Published online 11 September 2006 *Nucleic Acids Research, 2006, Vol. 34, No. 17* ***4655–4666***
doi:10.1093/nar/gkl638

PLPD: reliable protein localization prediction from imbalanced and overlapped datasets

KiYoung Lee[1,2], Dae-Won Kim[3], DoKyun Na[1], Kwang H. Lee[1,2] and Doheon Lee[1,*]

[1]Department of BioSystems, KAIST, Daejeon City, Republic of Korea, [2]Advanced Information Technology Research Center, KAIST, Daejeon City, Republic of Korea and [3]School of Computer Science and Engineering, Chung-Ang University, Seoul City, Republic of Korea

Received May 5, 2006; Revised July 25, 2006; Accepted August 10, 2006

ABSTRACT

Subcellular localization is one of the key functional characteristics of proteins. An automatic and efficient prediction method for the protein subcellular localization is highly required owing to the need for large-scale genome analysis. From a machine learning point of view, a dataset of protein localization has several characteristics: the dataset has too many classes (there are more than 10 localizations in a cell), it is a multi-label dataset (a protein may occur in several different subcellular locations), and it is too imbalanced (the number of proteins in each localization is remarkably different). Even though many previous works have been done for the prediction of protein subcellular localization, none of them tackles effectively these characteristics at the same time. Thus, a new computational method for protein localization is eventually needed for more reliable outcomes. To address the issue, we present a protein localization predictor based on D-SVDD (PLPD) for the prediction of protein localization, which can find the likelihood of a specific localization of a protein more easily and more correctly. Moreover, we introduce three measurements for the more precise evaluation of a protein localization predictor. As the results of various datasets which are made from the experiments of Huh *et al.* (2003), the proposed PLPD method represents a different approach that might play a complimentary role to the existing methods, such as Nearest Neighbor method and discriminate covariant method. Finally, after finding a good boundary for each localization using the 5184 classified proteins as training data, we predicted 138 proteins whose subcellular localizations could not be clearly observed by the experiments of Huh *et al.* (2003).

INTRODUCTION

Recent advances in large-scale genome sequencing have resulted in the huge accumulation of protein amino acid sequences (1). Currently, many researchers are trying either to discover or to clarify the unknown functions of these proteins. Since knowing the subcellular localization where a protein resides can give important insight into its possible functions (2), it is indispensable to identify the subcellular localization of a protein. However, it is time consuming and costly to identify the subcellular localizations of newly found proteins entirely by performing experimental tests. Thus, a reliable and efficient computational method is highly required to directly extract localization information.

From a machine learning point of view, a task that predicts the subcellular localizations of given proteins has several characteristics which demonstrate the task's complexity (see Table 1). First, there are too many localizations in a cell. For example, according to the work of Huh *et al*. (3), there are 22 distinct subcellular localization categories in budding yeast. It means that the possibility of correct prediction of one localization is <4.55% with a random guess. Second, the prediction task is a 'multi-label' classification problem; some proteins may have several different subcellular localizations (1). For instance, the YBR156C can be located either in 'microtubule' or 'nucleus' according to the work of Huh *et al*. (3). Thus, a computational method should be able to handle the multi-label problem. Finally, the number of proteins in each localization is too different making a protein localization data set highly 'imbalanced'. It is generally accepted that proteins located in some organelles are much more abundant than in others (2). It also can be checked with the data of Huh *et al*. (3); the number of proteins in 'cytoplasm' is 1782, while the number of proteins in 'ER to Golgi' is only 6 (see the second column of Table 1). All these three characteristics make the task difficult. Thus, not only good features for a protein but also a good computational algorithm is ultimately needed for the reliable prediction of protein subcellular localization.

Actually, many works have been done during the last decade or so in this field. The efforts in these works have followed several trends (see Table 2).

*To whom correspondence should be addressed. Tel: +82 42 869 4316; Fax: +82 42 869 8680; Email: dhlee@biosoft.kaist.ac.kr

Table 1. The number of proteins in the original Huh *et al*. Dataset (2003) and three training datasets

Subcellular localization	Huh *et al*. Dataset	Dataset-I	Dataset-II	Dataset-III
1. Actin	32	32	27	27
2. Bud	25	25	19	19
3. Bud neck	61	61	48	48
4. Cell periphery	130	130	98	98
5. Cytoplasm	1782	1782	1472	1472
6. Early golgi	54	54	39	39
7. Endosome	46	46	37	37
8. ER	292	292	207	207
9. ER to golgi	6	6	5	5
10. Golgi	41	41	30	30
11. Late golgi	44	44	38	38
12. Lipid particle	23	23	15	15
13. Microtubule	20	20	17	17
14. Mitochondrion	522	522	389	389
15. Nuclear periphery	60	60	38	38
16. Nucleolus	164	164	122	122
17. Nucleus	1446	1446	1126	1126
18. Peroxisome	21	21	16	16
19. Punctate composite	137	137	91	91
20. Spindle pole	61	61	27	27
21. Vacuolar membrane	58	58	47	47
22. Vacuole	159	159	124	124
Total number of classified proteins, $\tilde{N}$	5184	5184	4032	4032
Total number of different proteins, N	3914	3914	3017	3017
Dimension of features		9620D	2372D	11992D
Coverage		100%	77.08% $\left(\frac{3017}{3914}\right)$	77.08% $\left(\frac{3017}{3914}\right)$

1) *Feature Extraction:* one trend is to try to extract good information (or features) from given proteins. One category of the features used is based on amino acid composition (AA) (1,2,4–14,15–22). Many works have used AA as the unique feature or the complementary feature of a protein owing to its simplicity and its high coverage. Prediction based on only AA features would lose sequence order information. Thus, to give sequential information to the AA, Nakashima and Nishikawa (4) also used amino acid pair composition (PairAA), Chou (5) used pseudo amino acid composition (PseAA) using sequence-order correlation (SOC) factor (6), and Park and Kanehisa (18) also used gapped amino acid composition (GapAA). For more sequence order effect, Pan *et al*. (19) used digital signal processing filter technique to the PseAA. These features based on AA have the advantage of achieving a very high coverage but may have limit on the high performance. Other researchers have used several kinds of motif information as the feature of proteins (1,9–12,23,24–26). Since Nakai and Horton (23) used protein sorting signal motifs in the N-terminal portion of a protein, some researchers (24) have used the motif in the prediction of localization. Using the 2005 functional domain sequences of SBASE-A which is a collection of well known structural and functional domain types (28), Chou and Cai (9,25) represented a protein as a vector with a 2005-dimensional functional domain composition (SBASE-FunD). They also introduced the dimensional functional domain composition (InterPro-FunD) (1,11) using the InterPro database (29). In contrast, Nair and Rost (26) represented a protein with functional annotations from the SWISS-PROT database (30). Recently, for higher prediction accuracy Cai and Chou (1,11) used Gene Ontology (GO) term as a auxiliary feature of a protein. Even though motif information and GO can improve the prediction accuracy, the information has a limited coverage of the proteins.

2) *Class coverage extension:* another trend is to increase the coverage of protein localization for practical use. At the beginning, Nakashima and Nishikawa (4) distinguished between intracellular proteins and extracellular proteins using the AA and the PairAA features. After that, many researchers enlarged the number of localization classes to 5 classes (13), to 8 classes (27), to 11 classes (23), to 12 classes (2), then to 14 classes (14). Recently Chou and Cai (1) used up to 22 localization classes using the dataset of Huh *et al*. (3), which is the biggest coverage of protein localization up to now.

3) *Computational algorithm:* to improve the prediction quality, another trend is to try to use an efficient computational algorithm in the prediction stage. Current computational methods include the following: a Least Distance Algorithm using various distance measures [a distance in PlotLock (31) that is modified from Mahalanobis distance originally introduced by Chou in predicting protein structural class (32), a Covariant discriminant algorithm (CD) in (2,27), and an augmented CD in (6)], an Artificial Neural Network approach in (15,20), a Nearest Neighbor approach in (1,11,17,23,26), a Markov Model (MM) in (21), a Bayesian Network (BN) approach in (24), and Support Vector Machines (SVMs) approach in (7,9,16,18,25). In (12), three algorithms, such as SVMs, a Hidden MM and a BN are used for improving prediction accuracy.

