THE JOURNAL OF BIOLOGICAL CHEMISTRY
Vol. 277, No. 48, Issue of November 29, pp. 45765–45769, 2002

Printed in U.S.A.

Using Functional Domain Composition and Support Vector Machines for Prediction of Protein Subcellular Location*

Received for publication, April 29, 2002, and in revised form, August 8, 2002
Published, JBC Papers in Press, August 16, 2002, DOI 10.1074/jbc.M204161200

Kuo-Chen Chou‡ and Yu-Dong Cai§¶

From ‡Upjohn Laboratories, Pharmacia, Kalamazoo, Michigan 49001-4940 and §Shanghai Research Centre of Biotechnology, Chinese Academy of Sciences, Shanghai 200233, China

Proteins are generally classified into the following 12 subcellular locations: 1) chloroplast, 2) cytoplasm, 3) cytoskeleton, 4) endoplasmic reticulum, 5) extracellular, 6) Golgi apparatus, 7) lysosome, 8) mitochondria, 9) nucleus, 10) peroxisome, 11) plasma membrane, and 12) vacuole. Because the function of a protein is closely correlated with its subcellular location, with the rapid increase in new protein sequences entering into databanks, it is vitally important for both basic research and pharmaceutical industry to establish a high throughput tool for predicting protein subcellular location. In this paper, a new concept, the so-called "functional domain composition" is introduced. Based on the novel concept, the representation for a protein can be defined as a vector in a high-dimensional space, where each of the clustered functional domains derived from the protein universe serves as a vector base. With such a novel representation for a protein, the support vector machine (SVM) algorithm is introduced for predicting protein subcellular location. High success rates are obtained by the self-consistency test, jackknife test, and independent dataset test, respectively. The current approach not only can play an important complementary role to the powerful covariant discriminant algorithm based on the pseudo amino acid composition representation (Chou, K. C. (2001) *Proteins Struct. Funct. Genet.* 43, 246–255; Correction (2001) *Proteins Struct. Funct. Genet.* 44, 60), but also may greatly stimulate the development of this area.

According to the localization or compartment in a cell, proteins are generally classified into the following 12 categories: 1) chloroplast, 2) cytoplasm, 3) cytoskeleton, 4) endoplasmic reticulum, 5) extracellular, 6) Golgi apparatus, 7) lysosome, 8) mitochondria, 9) nucleus, 10) peroxisome, 11) plasma membrane, and 12) vacuole. Given the sequence of a protein, how can we predict which category or subcellular location it belongs to? This is certainly a very important problem because the subcellular location of a protein is closely correlated with its biological function. Although the information about protein subcellular location can be determined by conducting various experiments, that is both time consuming and costly. Because of the fact that the number of sequences entering into databanks has been rapidly increasing, *e.g.* in 1986 the total sequence entries in SWISS-PROT (1) was only 3,939 while the number was increased to 80,000 in 1999, the problem has become an urgent challenge. Particularly, it is anticipated that many more new protein sequences will be derived soon because of the recent success of the human genome project, which has provided an enormous amount of genomic information in the form of 3 billion base pairs assembled into tens of thousands of genes. Therefore, the challenge will become even more urgent and critical. Actually, many efforts have been made trying to develop some computational methods for quickly predicting the subcellular locations of proteins (2–13). It is instructive to point out that, of these algorithms, most are based on the amino acid composition alone without including any sequence-order effects, and some (9, 12, 13) are based on the pseudo amino acid composition that incorporated partial sequence-order effects. To further improve the prediction quality, a logical and key step would be to find an effective way to incorporate the sequence-order effects. The present study was initiated in an attempt to explore a different approach to incorporate these kinds of effects. The core of the new approach is based on a novel concept, the so-called "functional domain composition," as will be further described below.

