Virus-ECC-mPLoc: A Multi-Label Predictor for Predicting the Subcellular Localization of Virus Proteins with Both Single and Multiple Sites Based on a General Form of Chou's Pseudo Amino Acid Composition

Xiao Wang¹, Guo-Zheng Li¹,* and Wen-Cong Lu²

¹The MOE Key Laboratory of Embedded System and Service Computing, Department of Control Science and Engineering, Tongji University, Shanghai 201804, China; ²Department of Chemistry, College of Science, Shanghai University, Shanghai 200444, China

Abstract: Protein subcellular localization aims at predicting the location of a protein within a cell using computational methods. Knowledge of subcellular localization of viral proteins in a host cell or virus-infected cell is important because it is closely related to their destructive tendencies and consequences. Prediction of viral protein subcellular localization is an important but challenging problem, particularly when proteins may simultaneously exist at, or move between, two or more different subcellular location sites. Most of the existing protein subcellular localization methods specialized for viral proteins are only used to deal with the single-location proteins. To better reflect the characteristics of multiplex proteins, a new predictor, called Virus-ECC-mPLoc, has been developed that can be used to deal with the systems containing both singleplex and multiplex proteins by introducing a powerful multi-label learning approach which exploits correlations between subcellular locations and by hybridizing the gene ontology information with the dipeptide composition information. It can be utilized to identify viral proteins among the following six locations: (1) viral capsid, (2) host cell membrane, (3) host endoplasmic reticulum, (4) host cytoplasm, (5) host nucleus, and (6) secreted. Experimental results show that the overall success rates thus obtained by Virus-ECC-mPLoc are 88.9% for jackknife test and 87.2% for the independent data set test, which are significantly higher than that by any of the existing predictors. As a user-friendly web-server, Virus-ECC-mPLoc is freely accessible to the public at the web-site http://levis.tongji.edu.cn:8080/bioinfo/Virus-ECC-mPLoc/.

Keywords: Protein subcellular localization, multi-label learning, classifier chain, multiplex proteins.

1. INTRODUCTION

A virus is a small particle that infects cells in various organisms. As acellular organisms and obligate intracellular parasites, viruses can reproduce themselves only by invading and taking over other cells as they lack the cellular machinery for self-reproduction. Although viruses are acellular organisms, viral proteins are required to reside in various cellular compartments of the host cell or virus-infected cell to perform their functions. Therefore, knowledge of the subcellular localization of viral proteins within a host cell or virus-infected cell is very useful for studying the function of viral proteins and designing antiviral drugs because it is closely related to their destructive tendencies and consequences.

Although the subcellular localization of a protein can be determined by carrying out various biochemical experiments, the approach by purely doing experiments is both time-consuming and high cost. In the post-genomic age, the gap between newly found protein sequences and the information of their subcellular localization is becoming increasingly wide. For example, according to the Swiss-Prot database, 14

*Address correspondence to this author at the MOE Key Laboratory of Embedded System and Service Computing, Department of Control Science and Engineering, 4800 Cao An Road, Shanghai 201804, China; Tel: +86 021 6958 3706; Fax: +86 021 6958 3706; E-mail: gzli@tongji.edu.cn

version 50.0 released on May 30, 2006, the number of viral proteins with reliable subcellular location annotations is about 12% of all the viral protein entries concerned [1]. To bridge such a gap, it is highly desirable to develop computational methods to predict protein subcellular localization automatically and accurately. During the past decade, many efforts have been devoted to deal with such a challenge, and a large number of computational methods have been developed in an attempt to predict the subcellular localization of proteins (see, e.g., [2-21]). In particular, machine-learning based approaches, such as Neural Networks, K Nearest Neighbor, Support Vector Machine, AdaBoost, are widely used to solve this problem. Among these approaches, Support Vector Machine is widely adopted in bioinformatics and is shown to achieve better performance as compared with others.

However, proteins may simultaneously reside at, or move between, two or more different subcellular locations. Unfortunately, the aforementioned methods didn't take multiple-location or multiplex proteins into account when predicting protein subcellular localization. In general, they were established under the assumption that a protein resides at one, and only one, subcellular location. Proteins with multiple location sites or dynamic feature of this kind are particularly interesting because they may have some unique biological functions worthy of our special notice [22]. In particular,
recent evidences have indicated that an increasing number of proteins have multiple locations in the cell, as indicated by Millar et al. [23].