Even though many previous works have been done for the prediction of protein subcellular localization, none of them tackled effectively the three characteristics of protein localization prediction at the same time. For example, many existing predictors use only less than five different subcellular localizations. Moreover, very few predictors deal with the issue of multiple-localization proteins except for Chou and Cai (1). The majority only assumed that there is no multiple-localization protein. Furthermore, almost all previous methods did not consider the imbalanced problem in a given dataset. That means these methods achieve high accuracy only for the most populated localizations, such as the 'nucleus' and 'cytosol'. They, however, are generally less accurate on the numerous localizations containing fewer individual proteins. Thus, a new computational method is eventually needed for more reliable prediction which should have the following characteristics: (i) it can show relatively good performance in case many classes exist, (ii) it can handle a multi-label problem and (iii) it should be robust in an imbalanced dataset. Our study is aimed to address these issues.

To achieve the purpose, we developed a PLPD method which can predict better the localization information of proteins using a Density-induced Support Vector Data Description (D-SVDD) approach. The PLPD stands for 'Protein Localization Predictor based on D-SVDD'. The D-SVDD (33) is a general extension of conventional Support Vector Data Description (C-SVDD) (34–36) inspired by the SVMs (37). According to the work of Lee *et al*. (33), D-SVDD

Table 2. The previous researches in the prediction of protein subcellular localization

Author(s)	Algorithm	Feature	# of Classes	Multi-label	Imbalance
Nakashima and Nishikawa (4)	Scoring System	AA[k], PairAA[l]	2	x	x
Cedano *et al*. (13)	LD[a] (Mahalanobis)	AA[k]	5	x	x
Reinhardt and Hubbard (20)	ANN[c] Approach	AA[k]	3, 4	x	x
Chou and Elrod (2)	CD[d]	AA[k]	12	x	x
Yuan (21)	Markov Model	AA[k]	3, 4	x	x
Nakai and Horton (23)	k-NN[e] Approach	Signal Motif	11	x	x
Emanuelsson *et al*. (10)	Neural network	Signal Motif	4	x	x
Drawid *et al*. (27)	CD[d]	Gene Expression Pattern	8	x	x
Drawid and Gerstein (24)	BN[b] Approach	Signal Motif, HDEL motif	5, 6	x	x
Cai *et al*. (8)	SVMs[i]	AA[k]	12	x	x
Chou (6)	Augumented CD[d]	AA[k], SOC[n] factor	5, 7, 12	x	x
Hua and Sun (16)	SVMs[i]	AA[k]	4	x	x
Chou and Cai (25)	SVMs[i]	SBASE-FunD[o]	12	x	x
Nair and Rost (26)	NN[e] Approach	functional annotation	10	x	x
Cai *et al*. (9)	SVMs[i]	SBASE-FunD[o], PseAA[m]	5	x	x
Chou and Cai (11)	NN[e] Approach	GO[p], InterPro-FunD[q], PseAA[m]	3, 4	x	x
Chou and Cai (14)	LD[a]	PseAA[m]	14	x	x
Pan *et al*. (19)	Augmented CD[d]	PseAA[m] with filler	12	x	x
Park and Kanehisa (18)	SVMs[i]	AA[k], PairAA[m], GapAA[k]	12	x	x
Zhou and Doctor (22)	CD[d]	AA[k]	4	x	x
Gardy *et al*. (12)	SVMs[i], HMM[h], BN[b]	AA[k], motif, homology analysis	5	x	x
Huang and Li (17)	fuzzy k-NN[e]	PairAA[l]	4, 11	x	x
Guo *et al*. (15)	p-ANN[j]	AA[k]	8	x	x
Bhasin and Raghava (7)	SVMs[i]	AA[k], PairAA[l]	4	x	x
Chou and Cai (1)	NN[e] Approach	GO[p], InterPro-FunD[q], PseAA[m]	22	Considering	x

[a]LD: Least Distance algorithm.
[b]BN: Bayesian Network.
[c]ANN: Artificial Neural Network.
[d]CD: Covariant Discriminant algorithm.
[e]NN: Nearest Neighbor.
[h]HMM: Hidden Markov Model.
[i]SVMs: Support Vector Machines.
[j]p-ANN: probabilistic Artificial Neural Network.
[k]AA: amino acid composition.
[l]PairAA: amino acid pair composition.
[m]PseAA: pseudo amino acid composition.
[n]SOC: sequence-order correlation.
[o]SBASE-FunD: functional domain composition using SBASE.
[p]GO: gene ontology.
[q]InterPro-FunD: InterPro functional domain composition.
[r]FunDC: functional domain composition. (Here, 'x' means 'Not Considering'.)

highly outperformed the C-SVDD. D-SVDD is one of one-class classification methods whose purpose is to give a compact description of a set of data referred to as target data. One-class classification methods are suitable for imbalanced datasets since find compact descriptions for target data independently from other data (33,36). Moreover, they are easily used for the dataset whose number of classes is big owing to linear complexity with regard to the number of classes. However, original D-SVDD is not for a multi-class and multi-label problem. For the protein localization problem, thus, we propose the PLPD method by extending the original D-SVDD method using the likelihood of a specific protein localization.

The structure of the paper is organized as follows: First, we briefly provide the information on the C-SVDD and the D-SVDD in Section 2. In this section, we also introduce the proposed PLPD method for protein localization prediction. Section 3 highlights the potentials of the proposed approach through experiments with datasets from the work of Huh *et al*. (3). Concluding remarks are presented in Section 4.

MATERIALS AND METHODS

Conventional support vector data description

Since C-SVDD is not well introduced in bioinformatics fileds, we first briefly describe the basic ideas of the C-SVDD. Suppose a dataset containing n target data, $\{\mathbf{x}_i \mid i = 1, \ldots, n\}$, is given in the task of one-class classification. The basic idea of a C-SVDD (36) is to find the hypersphere $(\mathbf{a}, R)$ with minimum volume which includes most of the target data, where $\mathbf{a}$ and R are respectively the center and the radius of the solution (the hypersphere) as shown in Figure 1. To permit $\mathbf{x}_i$ to be located in the outside of a hypersphere, the C-SVDD introduces a slack variable $\xi_i \geqslant 0$ for each target data point analogous to SVMs (38). Thus the solution of the C-SVDD is obtained by minimizing the objective function O:

$$O = R^2 + C^+ \sum_{i=1}^{n} \xi_i \qquad (1)$$

subject to $(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a}) \leqslant R^2 + \xi_i$ where the parameter $C^+ > 0$ gives the trade-off between volume of a

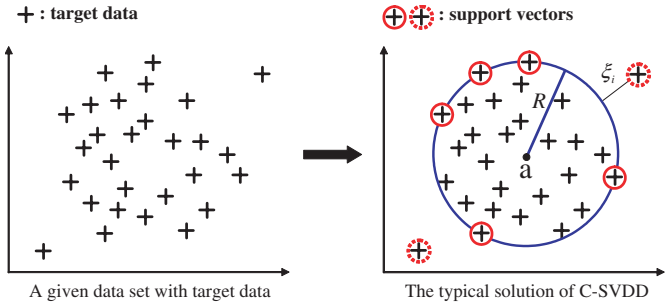

Figure 1. A typical solution of C-SVDD when outliers are permitted. The C-SVDD finds the minimum-volume hypersphere which includes most of target data. The data which resides on the boundary and outside the boundary are called support vectors which fully determine the compact boundary. Thus, the data with solid circle are the support vectors on the boundary, and the data with dotted circle are also support vectors which are the outliers.