THEORY

The Functional Domain Composition Representation

To improve the quality of statistical prediction for protein subcellular location, one of the most important steps is to give an effective representation for a protein. This is indeed a crucial problem but meanwhile a quite subtle one, which might lead us to face the dilemma discussed below. According to common sense, an effective representation should include as much information a protein has as possible. Compared with the amino acid composition (14–16) and the pseudo amino acid composition (12), the entire protein sequence contains of course the most complete information. Unfortunately, if using the entire sequence of a protein as its representation to formulate the statistical prediction algorithm, one would face the difficulty of dealing with almost an infinity of sample patterns, as elaborated by Chou (12). Accordingly, to formulate a feasible statistical prediction algorithm, a protein must be expressed in terms of a set of discrete numbers. The earliest approach (2–5) in this regard was to use the amino acid composition that consists of 20 components representing the occurrence frequencies of the 20 native amino acids in a protein. However, if using the amino acid composition as the representation for a protein, all the sequence-order effects would be missed. Therefore we are actually confronted with the dilemma that, if wishing to include the complete information, the prediction would become unfeasible; if wishing to make the prediction feasible, some important information must be ignored. In view of this, can we find a compromise scenario, *i.e.* a new protein representation that is constituted by a set of discrete numbers but that also contains as much of the sequence-order effects as possible? The introduction of the pseudo amino acid composition is a pioneer effort in this regard that has no doubt made one important step forward for such a goal. The pseudo amino acid composition consists of $20 + \lambda$ discrete numbers, where the first 20 numbers are the same as those in the amino acid composition and the remaining numbers represent $\boldsymbol{\lambda}$ different ranks of sequence-correlation factors (12). In this paper, we would like to introduce a

* The costs of publication of this article were defrayed in part by the payment of page charges. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

¶ Current address and to whom correspondence should be addressed: Biomolecular Sciences Dept., UMIST, P. O. Box 88, Manchester, M60 1QD, United Kingdom. Tel.: 44-161-2008936; Fax: 44-161-2360409; E-mail: y.cai@umist.ac.uk.

completely different set of discrete numbers; *i.e.* instead of using each of the 20 amino acid components or each of the 20 + λ pseudo amino acid components as a vector base to define a protein, we shall use each of the native functional domains as a vector base to define a protein.

By searching and clustering 139,765 annotated protein sequences, Murvai *et al.* (17) have constructed a data base called SBASE-A that contains 2005 sequences with well known structural and functional domain types. With each of the 2005 functional domains as a vector-base, a protein can be defined as a 2005-dimensional (D)[1] vector according to the following procedures. 1) Use BLASTP to compare a protein with each of the 2005 domain sequences in SBASE-A to find the high-scoring segment pairs (HSPs) and the smallest sum probability (P). A detailed description about this operation can be found in Altschul (18). 2) If the HSP score $\gg 75$ and $P < 0.8$ in comparing the protein sequence with the i*th* domain sequence, then the i*th* component of the protein in the 2005-D space is assigned 1; otherwise, 0. 3) The protein can thus be explicitly formulated as follows.

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{2005} \end{bmatrix}, \qquad \text{(Eq. 1)}$$

where

$$x_i = \begin{cases} 1, \text{ when HSP score} >> 75 \text{ and } P < 0.8 \\ 0, \text{ otherwise} \end{cases} \qquad \text{(Eq. 2)}$$

Defined in this way, a protein corresponds to a 2005-D vector $\mathbf{X}$ with each of the 2005 functional domain sequences as a base for the vector space; *i.e.* rather than the 20-D space (15) of the amino acid composition approach or the (20 + λ)-D space of the pseudo amino acid composition approach (12), a protein is represented in terms of the functional domain-composition. By using such a representation, not only some sequence-order effects but also some functional information is included. In other words, the representation thus obtained for a protein would bear some sequence-order mark as well as the structural and functional type mark. Because the function of a protein is closely related to its subcellular location, the prediction algorithm established based on the new representation would naturally incorporate those factors that might be directly correlated with the protein subcellular location.

Support Vector Machines

Support Vector Machines (SVMs) are kinds of learning machines based on statistical learning theory. The most remarkable characteristics of SVMs are the absence of local minima, the sparseness of the solution, and the use of the kernel-induced feature spaces. The basic idea of applying SVMs to pattern classification can be outlined as follows. First, map the input vectors into a feature space (possible with a higher dimension) either linearly or non-linearly, which is relevant to the selection of the kernel function. Then, within the feature space, seek an optimized linear division; *i.e.* construct a hyper-plane that can separate two classes (this can be extended to multi-classes) with the least error and maximal margin. The SVMs training process always seeks a global optimized solution and avoids over-fitting, so it has the ability to deal with a large number of features. A complete description to the theory of SVMs for pattern recognition is given in the book by Vapnik (19). SVMs have been used to deal with protein fold recognition (20), protein-protein interaction prediction (21), and protein secondary structure prediction (22).

