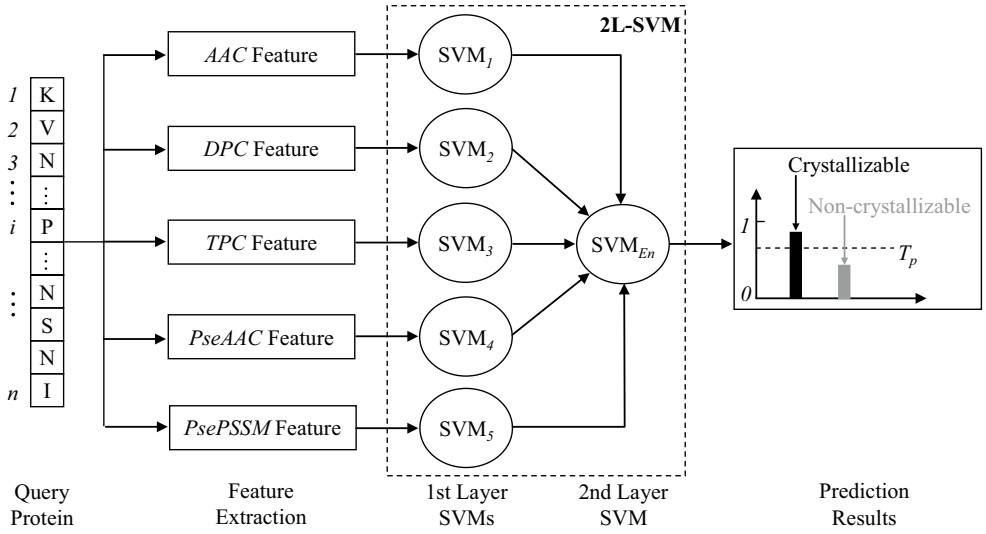


**Fig. 3** Workflow of the proposed TargetCrys

For each protein sequence submitted from the client, the server first extracts its multi-view features including AAC, DPC, TPC, PseAAC, and PsePSSM; then, the five extracted features are fed to their corresponding trained 1st layer SVMs. Next, the outputs of the 1st layer SVMs are sent to the 2nd layer SVM, the output of which is the probability of being a crystallizable protein. The final decision is performed based on the output of the 2nd layer SVM and the prescribed threshold $T$: a protein with probability above the threshold $T$ is marked as crystallizable. How to choose the threshold $T$ will be further discussed in the section "Evaluation indices".

### Evaluation indices

Four routinely used evaluation indices in this field [i.e., Sensitivity (Sn), Specificity (Sp), Accuracy (Acc), and the Mathew's Correlation Coefficient (MCC)] were used to examine the prediction quality of the proposed method as follows:

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{8}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{TN} + \mathrm{FP}} \tag{9}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FN} \cdot \mathrm{FP}}{\sqrt{(\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP})}} \tag{10}$$

where TP, FN, TN, and FP are the abbreviations of true positive, false negative, true negative, and false positive, respectively.

For a soft-type classifier (i.e., a classifier that outputs a continuous numeric value to represent the confidence of a sample belonging to the predicted class), gradually adjusting the classification threshold will produce a series of confusion matrices (Haibo and Garcia 2009). From each confusion matrix, denoted as [TP TN; FP FN], the corresponding Sn, Sp, Acc, and MCC can then be computed (Yu et al. 2013b). In other words, these four evaluation indices are threshold dependent (i.e., their values vary with the threshold chosen). Considering this, the threshold that maximizes the value of MCC of the predictions is used to report the above-mentioned four indices because many other protein attribute predictors including protein crystallization predictors have used this strategy to report their prediction performances (Mizianty and Kurgan 2011; Chen et al. 2012; Hu et al. 2014b). Specifically, we first identify the threshold that maximizes the value of MCC of the predictions on the training subset (e.g., TRAIN3587) using cross-validation, and then the identified threshold was used to evaluate the performance of the proposed method on the corresponding independent validation subset (e.g., TEST3585).