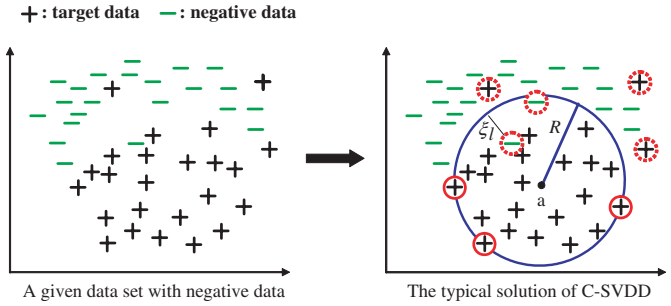

Figure 2. A typical solution of C-SVDD when negative data are available. The C-SVDD finds the minimum-volume hypersphere which includes most of target data and at the same time, excludes most of negative data.

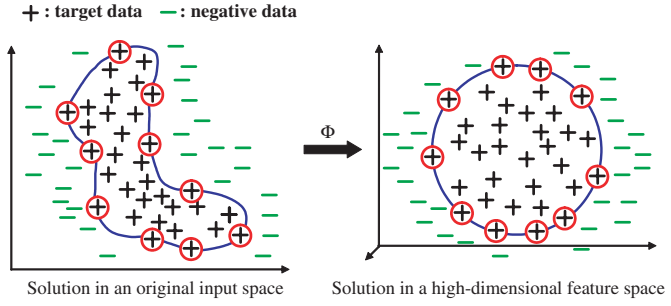

Figure 3. A typical solution of C-SVDD when a kernel function is used. The C-SVDD finds a more flexible solution in a high-dimensional feature space without mapping data into the feature space using some kernel function; C-SVDD finds a flexible solution directly in the original input space with a kernel function as shown in the left figure.

hypersphere and the number of errors (number of target data rejected) (36).

When negative data $\mathbf{x}_l$ which should not be included in a compact description of target data are available during training, C-SVDD utilizes it. In this case, the C-SVDD finds the minimum-volume hypersphere that includes most of target data and at the same time, excludes most of negative data as shown in Figure 2. Suppose we have both n target data and m negative data. (Note: the total number of training data are N where $N = n + m$.) By using another slack variable $\xi_l \geqslant 0$ for the permission of including negative data in a compact description, the objective function of this case is written as:

$$O = R^2 + C^+ \sum_{i=1}^{n} \xi_i + C^- \sum_{l=1}^{m} \xi_l \qquad (2)$$

subject to $(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a}) \leqslant R^2 + \xi_i$ and $(\mathbf{x}_l - \mathbf{a}) \cdot (\mathbf{x}_l - \mathbf{a}) > R^2 - \xi_l$ where C^- is another parameter which controls the trade-off between the volume of data description and the errors of negative data (36).

By using Lagrange Multipliers α_k for $\mathbf{x}_k$, the dual problem of the C-SVDD is reduced to maximizing $D(\alpha)$:

$$D(\alpha) = \sum_{k=1}^{N} y_k \alpha_k \mathbf{x}_k \cdot \mathbf{x}_k - \sum_{p=1}^{N} \sum_{q=1}^{N} y_p y_q \alpha_p \alpha_q \mathbf{x}_p \cdot \mathbf{x}_q \qquad (3)$$

subject to $\sum_{k=1}^{N} y_k \alpha_k = 1$, $0 \leqslant \alpha_i \leqslant C^+$, and $0 \leqslant \alpha_l \leqslant C^-$ where y_k is the label of data $\mathbf{x}_k$. (Note: $y_k = 1$ for a target data point, otherwise $y_k = -1$.) After solving $D(\alpha)$ with regard to α_k, the $\mathbf{a}$ of the optimal hypersphere can be calculated by $\mathbf{a} = \sum_{k=1}^{N} y_k \alpha_k \mathbf{x}_k$, and the radius R can be obtained by the distance between $\mathbf{a}$ and any target data point $\mathbf{x}_i$ that is located on the boundary of the hyperplane.

By comparing the distance between a test data point $\mathbf{x}_t$ and $\mathbf{a}$ with R, the C-SVDD determines the decision whether $\mathbf{x}_t$ is the same data type with the target data or not as:

$$f(\mathbf{x}_t) = I(\mathbf{x}_t \cdot \mathbf{x}_t - 2\mathbf{x}_t \cdot \mathbf{a} + \mathbf{a} \cdot \mathbf{a} \leqslant R^2) \qquad (4)$$

where I is an indicator function (35). Thus, if the distance between $\mathbf{x}_t$ and $\mathbf{a}$ is less than R, then we predict that $\mathbf{x}_t$ is included in the given target dataset; otherwise, we predict that $\mathbf{x}_t$ is not included.

Similar to SVMs, the C-SVDD can find a more flexible data description in a high-dimensional feature space without directly mapping the space using a kernel function $K(\cdot,\cdot)$ (38). As shown in Figure 3, the C-SVDD finds a solution by mapping the input training data into a possible high-dimensional feature space using a mapping function Φ, and seeking a hypersphere in the feature space. However, instead of explicitly mapping the training data into the feature space, C-SVDD directly finds the solution in the input space using the corresponding kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ between any two data $\mathbf{x}_i$ and $\mathbf{x}_j$, defined by:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \qquad (5)$$

Thus, the kernelized form of Equation 3 is:

$$D(\alpha) = \sum_{k=1}^{N} y_k \alpha_k K(\mathbf{x}_k, \mathbf{x}_k) - \sum_{p=1}^{N} \sum_{q=1}^{N} y_p y_q \alpha_p \alpha_q K(\mathbf{x}_p, \mathbf{x}_q). \qquad (6)$$

Using the kernel trick, the C-SVDD directly finds a more flexible boundary in an original input space as shown in the left figure of Figure 3 in an original input space.

Density-induced support vector data description

As mentioned earlier, a C-SVDD finds a compact description that includes most of target data in a high-dimensional feature space, and the data that are not fully included in the

hypersphere are called support vectors (36). In a C-SVDD, these support vectors completely determine the compact description (36) even though most target data can be non-support vectors. This can be a problem in data domain description especially when support vectors do not have the characteristics of a target dataset regarding its density distribution (33).

To address the problems outlined above and to identify the optimal description more easily, Lee *et al*. (33) recently proposed a D-SVDD which is a general extension of the C-SVDD by introducing the notion of a relative density degree for each data point. Using the relative density degree, Lee *et al*. defined a density-induced distance measurement for target data making data with higher density degrees give more influence on finding a compact description. By several experiments, the D-SVDD highly improved the performance of the C-SVDD (33). However, the original work of Lee *et al*. (33) only introduced the mechanism that can incorporate the density degrees of the target data, not negative data. Thus, in this study, we also introduce the mechanism of incorporating the density degrees of negative data by defining a density-induced distance measurement for negative data.

1) *Relative density degree and density induced distance*. to reflect the density distribution into searching the boundary of the C-SVDD, Lee *et al*. first introduced the notion of a relative density degree for each target data point which means the density of the region of the corresponding data point compared to other regions in a given dataset (33). Suppose we calculate a relative density degree for a target data point $\mathbf{x}_i$. By using $d(\mathbf{x}_i, \mathbf{x}_i^K)$, the distance between $\mathbf{x}_i$ and $\mathbf{x}_i^K$ (the Kth nearest neighborhood of $\mathbf{x}_i$), and the mean distance of Kth nearest neighborhoods of all target data, $\Im^K$, they defined the local density degree $\rho_i > 0$ for $\mathbf{x}_i$ as:

$$\rho_i = \exp\left\{\omega \times \frac{\Im^K}{d(\mathbf{x}_i, \mathbf{x}_i^K)}\right\}, \quad i = 1, \ldots, n \tag{7}$$

where $\Im^K = \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{x}_i^K)$, n is the number of data in a target class, and $0 \leqslant \omega \leqslant 1$ is a weighting factor. Note that the measure reports higher local density degree ρ_i for the data in a higher density region: the data with lower $\mathbf{x}_i^K$ have higher ρ_i values. Moreover, a bigger ω produces higher local density degrees. In a similar manner, the relative density degrees for negative data are also calculated.