In this paper, the Vapnik's Support Vector Machine (23) was introduced to predict protein subcellular location. Specifically, SVMlight, which is an implementation (in C Language) of SVM for the problems of pattern recognition, was used for computations. The optimization algorithm used in SVMlight can be found in Joachims (24). The relevant mathematical principles can be briefly formulated as follows.

Given a set of n samples, *i.e.* a series of input vectors

$$\mathbf{X}_k \in \Re^\tau \, (k = 1, \ldots, N), \qquad \text{(Eq. 3)}$$

where $\mathbf{X}_k$ can be regarded as the kth protein or vector defined in the 2005-D space according to the functional domain composition (see Eq. 1), and $\Re^\tau$ is a Euclidean space with τ dimensions. Because the multi-class identification problem can always be converted into a two-class identification problem, without loss of the generality, the formulation below is given for the two-class case only. Suppose the output derived from the learning machine is expressed by $y_k \in \{+1, -1\}$ $(k = 1, \ldots, N)$ where the indexes −1 and +1 are used to stand for the two classes concerned, respectively. The goal here is to construct one binary classifier or derive one decision function from the available samples that has a small probability of misclassifying a future sample. Here both the basic linear separable case and the most useful linear non-separable case for most real life problems are taken into consideration.

The Linear Separable Case

In this case, there exists a separating hyper-plane whose function is $\mathbf{W} \cdot \mathbf{X} + b = 0$, whose implication is shown in the following equation.

$$y_k(\mathbf{W} \cdot \mathbf{X}_k + b) \geq 1, \, (k = 1, \ldots, N) \qquad \text{(Eq. 4)}$$

By minimizing $\|\mathbf{W}\|^2$ subject to the above constraint, the SVM approach will find a unique separating hyper-plane. Here $\|\mathbf{W}\|^2$ is the Euclidean norm of $\mathbf{W}$, which maximizes the distance between the hyper-plane or the optimal separating hyper-plane (25) and the nearest data points of each class. The classifier thus obtained is called the maximal margin classifier. By introducing Lagrange multipliers α_i, and using the Karush-Kuhn-Tucker conditions (26, 27) as well as the Wolfe dual theorem of optimization theory (28), the SVM training procedure amounts to solving the following convex quadratic programming problem

$$\textbf{Max:} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{X}_i \cdot \mathbf{X}_j \qquad \text{(Eq. 5)}$$

subject to the following two conditions.

$$\alpha_i \geq 0, \, (i = 1, 2, \ldots, N) \qquad \text{(Eq. 6)}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \text{(Eq. 7)}$$

The solution is a unique globally optimized result, which can be expressed with the following expansion.

$$\mathbf{W} = \sum_{i=1}^{N} y_i \alpha_i \mathbf{X}_i \qquad \text{(Eq. 8)}$$

Only if the corresponding $\alpha_i > 0$, are these $\mathbf{X}_i$ called the Support Vectors. Now suppose $\mathbf{X}$ is a query protein defined in the same 2005-D space based on the functional domain composition (see Eq. 1). After the SVM has been trained, the decision function for identifying which class the query protein belongs to can be formulated as

$$f(\mathbf{X}) = \text{sgn} \left(\sum_{i=1}^{N} y_i \alpha_i \mathbf{X} \cdot \mathbf{X}_i + b \right) \qquad \text{(Eq. 9)}$$

where sgn() in the above equation is a sign function, which equals to +1 or −1 when its argument is ≥0 or <0, respectively.

The Linear Non-separable Case

For this case two important techniques are needed that are given below.

The "Soft Margin" Technique—To allow for training errors, Cortes and Vapnik (25) introduced the slack variables shown below.

$$\zeta_i > 0 \, (i = 1, \ldots, N), \qquad \text{(Eq. 10)}$$

The relaxed separation constraint is given below.

$$y_i(\mathbf{W} \cdot \mathbf{X}_i + b) \geq 1 - \zeta_i, \, (i = 1, \ldots, N) \qquad \text{(Eq. 11)}$$

The optimal separating hyper-plane can be found by minimizing

$$\frac{1}{2}\|\mathbf{W}\|^2 + c \sum_{i=1}^{N} \zeta_i \qquad \text{(Eq. 12)}$$

where c is a regularization parameter used to decide a trade-off between the training error and the margin.

The "Kernel Substitution" Technique—The SVM performs a nonlin-

[1] The abbreviations used are: D, dimensional; HSP(s), high-scoring segment pair(s); SVM(s), support vector machine(s).