## Experimental results and analysis

### Performance comparisons between features extracted from different views

In this section, we investigate the discriminative capabilities of the features extracted from five different views (i.e.,

**Table 3** Performance comparisons between features extracted from different views with a single SVM as the classifier for TRAIN3587 and TRAIN1500 using five-fold cross-validation

| Dataset | Feature type | Sn (%) | Sp (%) | Acc (%) | MCC |
|---|---|---|---|---|---|
| TRAIN3587 | AAC | 70.27 | 67.60 | 68.50 | 0.34 |
| | DPC | 48.42 | 90.81 | 76.58 | 0.45 |
| | TPC | 60.63 | 85.14 | 76.92 | 0.47 |
| | PseAAC | 62.87 | 72.35 | 69.17 | 0.34 |
| | PsePSSM | 46.26 | 88.67 | 74.44 | 0.39 |
| | Serially combined feature[a] | 53.74 | 88.08 | 76.55 | 0.45 |
| TRAIN1500 | AAC | 88.36 | 63.84 | 76.20 | 0.54 |
| | DPC | 90.21 | 62.10 | 76.27 | 0.55 |
| | TPC | 92.06 | 50.94 | 71.67 | 0.47 |
| | PseAAC | 90.61 | 62.10 | 76.47 | 0.55 |
| | PsePSSM | 85.98 | 77.42 | 81.73 | 0.64 |
| | Serially combined feature[a] | 87.43 | 75.94 | 81.73 | 0.64 |

[a] The serial combination of AAC, DPC, TPC, PseAAC, and PsePSSM features

AAC, DPC, TPC, PseAAC, and PsePSSM) (refer to the section "Multi-view feature representation"). Each feature was evaluated by performing five-fold cross-validation on both the TRAIN3587 and TRAIN1500 datasets with a single SVM as the classifier. We also evaluated the discriminative capability of the integrated feature that was obtained by serially combining the five features using the same procedure. Unless otherwise stated, all the results of the threshold dependent evaluation indices presented in this study were reported by choosing the threshold that maximized the value of *MCC* of the predictions on the training subset using the five-fold cross-validation (refer to the section "Evaluation indices").

Table 3 summarizes the performance comparisons between the five different features for both TRAIN3587 and TRAIN1500. From Table 3, several observations can be made as follows:

First, although each of the five considered features can be effectively used to perform protein crystallization predictions (Acc $\geq$68.50 %, MCC $\geq$0.34), none consistently performs best on the two training subsets. For example, the *TPC* feature performed best on TRAIN3587 with Acc $\geq$76.92 % and MCC = 0.47, whereas the PsePSSM feature achieved the best performance on TRAIN1500 with Acc = 81.73 % and MCC = 0.64. This observation suggests that the optimal feature representation of protein sequences varies with the training datasets used, and thus should be carefully considered when building a machine-learning-based protein crystallization predictor.

Second, among the five considered features we found that the PsePSSM feature acts as the best and the third-best performer on TRAIN3587 and TRAIN1500, respectively. This observation empirically demonstrates that the sequence evolutionary information (encoded in the PsePSSM feature) that has not been investigated by previous predictors can also be effectively used to perform protein crystallization predictions.

Third, serial combination of the five features does not definitely improve the prediction performance as expected. This observation is quite consistent with the conclusion made by Chen et al. (2008) that directly combining different features will, in most cases, lead to an "intermediate" prediction accuracy (i.e., the prediction accuracy of the combined features lies between the worst and best prediction accuracies of the individual features). The underlying reason is that the combination of features will simultaneously increase the information redundancy that could, in turn, deteriorate the prediction accuracy (Kohavi and John 1997). In view of this, researchers tend to perform either a feature selection technique to select the most discriminative feature components or a feature extraction technique to extract more discriminative feature from the serially combined features to further improve prediction performance (Saeys et al. 2007).

In summary, all of the five considered features contain useful information for protein crystallization predictions, and the simple serial combination of these features does not improve the prediction performance. Although feature extraction and feature selection may potentially enhance the discriminative capability of the combined features and improve the prediction performance, we suggest fusing useful information buried in the multi-view features with our newly proposed 2L-SVM, as will be illustrated in the subsequent sections.

## Enhancing prediction performance using the proposed 2L-SVM

In this section, we demonstrate the efficacy of the proposed 2L-SVM for improving the performance of protein crystallization predictions.