After calculating the relative density degrees, to incorporate the degrees into searching the optimal description in a C-SVDD, Lee *et al*. (33) proposed a new geometric distance called density-induced distance. They defined a positive density-induced distance δ_i^+ between target data point $\mathbf{x}_i$ and the center of a hyperspherical model ($\mathbf{a}$, R) of a target dataset as:

$$\delta_i^+ \equiv \{\rho_i(\mathbf{x}_i - \mathbf{a})\cdot(\mathbf{x}_i - \mathbf{a})\}^{1/2} \tag{8}$$

where $\mathbf{a}$ and R are the center and the radius of the hypersphere, respectively. The basic idea is simple. To give higher influence on the search of the minimum-sized hypersphere, they made the distance between $\mathbf{x}_i$ and the center $\mathbf{a}$ longer using relative density degre ρ_i. Note that to enclose the data point with increased distance owing to a higher relative density degree ρ_i, the radius of a minimum-sized hypersphere should be increased (a data point with higher ρ_i gives stronger influence on the boundary).

In a similar manner, we can incorporate the relative density degrees for negative data. After calculating the relative density degrees ρ_l for negative data $\mathbf{x}_l$ using Equation 7, we define another density-induced distance δ_l^- between negative data and $\mathbf{a}$, the center of the hyperspherical description of a target data set, as:

$$\delta_l^- \equiv \left\{\frac{1}{\rho_l}(\mathbf{x}_l - \mathbf{a})\cdot(\mathbf{x}_l - \mathbf{a})\right\}^{1/2} \tag{9}$$

where $\mathbf{a}$ and R are the center and the radius of a hypersphere, respectively.

Note that δ_l^- decreases with increasing ρ_l. Hence, to exclude the negative data point with decreased δ_l^- owing to higher relative density degree ρ_l, the radius of a minimum-sized hypersphere should be decreased; the negative data point with higher relative density degree gives higher degree of penalty on the search of the minimum-sized hypersphere for a target dataset.

2) *Mathematical formulation of D-SVDD*. Using the density-induced distance measure δ_i^+ for target data, Lee *et al*. reformulated the C-SVDD when only target data are available (33). Similar to C-SVDD, they permitted the possibility of training error using a slack variable $\zeta_i \geqslant 0$, which is the distance between the boundary Ω and $\mathbf{x}_i$ outside Ω. Here, ζ_i equals $(\delta_i^+)^2 - R^2$ for training error data, otherwise it is 0. It implies that ζ_i contains the information of a relative density degree of $\mathbf{x}_i$.

Using the slack variable for each target data point, they deduce Equation 10 from Equation 1:

$$O = R^2 + C^+ \sum_{i=1}^{n} \zeta_i \tag{10}$$

subject to $\rho_i(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a}) \leqslant R^2 + \zeta_i$.

By introducing Lagrange Multipliers, Lee *et al*. (33) could construct the dual problem: maximize $D(\alpha)$,

$$D(\alpha) = \sum_{i=1}^{n} \alpha_i \rho_i \mathbf{x}_i \cdot \mathbf{x}_i - \frac{1}{T}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \rho_i \rho_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{11}$$

subject to $\sum_{i=1}^{n} \alpha_i = 1$, $0 \leqslant \alpha_i \leqslant C^+$, and $T = \sum_{i=1}^{n} \alpha_i \rho_i$.

After deriving α_k that satisfy Equation 11, $\mathbf{a}$ of the optimal description can be calculated by:

$$\mathbf{a} = \frac{1}{T}\sum_{i=1}^{n} \alpha_i \rho_i \mathbf{x}_i, \; T = \sum_{i=1}^{n} \alpha_i \rho_i. \tag{12}$$

Different from the C-SVDD (36), the center $\mathbf{a}$ of the optimal hypersphere is weighted by the relative density degree ρ_i (Equation 12) where the center is shifted to a higher dense region. Moreover, the R of optimal description is calculated by the δ_i^+ distance between $\mathbf{a}$ and any $\mathbf{X}_i$ of which $0 < \alpha_k < C^+$ (33).

When negative data are available, D-SVDD can also utilize them to improve the description of the target dataset. In this case, we use negative density-induced distance δ_l^- for negative data in order to incorporating the density distribution of negative data. Using the negative density-induced distance

and another slack variable ζ_l for possibility of training error for each negative data point, we find the optimal hypersphere that includes most target data and excludes most negative data. Here, ζ_l for each negative data point is the distance between the boundary Ω and $\mathbf{x}_l$ inside Ω; that is, $\zeta_l = R^2 - (\delta_l^-)^2$, $(\zeta_l \geqslant 0)$. Thus the new objective function is defined as:

$$O = R^2 + C^+ \sum_{i=1}^{n} \zeta_i + C^- \sum_{l=1}^{m} \zeta_l \quad (13)$$

subject to $\rho_i(\mathbf{x}_i - \mathbf{a})\cdot(\mathbf{x}_i - \mathbf{a}) \leqslant R^2 + \zeta_i$ and $\frac{1}{\rho_l}(\mathbf{x}_l - \mathbf{a})\cdot(\mathbf{x}_l - \mathbf{a}) > R^2 - \zeta_l$ where $C^- > 0$ is a similar control parameter with that of the C-SVDD (34).

Similar to the previous case, after constructing the Lagrangian of Equation 13, the dual representation of this case is obtained as:

$$D(\alpha) = \sum_{k=1}^{N} y_k \alpha_k \rho'_k \mathbf{x}_k \cdot \mathbf{x}_k - \frac{1}{T} \sum_{p=1}^{N} \sum_{q=1}^{N} y_p y_q \alpha_p \alpha_q \rho'_p \rho'_q \mathbf{x}_p \cdot \mathbf{x}_q \quad (14)$$

subject to $\sum_{k=1}^{N} y_k \alpha_k = 1$, $0 \leqslant \alpha_i \leqslant C^+, 0 \leqslant \alpha_l \leqslant C^-$, and $T = \sum_{k=1}^{N} y_k \alpha_k \rho_k$ where y_k is the label of $\mathbf{x}_k$ ($y_k = 1$ for a target data point, otherwise $y_k = -1$), and N is the total number of data ($N = n + m$). Here, $\rho'_k = \rho_i$ for target data $\mathbf{x}_i$ and $\rho'_k = 1/\rho_l$ for negative data $\mathbf{x}_l$.

Note that using $\alpha'_k = y_k \alpha_k$ and ρ'_k (instead of ρ_k) the dual representation of negative version of D-SVDD becomes identical to the previous case of D-SVDD (Equation 11) except for the number of data considered (n becomes $N = n + m$). Therefore, when negative data are available, we can just use α'_k and ρ'_k in the optimization problem and in the decision function of the precious case. That means there are no extra computational complications except the complication caused by the increased data size.

As seen in Equations 11 and 14, the dual forms of the objective functions of D-SVDD are represented entirely in terms of inner products of input vector pairs. Thus, we can kernelize D-SVDD where the kernelized version of the dual representation of the objective function when negative data are available is:

$$D(\alpha) = \sum_{k=1}^{N} \alpha_k \rho'_k K(\mathbf{x}_k, \mathbf{x}_k) - \frac{1}{T} \sum_{p=1}^{N} \sum_{q=1}^{N} \alpha'_p \alpha'_q \rho'_p \rho'_q K(\mathbf{x}_p, \mathbf{x}_q) \quad (15)$$

where T and constraints are the same as in Equation 14.

PLPD for prediction of protein subcellular localization

As mentioned earlier, the prediction of protein localization is a kind of multi-class and multi-label classification problem. However, D-SVDD including conventional SVDD is not for the multi-class and multi-label classification problems; it is for one-class and mono-label classification problems. Thus, we propose a new method for the prediction of protein localization by modifying the D-SVDD.