TABLE I
Overall rates of correct prediction for the 12 subcellular locations of proteins by different algorithms and test methods

Algorithm	Input form	Test method		
		Resubstitution[a]	Jackknife[a]	Independent dataset[b]
Least Hamming distance (33)	Amino acid composition	$\frac{1067}{2191} = 48.7\%$	$\frac{1033}{2191} = 47.2\%$	$\frac{1151}{2494} = 46.2\%$
Least Euclidean distance (14)	Amino acid composition	$\frac{1096}{2191} = 50.0\%$	$\frac{1063}{2191} = 48.5\%$	$\frac{1197}{2494} = 48.0\%$
ProtLock (3)	Amino acid composition	$\frac{1023}{2191} = 46.7\%$	$\frac{971}{2191} = 44.3\%$	$\frac{1018}{2494} = 40.8\%$
Covariant-discriminant (7)	Amino acid composition	$\frac{1751}{2191} = 79.9\%$	$\frac{1492}{2191} = 68.1\%$	$\frac{1888}{2494} = 75.7\%$
Augmented covariant discriminant (9)	Pseudo amino acid composition (12)	$\frac{1880}{2191} = 85.8\%$	$\frac{1600}{2191} = 73.0\%$	$\frac{2017}{2494} = 80.9\%$
Support vector machines	Functional domain composition	$\frac{1913}{2191} = 87.3\%$	$\frac{1461}{2191} = 66.7\%$	$\frac{2037}{2494} = 81.7\%$

[a] Conducted for the 2191 proteins classified into 12 subcellular locations in the training dataset as described under "Results and Discussion."
[b] Conducted based on the rule parameters derived from the 2191 proteins in the training dataset for the 2494 proteins in the independent dataset (see "Results and Discussion").

ear mapping of the input vectors from the Euclidean space 194 ^{d} into a higher dimensional Hilbert space H, where the mapping is determined by the kernel function. Then like in the linear separable case, it finds the optimal separating hyper-plane in the Hilbert space H that would correspond to a non-linear boundary in the original Euclidean space. Two typical kernel functions are listed below

$$K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^\tau \quad \text{(Eq. 13)}$$

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp(-r\|\mathbf{X}_i - \mathbf{X}_j\|^2) \quad \text{(Eq. 14)}$$

where the first one is called the *polynomial kernel function of degree* τ, which will eventually revert to the linear function when $\tau = 1$, the second one is called the radial basic function kernel. Finally, for the selected kernel function, the learning task amounts to solving the following quadratic programming problem

$$\textbf{Max:} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{X}_i \cdot \mathbf{X}_j) \quad \text{(Eq. 15)}$$

which is subject to the following equations.

$$0 \le a_i \le c, (i = 1, 2, \ldots, N) \quad \text{(Eq. 16)}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{(Eq. 17)}$$

Accordingly, the form of the decision function is given by the equation shown below.

$$f(\mathbf{X}) = \text{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i K(\mathbf{X}, \mathbf{X}_i) + b\right) \quad \text{(Eq. 18)}$$

For a given dataset, only the kernel function and the regularity parameter c must be selected to specify the SVM.

RESULTS AND DISCUSSION

To facilitate comparison, the same dataset constructed by Chou and Elrod (6) was used to demonstrate the current method. However, as mentioned in Chou (12), because the change of code names, some protein sequences could no longer be retrieved from the SWISS-PROT data bank (1). Of the 2319 proteins originally listed in Appendix A of Chou and Elrod (6), 2191 protein sequences were retrieved. These sequences consist of 145 chloroplast proteins, 571 cytoplasm, 34 cytoskeleton, 49 endoplasmic reticulum, 224 extracellular, 25 Golgi apparatus, 37 lysosome, 84 mitochondria, 272 nucleus proteins, 27 peroxisome, 699 plasma membrane, and 24 vacuole.

During the operation, the width of the Gaussian radial basic functions was selected for minimizing the estimation of the Vapnik-Chervonenkis dimension (19). The parameter c that controlled the error-margin trade-off was set at 1000. After being trained, the hyper-plane output by the SVM was obtained. This indicates that the trained model, *i.e.* hyper-plane output that includes the important information, has the function to identify the protein subcellular locations.

The demonstration was conducted by the three most typical approaches in statistical prediction (29), *i.e.* the resubstitution test, jackknife test, and independent dataset test as reported below.