On each of the training subsets TRAIN3587 and TRAIN1500, we performed performance comparisons between the optimal feature, the serially combined features, and the features fused by 2L-SVM using five-fold cross-validation, as shown in Table 4. When applying 2L-SVM, the value of $K$ (i.e., the number of partitions) was set to 7; the reason for this setting will be discussed in the section "Choosing the value of $K$ in 2L-SVM".

From Table 4, two observations can be made as follows:

First, the prediction performance was remarkably improved by applying the proposed 2L-SVM. Compared with the serially combined feature, improvements of

**Table 4** Performance comparisons between the optimal feature, the serially combined feature, and the feature fused by 2L-SVM for TRAIN3587 and TRAIN1500 using five-fold cross-validation

| Dataset | Method | Sn (%) | Sp (%) | Acc (%) | MCC |
|---|---|---|---|---|---|
| TRAIN3587 | Optimal feature (TPC) + SVM | 60.63 | 85.14 | 76.92 | 0.47 |
| | Serially combined feature + SVM | 53.74 | 88.08 | 76.55 | 0.45 |
| | Feature fused by 2L-SVM | 58.97 | 91.73 | 80.74 | 0.55 |
| TRAIN1500 | Optimal feature (PsePSSM) + SVM | 85.98 | 77.42 | 81.73 | 0.64 |
| | Serially combined feature + SVM | 87.43 | 75.94 | 81.73 | 0.64 |
| | Feature fused by 2L-SVM | 88.23 | 77.96 | 83.13 | 0.67 |

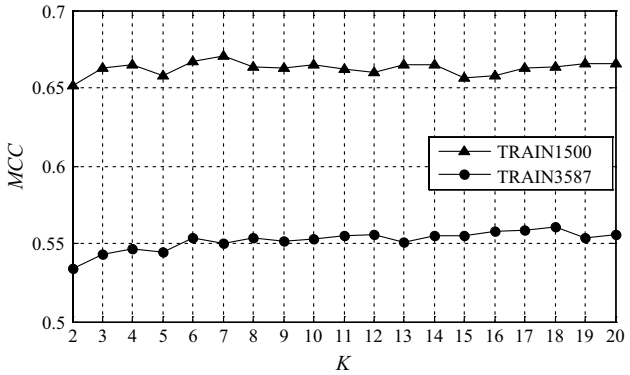


**Fig. 4** The performance variation curves of *MCC* versus *K* on TRAIN3587 and TRAIN1500

10 and 3 % in *MCC* were achieved by the feature fused by 2L-SVM on TRAIN3587 and TRAIN1500, respectively. Additionally, the feature fused by 2L-SVM consistently performs better than the optimal feature (i.e., TPC on TRAIN3587 and PsePSSM on TRAIN1500) with improvements of 8 and 3 % in MCC, respectively. The Sn of the optimal feature (TPC) on TRAIN3587 is 60.63 %, which is 1.66 % higher than that of the feature fused by 2L-SVM; however, the Sp of the TPC is only 85.14 %, which is strikingly lower than that (i.e., 91.73 %) of the feature fused by 2L-SVM, denoting too many false positives were induced with the optimal feature (TPC).

## Choosing the value of *K* in 2L-SVM

In the section "Multi-view feature fusion with 2L-SVM", we declared that the choice of the value of *K* (i.e., the number of partitions) in 2L-SVM is a tradeoff between performance improvement and time consumption. We also suggested that the users should choose a relatively small value of *K*. In this section, we will empirically demonstrate the reason for this suggestion.

We evaluated the performance variations of the proposed 2L-SVM by gradually varying the value of *K* from 2 to 20 with a step size of 1. For each $K \in [2, 20]$, we evaluated the MCC performance of the 2L-SVM on TRAIN3587 using a five-fold cross-validation strategy. The performance variation curve of MCC versus *K* is plotted in Fig. 4.