For a multi-class and multi-label classification problem, a predictor should answer two points. First, for a multi-class problem, a predictor should report the degree of being a member of each class for a test data point $\mathbf{x}_t$ using some score function $f(\mathbf{x}_t, \cdot)$ where f: $\aleph \times Ł \rightarrow \Re$. (Here, $\aleph$ is the domain of data, Ł is the domain of class labels, and $\Re$ is the domain of real numbers.) That is, a label l_1 is considered to be ranked higher than l_2 if $f(\mathbf{x}_t, l_1) > f(\mathbf{x}_t, l_2)$. Second, owing to multi-label cases, the score function should be able to report multiple labels not a mono label, and the true positive labels of a protein should rank higher than any other labels. That is, if $Ł_p$ is a set of the true positive labels of $\mathbf{x}_t$, then a successful learning system will tend to rank the labels in the $Ł_p$ higher than any other labels not in $Ł_p$.

To address the requirements mentioned above, we adopt the following procedure.

(i) if a training dataset is given, we divide it into a target dataset and a negative dataset by class. For a label l_i, for instance, a data point whose label set has l_i is included in the target data; otherwise, it is included in the negative dataset.
(ii) if a target dataset and a negative dataset are prepared for each class, we find the optimal boundary for the target data by using D-SVDD as formulated in Equation 15.
(iii) we calculate the degree of being a member of each class l_i for a test data point $\mathbf{x}_t$ using the following score function f:

$$f(\mathbf{x}_t, l_i) = \frac{R_i}{d(\mathbf{x}_t, \mathbf{a}_i)} \quad (16)$$

where $(\mathbf{a}_i, R_i)$ is the optimal hypersphere of target data which are included in the class label l_i. Note that this score function reports a higher value for a test data point with smaller distance between $\mathbf{x}_t$ and $\mathbf{a}_i$ regarding to the distance of R_i. If the distance between $\mathbf{x}_t$ and $\mathbf{a}_i$, for example, is smaller than R_i, the score function reports higher values than 1; otherwise, the score function reports smaller values than 1. That means if the location of x_t is closer to the center of the hypershpere, then the score function reports a higher degree of value.
(iv) Finally, according to the values of the score function for all classes, we rank the labels, and report them.

With this procedure, we can easily and intuitively modify the D-SVDD for a multi-class and multi-label classification problem like the prediction of protein localization by reflecting the overall characteristics of each class. We call the proposed method a PLPD.

RESULTS AND DISCUSSION

Data preparation and evaluation measures

1) *Data preparation*. To test the effectiveness of the proposed PLPD method, we experimented with the data of Huh *et al*. (3), which is currently the biggest class coverage and is a multi-label dataset. From the website http://yeastgfp.ucsf.edu, we get 3914 unique proteins whose locations are clearly identified. The breakdown of the original 3914 different proteins ($N = 3914$) is given in the second column of Table 1. Owing to some proteins may coexist several localizations, the so-called multi-label feature as mentioned earlier, the total sum of proteins in all localizations, denoted by $\tilde{N}$, is 5184; In $N = 3914$ total different proteins, there are 2725 proteins in unique localization, 1117 proteins in two different

localizations, 64 proteins in three different localizations, 7 proteins in four different localizations. And there is only 1 protein in five different localizations. Similar to the formulation by Chou and Cai (1),

$$\tilde{N} = 1 \times 2725 + 2 \times 1117 + 3 \times 64 + 4 \times 7 + 5 \times 1 = 5184. \quad (17)$$

Using the 3914 proteins, we represent a protein in three different ways, and make three datasets: Dataset-I, Dataset-II and Dataset-III to evaluate the performance of the proposed PLPD in various manners (see Table 1). In Dataset-I, we used AA-based features for a protein. After finding all amino acid sequences of 3914 proteins from the SWISS-PROT DataBank (30), we used PairAA features (400D; features from a_{21} to a_{420}), and GapAA up to the maximal allowed gap (9200D = 400D × 23; the minimum length among all 3914 proteins is 25; features from a_{421} to a_{9620}) for sequence information including the amino acid composition (20 Dimensions; features from a_1 to a_{20}). That is, a protein P is represented as a vector in a 9620-dimensional space (9620D = 20D + 400D + 9200D) as:

$$P = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_{9620} \end{bmatrix} \quad (18)$$

This dataset is described in the third column of Table 1. As you can see in the column, the coverage of this representation is 100%; that means, all of the 3914 proteins can be represented by this manner.

In Dataset-II, we adopted a similar manner with the Chou and Cai's approach (5) using the InterPro Motifs (29). Different from Chou and Cai, we first extracted 2372 unique motifs which are occurred only in the 3914 proteins in order to delete meaningless motifs. Using the unique motif set, we represent a protein as:

$$P = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_{2372} \end{bmatrix}, \quad (19)$$

where $b_i = 1$ if the protein has the ith motif in the unique motif set; otherwise, $b_i = 0$. The coverage of each localization is depicted in the fourth column is Table 1. With this second manner, the overall coverage is 77.08% (3017 proteins among 3914 proteins).

For Dataset-III, we combined the previous two features only for the proteins which have more than one motif in the unique motif set. In this case, thus, a protein is characterized as:

$$P = \begin{bmatrix} c_1 \\ c_2 \\ \cdots \\ c_{11992} \end{bmatrix}, \quad (20)$$

where the elements from c_1 to c_{9620} are derived from Equation 18, and the elements from c_{9621} to c_{11992} are derived from Equation 19. The coverage is same with the Dataset-II (see the fifth column is Table 1).

2) *Evaluation measures*. For multi-label learning paradigms, only one measurement is not sufficient to evaluate the performance of a predictor owing to the variety of correctness in prediction (39,40). Thus, we introduce three measurements (Measure-I, Measure-II and Measure-III) for the evaluation of a protein localization predictor. First, to check the overall success rate regarding to the total number of the unique proteins N, we define Measure-I as:

$$\frac{1}{N} \sum_{i=1}^{N} \psi\,[L(P_i), Y_i^k], \quad (21)$$

where $L(P_i)$ is the true label set of a protein P_i, Y_i^k is the predicted top-k labels by a predictor, and

$$\psi\,[L(P_i), Y_i^k] = \begin{cases} 1, & \text{if any label in } Y_i^k \text{ is in } L(P_i), \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

Note that owing to the multi-label protein localization problem, it is not sufficient to evaluate the performance of a predictor by checking only the topmost label predicted true. Thus, we check the real label set with the predicted top-k labels using the $\psi[\cdot, \cdot]$ function. The k value in Equation 22 is given by user, and we use 3 in this study since the numbers of true localization sites of most proteins are less than or equal to 3 (3).

Similar to the formulation by Chou and Cai (1), to check the overall success rate regarding to the total number of classified proteins, $\tilde{N}$, we also evaluate the performance of a predict by using Measure-II:

$$\frac{1}{\tilde{N}} \sum_{i=1}^{N} \Psi\,[L(P_i), Y_i^{k_i}], \quad (23)$$

where $Y_i^{k_i}$ is the predicted top-k_i labels by a predictor, and the $\Psi[\cdot, \cdot]$ function returns the number of labels which is predicted correctly. Note that the k_i value determined by the number of true labels of a protein P_i, not by user.