Resubstitution Test—The so-called resubstitution test is an examination for the self-consistency of an identification method. When the resubstitution test is performed for the current study, the subcellular location of each protein in the dataset is in turn identified using the rule parameters derived from the same dataset, the so-called training dataset. The success rate thus obtained for predicting the 12 subcellular locations of the 2191 proteins is summarized in Table I, from which we can see that 1913 proteins were correctly predicted for their subcellular locations and that only 278 proteins were incorrectly predicted. The overall success rate is 87.3%, indicating that after being trained, the SVMs model has grasped the complicated relationship between the functional domain composition and the subcellular location of proteins. However, during the process of the resubstitution test, the rule parameters derived from the training dataset include the information of the query protein later plugged back in the test. This will certainly underestimate the error and enhance the success rate because the same proteins are used to derive the rule parameters and to test themselves. Accordingly, the success rate thus obtained represents an optimistic estimation (6, 15, 30, 31). Nevertheless, the resubstitution test is absolutely necessary because it reflects the self-consistency of an identification method, especially for its algorithm part. An identification algorithm certainly cannot be deemed as a good one if its self-consistency is poor. In other words, the resubstitution test is necessary but not sufficient for evaluating an identification method. As a complement, a cross-validation test for an independent testing dataset is needed because it can reflect the effectiveness of an identification method in practical application. This is important especially for checking the validity of a training data base: whether it contains sufficient information to reflect all the important features concerned so as to yield a high success rate in application.

Jackknife Test—As is well known, the independent dataset test, subsampling test and jackknife test are the three methods often used for cross-validation in statistical prediction. Among these three, however, the jackknife test is deemed as the most

effective and objective one; see, *e.g.* a relevant review (29) for a comprehensive discussion about this and a monograph (32) for the mathematical principle. During jackknifing, each protein in the dataset is in turn singled out as a tested protein and all the rule parameters are calculated based on the remaining proteins. In other words, the subcellular location of each protein is identified by the rule parameters derived using all the other proteins except the one which is being identified. During the process of jackknifing both the training dataset and testing dataset are actually open, and a protein will in turn move from one to the other. The results of jackknife test thus obtained for the 2191 proteins are given in Table I as well.

Independent Dataset Test—Moreover, as a demonstration of practical application, predictions were also conducted for an independent dataset based on the rule parameters derived from the 2191 proteins in the training dataset. The independent dataset was also adopted from Ref. 6. However, for the same reason as mentioned in Ref. 12, of the 2591 independent proteins originally studied by Ref. 6, only 2494 protein sequences were retrieved. They are: 112 chloroplast proteins, 761 cytoplasm, 19 cytoskeleton, 106 endoplasmic reticulum, 95 extracellular, 4 Golgi apparatus, 31 lysosome, 163 mitochondria, 418 nucleus proteins, 23 peroxisome, and 762 plasma membrane. None of these proteins occurs in the training dataset. The predicted results thus obtained for the 2494 proteins in the independent dataset are also summarized in Table I, from which we can see that 2037 proteins were correctly predicted for their subcellular locations, and only 478 proteins were incorrectly predicted. The overall success rate is 81.7%.

Furthermore, to facilitate comparison, listed in Table I are also the results predicted by various other methods on the same datasets. From Table I the following can be observed. 1) The success prediction rates by both the functional domain composition approach and the pseudo amino acid composition approach are significantly higher than those by the simple geometry approaches (14, 33) and the ProtLock algorithm (3). This is fully consistent with what is expected because both of these two approaches bear the marks of some sequence-order effects although by means of different avenues. 2) A comparison between the functional domain composition approach and the pseudo amino acid composition approach indicates that the success rates by the former are higher than the latter in both the self-consistency test and independent dataset test, indicating the current functional domain composition approach is quite promising with a considerable potential for further development. However, its success rate by the jackknife test is lower than that of the pseudo amino acid approach (12) by using the augmented covariant discriminant algorithm (9) and even 1.4% lower than that of the conventional amino acid approach by using the covariant discriminant algorithm (6). The setback might be due to the reason that the functional domain data base used in the current study is far from a complete one yet. Also, some subsets in the training dataset are too small (*e.g.* with less than 50 sequences) to have a high cluster-tolerant capacity (34) for the 2005-D functional domain composition space. It is anticipated that with the continuous improvement of the functional domain data base as well as the training data of small subsets by adding into them more new proteins that have been found belonging to the subcellular locations defined by these subsets the setback would be naturally overcome. As a demonstration, the testing dataset was incorporated into the original training dataset to form an augmented training dataset of 2191 + 2494 = 4685 proteins, and then the jackknife test was reapplied. It was observed that the success rate thus obtained by the current functional domain composition approach was increased from 66.7 to 87.9%, but the corresponding rate by the augmented covariant discriminant algorithm was increased from 73.0 to 79.5%, convincingly indicating the strong potential of the current approach.