From Fig. 4, we can see that the overall trend of the values of MCC tends to increase with the increment of *K* ($K \leq 7$) with slight fluctuations for both TRAIN3587 and TRAIN1500. When $K > 7$, the value of MCC on TRAIN1500 fluctuates and no improvements can be observed; similarly, no significant improvements in MCC can be observed for TRAIN3587. In view of this, we will set $K = 7$ in all the experiments that utilize 2L-SVM in this study.

## Comparisons with existing predictors

In this section, we will compare the proposed method with other popular protein crystallization predictors including OB-Score (Overton and Barton 2006), XtalPred (Slabinski et al. 2007), ParCrys (Overton et al. 2008), MetaPPCP (Mizianty and Kurgan 2009), CRYSTALP2 (Kurgan et al. 2009), SVMCRYS (Kandaswamy et al. 2010), PPCpred (Mizianty and Kurgan 2011), RFCRYS (Jahandideh and Mahdavi 2012), SCMCRYS (Charoenkwan et al. 2013), and PPCinter (Gao et al. 2014). For the convenience of the subsequent descriptions, we term the proposed method as TargetCrys.

### *Performance comparisons of the cross-validation test*

First, we compare the proposed TargetCrys with other protein crystallization predictors by performing a cross-validation test on the training subsets listed in Table 2. Because none of the existing predictors have reported prediction performances using TRAIN3587 in cross-validation tests, we only present the comparison results for TRAIN1500 using five-fold cross-validation as has been reported for other predictors such as MetaPPCP (Mizianty and Kurgan 2009). Table 5 summarizes the comparison results between TargetCrys and the other predictors.

Note that the results of OB-Score (Overton and Barton 2006), XtalPred (Slabinski et al. 2007), CRYSTALP2 (Kurgan et al. 2009), and ParCrys (Overton et al. 2008) were obtained by feeding TRAIN1500 sequences to their web servers. Based on the results shown in Table 5, the proposed

**Table 5** Performance comparisons between the proposed TargetCrys and other popular predictors on TRAIN1500 using five-fold cross-validation. [a]Data excerpted from (Mizianty and Kurgan 2009)

| Predictor | Sn (%) | Sp (%) | Acc (%) | MCC |
|---|---|---|---|---|
| OB-score (Overton and Barton 2006)[a] | 85.00 | 53.00 | 68.80 | 0.39 |
| XtalPred (Slabinski et al. 2007)[a] | 75.00 | 63.00 | 69.27 | 0.39 |
| CRYSTALP2 (Kurgan et al. 2009)[a] | 77.00 | 62.00 | 69.60 | 0.40 |
| ParCrys (Overton et al. 2008)[a] | 83.00 | 56.00 | 69.73 | 0.41 |
| MetaPPCP (Mizianty and Kurgan 2009) | 88.00 | 69.00 | 78.40 | 0.58 |
| TargetCrys | 88.23 | 77.96 | 83.13 | 0.67 |

**Table 6** Performance comparisons between TargetCrys and other popular protein crystallization predictors on the independent validation subset TEST3585. [a]Data excerpted from (Mizianty and Kurgan 2011)

| Predictor | Sn (%) | Sp (%) | Acc (%) | MCC |
|---|---|---|---|---|
| ParCrys (Overton et al. 2008)[a] | 78.60 | 31.80 | 47.50 | 0.11 |
| OB-score (Overton and Barton 2006)[a] | 80.30 | 31.40 | 47.80 | 0.12 |
| CRYSTALP2 (Kurgan et al. 2009)[a] | 74.40 | 45.70 | 55.30 | 0.20 |
| MetaPPCP (Mizianty and Kurgan 2009)[a] | 61.70 | 59.00 | 59.90 | 0.20 |
| SVMCRYS (Kandaswamy et al. 2010)[a] | 75.20 | 46.70 | 56.30 | 0.21 |
| XtalPred (Slabinski et al. 2007)[a] | 67.00 | 62.30 | 63.90 | 0.28 |
| SCMCRYS (Charoenkwan et al. 2013) | 46.00 | 91.00 | 76.10 | 0.44 |
| PPCpred (Mizianty and Kurgan 2011) | 61.20 | 84.80 | 76.80 | 0.47 |
| RFCRYS (Jahandideh and Mahdavi 2012) | 51.00 | 95.00 | 80.00 | 0.53 |
| PPCinter (Gao et al. 2014) | 61.70 | 89.80 | 80.40 | 0.55 |
| TargetCrys | 57.97 | 92.65 | 81.00 | 0.56 |