As mentioned earlier, a dataset of protein localization is imbalanced in nature. Including overall success rate, thus, the information on the success rate of each class and the average rate of the success rates of each class is useful to evaluate the performance of a predictor. To achieve this, we define Measure-III as:

$$\frac{1}{\mu} \sum_{l=1}^{\mu} \left(\frac{1}{\tilde{n}_l} \sum_{i=1}^{\tilde{n}_l} \Delta[Y_i^{k_i}, l] \right) \quad (24)$$

where $\mu = 22$ is total number of labels or classes, l is a label index, $\tilde{n}_l$ is the number of proteins in the lth label, and

$$\Delta\,[Y_i^{k_i}, l] = \begin{cases} 1, & \text{if any label in } Y_i^{k_i} \text{ is equal to } l, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

Performance comparison

To investigate the success of the proposed PLPD method and analyze it, we conducted several tests with the three datasets

Table 3. Prediction performance (%) of ISort and PLPD to the Dataset-I

	Measure	ISort method (%)	PLPD method (%)
Measure-I		65.14	73.89
Measure-II		35.91	53.09
	1. Actin	0.00	0.00
	2. Bud	0.00	0.00
	3. Bud neck	0.00	0.00
	4. Cell periphery	0.00	0.00
	5. Cytoplasm	77.55	99.89
	6. Early golgi	5.56	5.56
	7. Endosome	6.52	6.52
	8. ER	0.00	0.00
	9. ER to golgi	16.67	16.67
	10. Golgi	0.00	4.88
	11. Late golgi	2.27	6.82
Measure-III	12. Lipid particle	8.70	21.74
	13. Microtubule	10.00	10.00
	14. Mitochondrion	0.77	1.15
	15. Nuclear periphery	0.00	0.00
	16. Nucleolus	6.10	1.83
	17. Nucleus	83.82	65.08
	18. Peroxisome	0.00	9.52
	19. Punctate composite	0.73	0.00
	20. Spindle pole	0.00	0.00
	21. Vacuolar membrane	1.72	1.72
	22. Vacuole	0.00	0.00
	Average	10.02	11.43

Table 4. Prediction performance (%) of ISort and PLPD to the Dataset-II

	Measure	ISort method (%)	PLPD method (%)
Measure-I		69.94	82.40
Measure-II		44.27	56.32
	1. Actin	0.00	22.22
	2. Bud	5.26	42.11
	3. Bud neck	10.42	31.25
	4. Cell periphery	41.84	26.53
	5. Cytoplasm	71.13	84.58
	6. Early golgi	7.69	25.64
	7. Endosome	10.81	21.62
	8. ER	16.43	12.56
	9. ER to golgi	0.00	60.00
	10. Golgi	6.67	33.33
	11. Late golgi	5.26	21.05
Measure-III	12. Lipid particle	0.00	46.67
	13. Microtubule	29.41	41.18
	14. Mitochondrion	18.77	16.20
	15. Nuclear periphery	0.00	13.16
	16. Nucleolus	17.21	26.23
	17. Nucleus	44.32	65.36
	18. Peroxisome	0.00	50.00
	19. Punctate composite	6.59	7.69
	20. Spindle pole	14.81	33.33
	21. Vacuolar membrane	0.00	10.64
	22. Vacuole	30.65	21.77
	Average	15.33	32.41

(Dataset-I, Dataset-II and Dataset-III). For comparative analysis we also experimented with the ISort method (1). To the best of our knowledge, ISort method showed the best performance for the prediction of yeast protein multiple localization up to now (1). Even though the jackknife cross-validation is one of most rigorous and objective validation measures (41,42), we did a 2-fold cross-validation approach to prevent the overfitting problem to a given training dataset. For a more flexible boundary of the PLPD, we used a Gaussian RBF function for kernelization in Equation 15 owing to the Gaussian RBF function is one of most suitable functions for kernelization (36,37). In addition, we selected the model parameters, such as C^+ and C^-, and the width parameter of the Gaussian RBF kernel function by using cross-validation approach (38,43) to identify the solutions of the PLPD.

The results of the ISort Method and the PLPD method for the three datasets are given in Table 3–5. As we can see in Table 3, ISort method showed 65.14, 35.91 and 10.02% according to Measure-I, Measure-II and Measure-III for the Dataset-I. For the same dataset, the PLPD method showed 73.89, 53.09 and 11.43% performance for the three measurements, respectively. This implies that the success rates of PLPD were 8.75, 17.18 and 1.41% higher than the ISort method regarding the Measure-I, the Measure-II and the Measure-III, respectively. Even though PLPD method showed better scores than the ISort method with regard to Measure-III, the two methods showed low degrees of average prediction accuracies for all localizations. For instance, ISort showed zero prediction accuracies at 10 localizations and PLPD at 9 localizations. As you can see in Table 1, those localizations whose prediction accuracies were zero have relatively small number of proteins in themselves. Thus, the prediction of localization based on only AA-based features has limitation to correctly predict all the localizations on average.

Table 5. Prediction performance (%) of ISort and PLPD to the Dataset-III

	Measure	ISort method (%)	PLPD method (%)
Measure-I		75.90	83.49
Measure-II		49.16	57.24
	1. Actin	3.70	18.52
	2. Bud	5.26	57.89
	3. Bud neck	4.17	33.33
	4. Cell periphery	30.61	33.67
	5. Cytoplasm	73.30	77.04
	6. Early golgi	12.82	25.64
	7. Endosome	24.32	35.14
	8. ER	22.71	21.26
	9. ER to golgi	0.00	60.00
	10. Golgi	13.33	43.33
	11. Late golgi	10.53	26.32
Measure-III	12. Lipid particle	0.00	53.33
	13. Microtubule	29.41	52.94
	14. Mitochondrion	27.51	33.16
	15. Nuclear periphery	15.79	23.68
	16. Nucleolus	28.69	31.97
	17. Nucleus	51.07	66.96
	18. Peroxisome	6.25	68.75
	19. Punctate composite	10.99	12.09
	20. Spindle pole	29.63	40.74
	21. Vacuolar membrane	4.26	14.89
	22. Vacuole	41.13	22.58
	Average	20.25	38.78

When the Dataset-II was used, the prediction accuracies of each method were higher than those of the two methods with Dataset-I. As you can see in Table 4, the ISort method showed 69.94, 44.27 and 15.33% accuracies for three measurements. On the contrary, the PLPD method showed 82.40, 56.32 and 32.41% accuracies for the three measurements. These improvements of the PLPD over the ISort were more conspicuous. Moreover, there was no zero performance for each localization in the PLPD method. In ISort

Table 6. The performance (%) of the proposed PLPD to the Dataset-I, Dataset-II, and Dataset-III only with regard to the Measure-III

Measure	Dataset-I	Dataset-II	Dataset-III
Measure-III Average	19.10%	44.61%	46.50%

method, however, there were 6 localizations(Actin, ER to Golgi, Lipid particle, Nuclear periphery, Peroxisome and Vacuolar membrane) whose prediction accuracies were zero. This means that the proposed method showed promising results even though a dataset is imbalanced when Dataset-II was used.

The best performance of each method was achieved when Dataset-III was used. Similarly, the PLPD method outperformed all the three measurements considered as shown in Table 5. The PLPD, for example, showed up to 83.49% accuracy with Measure-I, while the ISort method showed 75.90% accuracy. In regard to Measure-III, the PLPD method showed 38.78% average accuracy for all 22 localizations; it was 18.53% higher than the ISort method on average of all prediction accuracies for each localization.

In order to check the performance of PLPD with regard to Measure-III without considering the other two measurements, we did tests for the same data sets with similar manner to the previous tests. This means that we did parameter fitting process only regarding Measure-III for PLPD method. The results for Dataset-I, Dataset-II, and Dataset-III are depicted in Table 6. As you can see in Table 6, the average accuracies were highly increased. When the Dataset-III was used, for example, the average accuracy for all localization was up to 46.50%; it was 7.72% higher than the previous result of PLPD and 26.25% higher than the result of ISort method for the same dataset. It was remarkable.

From Tables 3–6, we could conclude that the proposed PLPD method outperformed the ISort method regardless of the kind of evaluation measurement and regardless of the dataset used. Moreover, Motif information could increase the prediction accuracies of the two methods considered even though its coverage is lower than AA-based information. Furthermore, the best performance was obtained when both the features were used.

Similar to the work of Cai and Chou (44), to avoid homology bias, we removed all the sequences with >40% sequence homology and after then, we performed similar experiments to the previous cases. The new datasets are depicted in Table 1 in the Supplementary Data and the results of this experiments are described in Tables 2–4 in Supplementary Data. As we can see in the tables, we observed similar phenomena with previous results that were depicted in Tables 3–5, which means that the PLPD method can play a complimentary role to existing methods, regardless of the existence of sequence homology.