The goal of this study is not to determine the possible upper limit of the success rate for the prediction of protein subcellular location, but to propose a novel and different approach to incorporate the sequence-order effect as well as the factors of structural and functional types that might help to open a new avenue to further increase our ability or options to deal with this very complicated and difficult problem. It should be realized that it is too premature to construct a complete or quasi-complete training dataset based on the knowledge available so far. Without a complete or quasi-complete training dataset, any attempt to determine such an upper limit would be unjustified, and the result thus obtained might be misleading no matter how powerful the prediction algorithm is.

It should be pointed out that some proteins are known to be shuttled from one subcellular compartment to another and back again. For example, if a query protein is shuttled between nucleus and cytoplasm, then only half-correction should be counted even its subcellular location was predicted to be one of the two locations. However, cases like that would not happen in the current study. This is because the same dataset constructed by Chou and Elrod (6) was used to demonstrate the current method. Compared with the other datasets in this area that only cover two-five subcellular locations, the dataset used here has covered many more subcellular locations. Even though, as clearly described by the authors of Ref. 6, "Sequences annotated by two or more locations are not included because of a lack of uniqueness. For example, a protein sequence labeled with "Golgi and nuclear" or "chloroplast or mitochondria" was omitted." Those proteins that are known to be shuttled between subcellular compartments must be annotated by two or more locations in SWISS-PROT. According to the screen procedure, they were already excluded from the dataset.

CONCLUSION

The above results, together with those obtained by the covariant discriminant prediction algorithm (6) and those further improved by introducing some sequence-order effects (9, 12), have indicated that the subcellular locations of proteins are predictable with a considerable accuracy. The development in statistical prediction of protein attributes generally consists of two cores: one is to construct a training dataset and the other is to formulate a prediction algorithm. The latter can be further separated into two subcores: one is how to give a mathematical expression to effectively represent a protein and the other is how to find an operational equation to accurately perform the prediction. The process in expressing a protein from the 20-D amino acid composition space (14, 15, 33, 35) to the (20 + λ)-D pseudo amino acid composition space (12) and to the current 2005-D functional domain composition space reflects the development of defining a protein in terms of different mathematical representations. The process in conducting prediction using from the simple geometry distance algorithm (14, 33, 35), to the Mahalanobis distance algorithm (3, 15, 16), to the covariant discriminant algorithm (6, 7, 36–38), and to the current SVM algorithm reflects the development of computation by means of different mathematical operations. One of the remarkable advantages for the pseudo amino acid composition representation is to use a set of simple and intuitive sequence-order-coupling modes to directly incorporate the sequence-order effects, while a remarkable advantage of the functional domain composition representation is to use the functional domain data base to incorporate the information of not only some sequence-order effects but also the structural and functional types. Each of the two representations has its own advantage. For some cases, the

functional domain composition representation yields better results than the pseudo amino acid composition representation; but for some other cases, the outcome may be just the reverse. This is just exactly the same in comparison of the covariant discriminant algorithm with the SVM algorithm. Therefore, when we are still in the situation of lacking a complete training dataset and functional domain data base, it would be wise to complement the covariant discriminant algorithm based on the pseudo amino acid composition representation with the SVM algorithm based on the functional domain composition representation for conducting practical predictions. Finally, the functional domain composition approach might have more room and potential for further development because it incorporates both the sequence-order information and the functional type information.

The program of the new prediction method, called CLPFD (Cellular Location Prediction based on Functional Domains), is available by contacting Y. D. Cai at y.cai@umist.ac.uk.