TargetCrys achieved much better prediction performance than the other five considered protein crystallization predictors [OB-Score (Overton and Barton 2006), XtalPred (Slabinski et al. 2007), CRYSTALP2 (Kurgan et al. 2009), ParCrys (Overton et al. 2008), and MetaPPCP (Mizianty and Kurgan 2009)]. The values of the four evaluation indices of TargetCrys are consistently superior to those of the other predictors. Compared with the second-best performer (i.e., MetaPPCP Mizianty and Kurgan 2009) TargetCrys achieved improvements of 8.96, 4.73, and 9.0 % on Sp, Acc, and MCC, respectively, while possessing a comparable performance on Sn (88.23 % of TargetCrys vs. 88.00 % of MetaPPCP).

*Performance comparisons over independent validation tests*

Performing independent validation tests is often the mandatory procedure to evaluate the generalization capability of a trained predictor. In view of this, we also performed independent validation tests in this study. First, we trained TargetCrys on each of the two training subsets TRAIN3587 and TRAIN1500; then, the trained TargetCrys was tested with the corresponding independent validation subset(s) listed in Table 2. Note that to fairly compare with other predictors and avoid potential overestimation of TargetCrys, the same threshold identified during the cross-validation test on each of the two training subsets was used to perform the corresponding independent validation test.

Performance comparisons between TargetCrys and other predictors on the independent validation subset TEST3585 are summarized in Table 6. Note that portions of the results indicated by asterisks in Table 6 are excerpted from Mizianty and Kurgan' work (Mizianty and Kurgan 2011), where they obtained the results of ParCrys (Overton et al. 2008), OB-Score (Overton and Barton 2006), CRYSTALP2 (Kurgan et al. 2009), MetaPPCP (Mizianty and Kurgan 2009), and SVMCRYS (Kandaswamy et al. 2010) by feeding TEST3585 sequences into their corresponding web servers (Mizianty and Kurgan 2011), whereas SCMCRYS (Charoenkwan et al. 2013), PPCpred (Mizianty and Kurgan 2011), RFCRYS (Jahandideh and Mahdavi 2012), PPCinter (Gao et al. 2014), and TargetCrys were all trained with TRAIN3587 and then tested with the independent validation subset TEST3585.

Based on the results in Table 6, TargetCrys again achieved the best performance. For example, TargetCrys achieved 81.00 % and 0.56 on Acc and MCC, respectively, which represent the two overall measurements of the quality of the binary predictions. The Acc and MCC results of TargetCrys demonstrate that TargetCrys is competitive to the state-of-the-art protein crystallization predictors such as RFCRYS (Jahandideh and Mahdavi 2012) and PPCinter (Gao et al. 2014), and consistently outperforms the other listed predictors.

To further demonstrate the superiority of TargetCrys over the existing predictors, TEST500, which is a common subset and has been utilized to perform independent validation tests for several recently described predictors including RFCRYS (Jahandideh and Mahdavi 2012), MetaPPCP (Mizianty and Kurgan 2009), and CRYSTALP2 (Kurgan et al. 2009), was also used to evaluate the generalization performance of TargetCrys.

Table 7 summarizes the performance comparisons between TargetCrys and other popular protein crystallization predictors on the independent validation subset

**Table 7** Performance comparisons between TargetCrys and other popular protein crystallization predictors on the independent validation subset TEST500. [a]Data excerpted from (Mizianty and Kurgan 2009)

| Predictor | Sn (%) | Sp (%) | Acc (%) | MCC |
|---|---|---|---|---|
| CRYSTALP2 (Kurgan et al. 2009)[a] | 73.00 | 64.00 | 68.40 | 0.37 |
| XtalPred (Slabinski et al. 2007)[a] | 77.00 | 68.00 | 72.40 | 0.45 |
| ParCrys (Overton et al. 2008)[a] | 84.00 | 63.00 | 73.40 | 0.48 |
| OB-score (Overton and Barton 2006)[a] | 89.00 | 58.00 | 73.00 | 0.49 |
| RFCRYS (Jahandideh and Mahdavi 2012) | 86.00 | 75.00 | 80.40 | 0.61 |
| MetaPPCP (Mizianty and Kurgan 2009) | 89.00 | 73.00 | 81.00 | 0.63 |
| TargetCrys | 90.57 | 80.08 | 85.20 | 0.71 |