Using the 5184 classified proteins as training data we predicted 138 proteins whose subcellular localizations could not be clearly observed by the experiments of Huh *et al*. (3). Since the prediction accuracies were highest when both AA-based features and the unique InterPro Motif set were used as features of a protein, we used the information together. Actually we could not know the number of true localizations of a protein whose localization is not known yet, we enlist top three localizations which have the highest likelihood to be the true localization. In some cases, proteins have similar degrees in the likelihood. Thus, to treat this issue, we quantize the degrees of likelihood and treat them in the same rank. The predicted results of first 35 unknown localization proteins are given in the Table 7, where the roman numerals indicate the rank of likelihood of the corresponding localization. (See Supplementary Data for all results.)

Among the predicted protein localizations, YBL105C, YBR072W and YDR313C are analyzed for validation as an example. As shown in Table 7, YBL105C is predicted to localize to 'cytoplasm' as the first rank, to 'bud neck' and 'cell periphery' as the second ranks. YBL105C (PKC1) is a Ser/Thr kinase and controls signaling pathway for cell wall integrity, for instance bud emergence and cell wall remodelling during growth (45,46). PKC1 contains C1, C2 and HR1 domains. HR1 domain targets PKC1 to bud tip and C1 domain targets it to cell periphery (47). Thus, PKC1 localizes to the 'bud neck' for its function. Those indicate that the localizations of YBL105C found from literature are consistent with our prediction results; 'bud neck' and 'cell periphery'.

YBR072W (HSP26) is a heat shock protein which transforms into high molecular weight aggregate on heat shock and binds to non-native proteins (48,49). Depending on cellular physiological conditions, HSP26 localizes differently; accumulation in the 'nucleus' or wide-spread throughout the cell (48). The localizations of YBR072W found from literature, 'nucleus' and 'cytoplasm', are predicted correctly by our method.

Finally, YDR313C (PIB1) is a RING-type ubiquitin ligase containing FYVE finger domain. PIB1 binds specifically to phosphatidylinositol(3)-phosphate. Phosphatidylinositol(3)-phosphate is a product of phosphoinositide 3-kinase which is an important regulator of signaling cascade and intracellular membrane trafficking (50). The FYVE domain targets PIB1 to 'endosome' and 'vacuolar membrane' (51). The localizations of YDR313C found from literature, 'endosome' and 'vacuolar membrane', are consistent with our prediction results.

CONCLUSION

Subcellular localization is one of the most basic functional characteristics of a protein, and an automatic and efficient prediction method for the localization is highly required owing to the need for large-scale genome analysis. Even though many previous works have been done for the task of protein subcellular localization prediction, none of them tackles effectively all the characteristics of the task: a multiple class problem (there are too many localizations), a multi-label classification problem (a protein may have several different localizations), and an imbalanced dataset (a protein dataset is too imbalanced in nature). To get more reliable results, thus, a new computational method is eventually needed.

In this paper, we developed a PLPD method for the prediction of protein localization, which can find the likelihood of the specific localization for a protein more easily and more

Table 7. The first 35 prediction results of proteins whose localizations are not clearly observed by the experiments.

Protein	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
YAL029C			I		II												III					
YAL053W			III	I	I	III		II						II			II		II			II
YAR019C			III		I								II									
YAR027W					II			I						II			II		III			II
YAR028W					II			I						II			II		III			I
YBL034C					I	III									II	III	II		III			
YBL067C					I									III		III	I		II			
YBL105C		III	II	II	I												III					
YBR007C					I									II			II		III			
YBR072W				III	I									II			I		III			
YBR168W				III	I									III			II					
YBR200W			II	III	I												II					
YBR235W				I	III												III					II
YBR260C			I	II	II												III					
YCL024W		III	II	III	I																	
YCR021C		I		II				III						III			III					
YCR023C				I		III	III				III										II	
YCR037C					I			III						III			II		III		III	III
YDL025C	II		I	III	I																III	
YDL171C			III	II	I									II		III	I		III		III	
YDL203C					I	III								II			II		III			
YDL238C					I			III						II			I		III			
YDL248W					II			I						II			II		III			I
YDR069C					I									III			II		III			
YDR072C					III	I		II		II							III				I	
YDR089W					I			I						III			I		II		II	II
YDR093W					II	I		II			III			III					III			
YDR164C					II	I				II	II			III			II				I	
YDR181C					I									II			I		III			
YDR182W					I									III			II		I			I
YDR251W					II			III						I			II		III			III
YDR261C					I	III		III						III		III	II					II
YDR276C				III	II			II						II		III	II		I		I	II
YDR309C					I			III						I			I		II			III
YDR313C				III	II		II										III				I	

The numbers in the first row indicate the specific localizations listed in Table 2.

correctly. PLPD method is developed by using the Density-induced Support Vector Data Description (D-SVDD) (33). D-SVDD is one of one-class classification methods which is suitable for imbalanced datasets since they finds a compact description of a target data independently from other data (33,36). Moreover, it is easily used for the dataset whose number of classes is big owing to linear complexity with regard to the number of classes. However, D-SVDD is originally not for a multi-class and multi-label problem. Thus, we extended the D-SVDD for the prediction of protein subcellular localization. Moreover, we have introduced three measurements (Measure-I, Measure-II and Measure-III) for the evaluation of a protein localization predictor to more precisely evaluate the predictor. As the results of three datasets which are made by the experimental results of Huh *et al*. (3), the proposed PLPD method represents a different approach that might play a complimentary role to the existing methods, such as Nearest Neighbor method and discriminate covariant method. Finally, after finding the good boundary of each localization using the 5184 classified proteins as training data, we predicted 138 proteins whose subcellular localizations could not be clearly observed by the experiments of Huh *et al*. (3).

For the reliable prediction of subcellular localizations of proteins, both good features for a protein and a good computational algorithm are ultimately needed. Actually, this study mainly focused on a good computational algorithm for protein localization prediction. In this paper, we represented proteins in three different ways: AA-based features in Dataset-I, Motif-based features in Dataset-II and AA-and-Motif-based features in Dataset-III. From this study we observed that Motif-based features are more informative than AA-based features in protein localization prediction even though Motif-based features have lower coverage than AA-based features. Moreover, best performance was achieved when both AA-based features and Motif-based features are used simultaneously. In the current study we achieved relatively high performance in the protein localization prediction problem. However, it is not sufficient yet and for better results, further study should focus on a mechanism that can extract better information from given proteins.

SUPPLEMENTARY DATA

Supplementary data are available at NAR online.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their helpful comments. This work was supported by National Research Laboratory Grant (2005–01450) and the Korean Systems

Biology Research Grant (2005–00343) from the Ministry of Science and Technology. The authors would like to thank CHUNG Moon Soul Center for supporting BioInformation and BioElectronics and the IBM SUR program for providing research and computing facilities. Funding to pay the Open Access publication charges for this article was provided by Korean Science and Engineering Foundation (KOSEF).

Conflict of interest statement. None declared.