REFERENCES

1. Bairoch, A., and Apweiler, R. (2000) *Nucleic Acids Research* **25,** 31–36
2. Nakashima, H., and Nishikawa, K. (1994) *J. Mol. Biol.* **238,** 54–61
3. Cedano, J., Aloy, P., P'erez-pons, J. A., and Querol, E. (1997) *J. Mol. Biol.* **266,** 594–600
4. Reinhardt, A., and Hubbard, T. (1998) *Nucleic Acids Res.* **26,** 2230–2236
5. Chou, K. C., and Elrod, D. W. (1998) *Biochem. Biophys. Res. Commun.* **252,** 63–68
6. Chou, K. C., and Elrod, D. W. (1999) *Protein Eng.* **12,** 107–118
7. Chou, K. C., and Elrod, D. W. (1999) *Proteins Struct. Funct. Genet.* **34,** 137–153
8. Chou, K. C. (2000) *Curr. Protein Peptide Sci.* **1,** 171–208
9. Chou, K. C. (2000) *Biochem. Biophys. Res. Comm.* **278,** 477–483
10. Cai, Y. D., and Chou, K. C. (2000) *Mol. Cell Biol. Res. Comm.* **4,** 172–173
11. Cai, Y. D., Liu, X. J., Xu, X. B., and Chou, K. C. (2000) *Mol. Cell Biol. Res. Comm.*, **4,** 230–233
12. Chou, K. C. (2001) *Proteins Struct. Funct. Genet.* **43**, 246–255 Correction (2001) *Proteins Struct. Funct. Genet.* **44,** 60
13. Cai, Y. D., Liu, X. J., Xu, X. B., and Chou, K. C. (2002) *J. Cell. Biochem.* **84,** 343–348
14. Nakashima, H., Nishikawa, K., and Ooi, T. (1986) *J. Biochem.* **99,** 152–162
15. Chou, K. C. (1995) *Proteins Struct. Funct. Genet.* **21,** 319–344
16. Chou, K. C., and Zhang, C. T. (1994) *J. Biol. Chem.* **269,** 22014–22020
17. Murvai, J., Vlahovicek, K., Barta, E., and Pongor, S. (2001) *Nucleic Acids Res.* **29,** 58–60
18. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) *J. Mol. Biol.* **215,** 403–410
19. Vapnik, V. (1998) *Statistical Learning Theory*, Wiley-Interscience, New York
20. Ding, C. H., and Dubchak, I. (2001) *Bioinformatics* **17,** 349–358
21. Bock, J. R., and Gough, D. A. (2001) *Bioinformatics* **17,** 455–460
22. Hua, S. J., and Sun, Z. R. (2001) *J. Mol. Biol.* **308,** 397–407
23. Vapnik, V. N. (1995) *The Nature of Statistical Learning Theory*, Springer-Verlag New York Inc., New York
24. Joachims, T. (1999) in *Advances in Kernel Methods-Support Vector Learning* (Schölkopf, B., Burges, C. J. C., and Smola, A. J., ed), pp. 169–184, MIT Press, Cambridge, MA
25. Cortes, C., and Vapnik, V. (1995) *Machine Learning* **20,** 273–293
26. Karush, W. (1939) *Department of Mathematics*, University of Chicago, Chicago
27. Cristianini, N., and Shawe-Taylor, J. (2000) *Support Vector Machines*, Cambridge University Press, Cambridge, UK
28. Wolfe, P. (1961) *Q. Appl. Math.* **19,** 239–244
29. Chou, K. C., and Zhang, C. T. (1995) *Crit. Rev. Biochem. Mol. Biol.* **30,** 275–349
30. Cai, Y. D. (2001) *Proteins Struct. Funct. Genet.* **43,** 336–338
31. Zhou, G. P., and Assa-Munt, N. (2001) *Proteins Struct. Funct. Genet.* **44,** 57–59
32. Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979) *Multivariate Analysis*, pp. 322 and 381, Academic Press, London
33. Chou, P. Y. (1980) *Abstracts of Papers*, *Part I*, *Second Chemical Congress of the North American Continent*, *Las Vegas*, May 1, 1980, Plenum Press, New York
34. Chou, K. C. (1999) *Biochem. Biophys. Res. Comm.* **264,** 216–224
35. Chou, P. Y. (1989) in *Prediction of Protein Structure and the Principles of Protein Conformation* (Fasman, G. D., ed), pp. 549–586, Plenum Press, New York
36. Chou, K. C., Liu, W., Maggiora, G. M., and Zhang, C. T. (1998) *Proteins Struct. Funct. Genet.* **31,** 97–103
37. Liu, W., and Chou, K. C. (1998) *J. Prot. Chem.* **17,** 209–217
38. Zhou, G. P. (1998) *J. Prot. Chem.* **17,** 729–738