TEST500. Note that the results of the OB-Score (Overton and Barton 2006), ParCrys (Overton et al. 2008), CRYSTALP2 (Kurgan et al. 2009), and XtalPred (Slabinski et al. 2007) listed in Table 7 were obtained by feeding TEST500 sequences to their corresponding web servers as described in (Mizianty and Kurgan 2009), whereas MetaPPCP (Mizianty and Kurgan 2009), RFCRYS (Jahandideh and Mahdavi 2012), and TargetCrys were trained with TRAIN1500 and then tested with the independent validation subset TEST500.

From Table 7 we can see that TargetCrys significantly outperformed the other six considered predictors and acted as the best performer. The values of MCC of TargetCrys on TEST500 is 0.71, which is 8 % higher than that of the second-best performer MetaPPCP (Mizianty and Kurgan 2009).

In summary, the results listed in Table 6 and 7 effectively demonstrate the good generalization capability of the proposed TargetCrys.

## Conclusions

In this study, we propose a 2L-SVM ensemble scheme for fusing multi-view features in protein crystallization predictions. The proposed 2L-SVM switches the feature-level fusion problem to a decision-level problem: the SVMs in the 1st layer are trained on each of the multi-view features, and the outputs of the 1st layer SVMs, which are the initial decisions obtained on individual view features, are further ensembled by the 2nd layer SVM. Based on the proposed 2L-SVM, we developed a new sequence-based predictor called TargetCrys for performing protein crystallization predictions. Cross-validation tests and independent validation tests on several benchmark datasets demonstrated the efficacy of the proposed 2L-SVM ensemble scheme. The results showed that the proposed TargetCrys outperforms most existing protein crystallization predictors and is competitive with the state-of-the-art predictors.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China (No. 61373062, 61175024, 61222306, and 61233011), the Natural Science Foundation of Jiangsu (No. BK20141403), the Jiangsu University Graduate Research and Innovation Project (No. KYZZ_0123), Jiangsu Postdoctoral Science Foundation (No. 1201027C), the Science and Technology Commission of Shanghai Municipality (No. 16JC1404300), "The Six Top Talents" of Jiangsu Province (No. 2013-XXRJ-022), and the Fundamental Research Funds for the Central Universities (No. 30916011327). D. J. Yu is the corresponding author for this paper.

**Compliance with ethical standards**

**Conflict of interest** The authors declare that they have no conflict of interest.

## Appendix A: Parameters of TargetCrys

See Table 8.

**Table 8** Parameters of SVM models identified with the grid search program of LIBSVM software on TRAIN3587 and TRAIN1500. [a]$SVM_{AAC}$, $SVM_{DPC}$, $SVM_{TPC}$, $SVM_{PseAAC}$, and $SVM_{PsePSSM}$ represent SVM models trained with AAC feature, DPC feature, TPC feature, PseAAC feature, and PsePSSM feature, respectively

| Dataset | Parameter | $SVM^{a}_{AAC}$ | $SVM^{a}_{DPC}$ | $SVM^{a}_{TPC}$ | $SVM^{a}_{PseAAC}$ | $SVM^{a}_{PsePSSM}$ | $SVM^{b}_{En}$ |
|---|---|---|---|---|---|---|---|
| TRAIN3587 | $c$ | $2^{15}$ | $2^{3}$ | $2^{7.5}$ | $2^{9}$ | $2^{9}$ | $2^{-1}$ |
| | $\gamma$ | $2^{-1}$ | $2^{3}$ | $2^{0}$ | $2^{1}$ | $2^{-5}$ | $2^{3}$ |
| TRAIN1500 | $c$ | $2^{7}$ | $2^{3}$ | $2^{15}$ | $2^{9}$ | $2^{2}$ | $2^{9}$ |
| | $\gamma$ | $2^{3}$ | $2^{3}$ | $2^{-9}$ | $2^{1}$ | $2^{1}$ | $2^{-7}$ |