REFERENCES

1. Chou,K.C. and Cai,Y.D. (2005) Predicting protein localizaiton in budding yeast. *Bioinformatics*, **21**, 944–950.
2. Chou,K.C. and Elrod,D.W. (1999) Protein subcellular location prediction. *Protein Eng*., **12**, 107–118.
3. Huh,W.K., Falvo,J.V., Gerke,L.C., Carroll,A.S., Howson,R.W., Weissman,J.S. and O'Shea,E.K. (2003) Global analysis of protein localization in budding yeast. *Nature*, **425**, 686–691.
4. Nakashima,H. and Nishikawa,K. (1994) Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *J. Mol. Biol*., **238**, 54–61.
5. Chou,K.C. (2001) Prediction of protein cellular attributes using pseudo-amino-acid-composition. *Proteins*, **43**, 246–255.
6. Chou,K.C. (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochem. Biophys. Res. Commun*., **278**, 477–483.
7. Bhasin,M. and Raghava,G.P. (2004) ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Res*., **32**, 414–419.
8. Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2000) Support vector machines for prediction of protein subcellular location. *Mol. Cell. Biol. Res. Commun*., **4**, 230–233.
9. Cai,Y.D., Zhou,G.P. and Chou,K.C. (2003) Support vector machines for predicting membrane protein types by using functional domain compostion. *Biophys. J*., **84**, 3257–3263.
10. Emanuelsson,O., Nielsen,H., Brunak,S. and von Heijne,G. (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol*., **300**, 1005–1016.
11. Chou,K.C. and Cai,Y.D. (2003) A new hybrid approach to predict subcellular localization of proteins by incorporating Gene ontology. *Biochem. Biophys. Res. Commun*., **311**, 743–747.
12. Gardy,J.L., Spencer,C., Wang,K., Ester,M., Tusnády,G.E., Simon,I., Hua,S., deFays,K., Lambert,C., Nakai,K. *et al*. (2003) PSORT-B: improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res*., **31**, 3613–3617.
13. Cedano,J., Aloy,P., P'erez-Pons,J.A. and Querol,E. (1997) Relation between amino acid composition and cellular location of proteins. *J. Mol. Biol*., **266**, 594–600.
14. Chou,K.C. and Cai,Y.D. (2003) Prediction and classificaiton of protein subcellular location: sequence-order effect and pseudo amino acid composition. *J. Cell. Biochem*., **90**, 1250–1260.
15. Guo,J., Lin,Y. and Sun,Z. (2004) A novel method for protein subcellular localization based on boosting and probabilistic neural network. *Proceedings of the second conference on Asia-Pacific bioinformatics*, **29**, 21–27.
16. Hua,S. and Sun,Z. (2001) Support vector mahcine approach for protrein subcellularlocalization prediction. *Bioinformatics*, **17**, 721–728.
17. Huang,Y. and Li,Y. (2004) Prediciton of protein subcellular locations using fuzzy k-NN method. *Bioinformatics*, **20**, 21–28.
18. Park,K.J. and Kanehisa,M. (2003) Prediction of protein subcellular locations by support vector machines suing compositions of amino acid and amino acid paris. *Bioinformatics*, **19**, 1656–1663.
19. Pan,Y.X., Shang,Z.Z., Guo,Z.M., Feng,G.Y., Huang,Z.D. and He,L. (2003) Application of pseudo amino acid composition for predicting protein subcellular location: stochastic signal processing approach. *J. Protein Chem*., **22**, 395–402.
20. Reinhardt,A. and Hubbard,T. (1998) Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res*., **26**, 2230–2236.
21. Yuan,Z. (1999) Prediction of protein subcellular locations using Markov chain models. *FEBS Lett*., **451**, 23–26.
22. Zhou,G.P. and Doctor,K. (2003) Subcellular location prediction of apoptosis proteins. *Proteins*, **50**, 44–48.
23. Nakai,K. and Horton,P. (1999) PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem Sci*., **24**, 34–35.
24. Drawid,A. and Gerstein,M. (2000) A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome. *J. Mol. Biol*., **301**, 1059–1075.
25. Chou,K.C. and Cai,Y.D. (2002) Using funcitonal domain composition and support vector machines for prediction of protein subcellular location. *J. Biol. Chem*., **277**, 45765–45769.
26. Nair,R. and Rost,B. (2002) Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, **18**, S78–S86.
27. Drawid,A., Jansen,R. and Gerstein,M. (2000) Genome-wide analysis relating expression level with protein subcellular localization. *Trends Genet*., **16**, 426–430.
28. Murvai,J., Vlahovicek,K., Barta,E. and Pongor,S. (2001) The SBASE protein domain library, release 8.0: a collection of annotated protein sequence segments. *Nucleic Acids Res*., **29**, 58–60.
29. Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R. *et al*. (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res*., **29**, 37–40.
30. Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*., **28**, 45–48.
31. Wang,M., Yang,J., Xu,Z.J. and Chou,K.C. (2005) SLLE for predicting membrane protein types. *J. Theor. Biol*., **232**, 7–15.
32. Chou,K.C. (1995) A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space. *Proteins*, **21**, 319–344.
33. Lee,K., Kim,D.-W., Lee,D. and Lee,K.H. (2005) Imporoving Support Vector data description using local density degree. *Pattern Recognition*, **38**, 1768–1771.
34. Tax,D.M.J. and Duin,R.P.W. (1999) Support vector domain description. *Pattern Recognition Lett*., **20**, 1191–1199.
35. Tax,D.M.J. (2001) One-class classification: Concept-learning in the absence of counter-examples. PhD Thesis, Delft University of Technology, June 2001, ISBN: 90-75691-05-x.
36. Tax,D.M.J. and Duin,R.P.W. (2004) Support Vector Data Description. *Machine Learning*, **54**, 45–66.
37. Vapnik,V. (1998) Statistical Learning Theory: Section II Support Vector Estimation of Functions. Wiley, NY, pp. 375–567.
38. Schölkopf,B. and Smola,A.J. (2002) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, London.
39. Thabtah,F.A., Cowling,P. and Peng,Y. (2004) MMAC: a new multi-class, multi-label associative classification approach. *Fourth IEEE Int'l Conf. on Data Mining*, **4**, 217–224.
40. Zhang,M.-L. and Zhou,Z.-H. (2005) A k-nearest neighbor based algorithm for multi-label classification. *First Int'l Conf. on Granular Computing*, **1**, 718–721.
41. Mardia,K.V., Kent,J.T. and Bibby,J.M. (1979) Multivariate Analysis: Chapter 11 Discriminatnt Analysis; Chapter 12 Multivariate Analysis of Variance; Chapter 13 Cluster Analysis. Academic Press, London, pp. 300–386.
42. Chou,K.C. and Zhang,C.T. (1995) Prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol*., **30**, 275–349.
43. Chapelle,O. and Vapnik,V. (2000) Model selection for Support Vector Machines. *Advances in Neural Information Processing Systems 12*. MIT Press, MA.
44. Cai,Y.D. and Chou,K.C. (2004) Predicting 22 protein localizaitons in budding yeast. *Biochem. Biophys. Res. Comm*., **323**, 425–428.
45. Gray,J.V., Ogas,J.P., Kamada,Y., Stone,M., Levin,D.E. and Herskowitz,I. (1997) A role for the Pkc1 MAP kinase pathway of *Saccharomyces cerevisiae* in bud emergence and identification of a putative upstream regulator. *EMBO J*., **16**, 4924–4937.
46. Sussman,A., Huss,K., Chio,L.C., Heidler,S., Shaw,M., Ma,D., Zhu,G., Campbell,R.M., Park,T.S., Kulanthaivel,P. *et al*. (2004) Discovery of

Cercosporamide, a known antifungal natural product, as a selective Pkc1 kinase inhibitor through high-throughput screening. *Eukaryotic Cell*, **3**, 932–943.
47. Denis,V. and Cyert,M.S. (2005) Molecular analysis reveals localization of *Saccharomyces cerevisiae* protein kinase C to sites of polarized growth and Pkc1p targeting to the nucleus and mitotic spindle. *Eukaryotic Cell*, **4**, 36–45.
48. Rossi,J.M. and Lindquist,S. (1989) The intracellular location of yeast heat-shock protein 26 varies with metabolism. *J. Cell Biol.*, **108**, 425–439.
49. Stromer,T., Fischer,E., Richter,K., Haslbeck,M. and Buchner,J. (2004) Analysis of the regulation of the molecular chaperone Hsp26 by temperature-induced dissociation: the N-terminal domain is important for oligomer assembly and the binding of unfolding proteins. *J. Biol. Chem.*, **279**, 11222–11228.
50. Burd,C.G. and Emr,S.D. (1998) Phosphatidylinositol(3)-phosphate signaling mediated by specific binding to RING FYVE domains. *Cell*, **2**, 157–162.
51. Shin,M.E., Ogburn,K.D., Varban,O.A., Gilbert,P.M. and Burd,C.G. (2001) FYVE domain targets Pib1p ubiquitin ligase to endosome and vacuolar membranes. *J. Biol. Chem.*, **276**, 41388–41393.