[b] $SVM_{En}$ represents the second-layer SVM model in proposed 2L-SVM

## References


Babnigg G, Joachimiak A (2010) Predicting protein crystallization propensity from protein sequence. J Struct Funct Genomics 11(1):71–80

Berman HM et al (2000) The Protein Data Bank. Nucleic Acids Res 28(1):235–242

Bradshaw NI et al (2012) 15: 30 structural elucidation of disc1 pathway proteins using electron microscopy, chemical cross-linking and mass spectroscopy. Schizophr Res 136:S74

Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27

Charoenkwan P, Shoombuatong W, Lee HC, Chaijaruwanich J, Huang HL, Ho SY (2013) SCMCRYS: predicting protein crystallization using an ensemble scoring card method with estimating propensity scores of P-collocated amino acid pairs. PLoS One 8(9):e72368

Chauhan JS, Mishra NK, Raghava GP (2009) Identification of ATP binding residues of a protein from its primary sequence. BMC Bioinform 10:434

Chen K, Kurgan L, Rahbari M (2007) Prediction of protein crystallization using collocation of amino acid pairs. Biochem Bioph Res Co 355(3):764–769

Chen C, Chen LX, Zou XY, Cai PX (2008) Predicting protein structural class based on multi-features fusion. J Theor Biol 253(2):388–392

Chen K, Mizianty MJ, Kurgan L (2011) ATPsite: sequence-based prediction of ATP-binding residues. Proteome Sci 9(Suppl 1):S4

Chen K, Mizianty MJ, Kurgan L (2012) Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors. Bioinformatics 28(3):331–341

Chou KC (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins Struct Funct Genetics 43(3):246–255

Chou K-C (2004) Structural bioinformatics and its impact to biomedical science. Curr Med Chem 11(16):2105–2134

Chou KC (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21(1):10–19

Chou K-C, Shen H-B (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem Bioph Res Co 360(2):339–345

Dieckmann A, Rieskamp J (2007) The influence of information redundancy on probabilistic inferences. Memory Cogn 35(7):1801–1813

Ding C, Yuan L-F, Guo S-H, Lin H, Chen W (2012) Identification of mycobacterial membrane proteins and their types using over-represented tripeptide compositions. J Proteom 77:321–328

Foulonneau M (2007) Information redundancy across metadata collections. Inf Process Manage 43(3):740–751

Gao JZ, Hu G, Wu ZH, Ruan JS, Shen SY, Hanlon M, Wang K (2014) Improved prediction of protein crystallization, purification and production propensity using hybrid sequence representation. Curr Bioinform 9(1):57–64

Gromiha MM (2010) Protein bioinformatics: from sequence to function. Academic Press, Cambridge

Haibo H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284

Hu G et al (2014a) Human structural proteome-wide characterization of Cyclosporine A targets. Bioinformatics 30(24):3561–3566

Hu J, He X, Yu D-J, Yang X-B, Yang J-Y, Shen H-B (2014b) A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction. PLoS One 9(9):e107676

Jackman L (2012) Dynamic nuclear magnetic resonance spectroscopy. Elsevier, New York

Jahandideh S, Mahdavi A (2012) RFCRYS: sequence-based protein crystallization propensity prediction by means of random forest. J Theor Biol 306:115–119

Kandaswamy KK, Pugalenthi G, Suganthan PN, Gangal R (2010) SVMCRYS: an SVM approach for the prediction of protein crystallization propensity from protein sequence. Protein Peptide Lett 17(4):423–430

Kantardjieff KA, Rupp B (2004) Protein isoelectric point as a predictor for increased crystallization screening efficiency. Bioinformatics 20(14):2162–2168

Kantardjieff KA, Jamshidian M, Rupp B (2004) Distributions of pI versus pH provide prior information for the design of crystallization screening experiments: response to comment on 'Protein isoelectric point as a predictor for increased crystallization screening efficiency'. Bioinformatics 20(14):2171–2174

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324

Kurgan L, Razib AA, Aghakhani S, Dick S, Mizianty M, Jahandideh S (2009) CRYSTALP2: sequence-based protein crystallization propensity prediction. BMC Struct Biol 9:50

Mizianty MJ, Kurgan L (2009) Meta prediction of protein crystallization propensity. Biochem Bioph Res Co 390(1):10–15

Mizianty MJ, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification and production propensity. Bioinformatics 27(13):i24–i33

Mizianty MJ, Kurgan LA (2012) CRYSpred: accurate sequence-based protein crystallization propensity prediction using sequence-derived structural characteristics. Protein Pept Lett 19(1):40–49

Mizianty MJ, Fan X, Yan J, Chalmers E, Woloschuk C, Joachimiak A, Kurgan L (2014) Covering complete proteomes with X-ray structures: a current snapshot. Biol Crystallogr 70(11):2781–2793

Nanni L, Lumini A, Gupta D, Garg A (2012) Identifying bacterial virulent proteins by fusing a set of classifiers based on variants of Chou's pseudo amino acid composition and on evolutionary information. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 9(2):467–475

Overton IM, Barton GJ (2006) A normalised scale for structural genomics target ranking: the OB-Score. FEBS Lett 580(16):4005–4009

Overton IM, Padovani G, Girolami MA, Barton GJ (2008) ParCrys: a Parzen window density estimation approach to protein crystallization propensity prediction. Bioinformatics 24(7):901–907

Price Ii WN et al (2009) Understanding the physical properties that control protein crystallization by analysis of large-scale experimental data. Nat Biotechnol 27(1):51–57

Rodrigues A, Hubbard RE (2003) Making decisions for structural genomics. Brief Bioinform 4(2):150–167

Roy A, Zhang Y (2012) Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. Structure 20(6):987–997

Rung J, Brazma A (2013) Reuse of public genome-wide gene expression data. Nat Rev Genet 14(2):89–99

Rupp B, Wang J (2004) Predictive models for protein crystallization. Methods 34(3):390–407

Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Schaffer AA et al (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res 29(14):2994–3005

Service R (2005) Structural biology. Structural genomics, round 2. Science 307(5715):1554–1558

Shen H-B, Chou K-C (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem 373(2):386–388

Singh H, Chauhan JS, Gromiha MM, Raghava GP (2011) ccPDB: compilation and creation of data sets from Protein Data Bank. Nucleic Acids Res gkr1150

Slabinski L, Jaroszewski L, Rychlewski L, Wilson IA, Lesley SA, Godzik A (2007) XtalPred: a web server for prediction of protein crystallizability. Bioinformatics 23(24):3403–3405

Smialowski P, Schmidt T, Cox J, Kirschner A, Frishman D (2006) Will my protein crystallize? A sequence-based predictor. Proteins 62(2):343–355

Todd AE, Marsden RL, Thornton JM, Orengo CA (2005) Progress of structural genomics initiatives: an analysis of solved target structures. J Mol Biol 348(5):1235–1260

Tramontano A, Cozzetto D (2004) The relationship between protein sequence, structure and function: protein function prediction. Supramolecular Struct Funct 8:15–29

Vapnik VN (ed) (1998) Statistical learning theory. Wiley, New York

Yu D, Wu X, Shen H, Yang J, Tang Z, Qi Y (2012) Enhancing membrane protein subcellular localization prediction by parallel fusion of multi-view features. IEEE Trans Nanobioscience 11(4):375–385

Yu D-J et al (2013a) Learning protein multi-view features in complex space. Amino Acids 44(5):1365–1379

Yu DJ, Hu J, Huang Y, Shen HB, Qi Y, Tang ZM, Yang JY (2013b) TargetATPsite: a template-free method for ATP-binding sites prediction with residue evolution image sparse representation and classifier ensemble. J Comput Chem 34(11):974–985

Yu DJ, Hu J, Tang ZM, Shen HB, Yang J, Yang JY (2013c) Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. Neurocomputing 104:180–190

Zhang Y (2014) Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10. Proteins Struct Funct Bioinform 82(S2):175–187

Zucker FH et al (2010) Prediction of protein crystallization outcome using a hybrid method. J Struct Biol 171(1):64–73