Recently, a powerful predictor, called iLoc-Virus [24] was developed that can be used to predict the subcellular localization of viral proteins among their 6 location sites in which some of the proteins may belong to two and more subcellular locations. To the best of our knowledge, iLoc-Virus is at present the best predictor able to deal with multiple-location or multiplex proteins when predicting viral protein subcellular localization. However, ML-KNN classifier used by iLoc-Virus is not optimal because it doesn’t take correlations between subcellular locations into account.

In this paper, to better reflect the characteristics of multiplex proteins, a new predictor, called Virus-ECC-mPLoc, has been developed that can be used to deal with the systems containing both singleplex and multiplex proteins by introducing a powerful multi-label learning approach which exploits correlations between subcellular locations and by hybridizing the gene ontology information with the dipeptide composition information. Our experimental results on a benchmark dataset consisting of 207 viral protein sequences show that the overall success rates thus obtained by Virus-ECC-mPLoc are 86.9% for jackknife test and 87.2% for independent data set test, which are significantly higher than that by iLoc-Virus predictor.

According to a recent comprehensive review [25], to establish a really useful statistical predictor for a protein system, we need to consider the following procedures: (i) construct or select a valid benchmark dataset to train and test the predictor; (ii) formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target concerned; (iii) introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) establish a user-friendly web-server for the predictor that is accessible to the public. Below, let us elaborate how to deal with these steps one-by-one.

2. MATERIALS AND METHODS

2.1. Datasets

In this study, we use the same dataset $X$ in iLoc-Virus [24] as the benchmark dataset for the current study. Using the dataset $X$ will make it easy to compare our new predictor with the existing one because the tested results by iLoc-Virus on $X$ have been reported [24]. Furthermore, the dataset is constructed specialized for viral proteins, where none of proteins included in $X$ has greater than or equal to 25% pairwise sequence identity to any other in a same subcellular location compared with most of the other benchmark datasets in this area.

The dataset $X$ contains 207 viral protein sequences, of which 165 belong to one subcellular location, 39 to two locations, 3 to three locations, and none to four or more locations. The dataset covers 6 subcellular locations as shown in Fig. (1), and hence can be represented as

$$X = X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5 \cup X_6,$$

where $X_i$ represents the subset for the subcellular location of “viral capsid”, $X_2$ for “host cell membrane”, $X_3$ for “host endoplasmic reticulum”, and so forth. A breakdown of the 207 viral proteins in the benchmark dataset $X$ according to their six location sites is given in Table 1. To avoid redundancy and homology bias, none of the proteins in $X$ has greater than or equal to 25% pairwise sequence identity to any other in a same subset. For convenience, hereafter let us just use the subscripts of Eq. (1) as the codes of the 6 location sites; i.e., “1” for “viral capsid”, “2” for “host cell membrane”, “3” for “host endoplasmic reticulum”, and so forth (Table 2).

Note that because some proteins may occur in two different locations, the 207 different proteins actually correspond to 252 “locative proteins” (Table 1). For the concept of locative proteins, readers are referred to [27-29] where the difference between “protein” and “locative protein” and their relationship are elaborated.

For readers’ convenience, the corresponding accession numbers and protein sequences in $X$ are given in Online Supporting Information A.

2.2. Feature Extraction

To develop a powerful method for statistically predicting protein subcellular localization, one of the most important things is to extract core and essential features of protein samples that are closely correlated with their subcellular locations. To avoid losing many important information hidden in protein sequences, the pseudo amino acid composition (PseAAC) was proposed [30, 31] to replace the simple amino acid composition (AAC) for representing the sample

| Table 1. Breakdown of the Viral Protein Benchmark Dataset $X$ Taken from [24] |
|------------------|------------------|------------------|
| **Subset**      | **Subcellular location** | **Number of proteins** |
| $X_1$           | Viral capsid      | 8                |
| $X_2$           | Host cell membrane| 33               |
| $X_3$           | Host endoplasmic reticulum | 20             |
| $X_4$           | Host cytoplasm    | 87               |
| $X_5$           | Host nucleus      | 84               |
| $X_6$           | Secreted          | 20               |
| Total number of locative proteins $N(\text{loc})$ | 252<sup>a</sup> |
| Total number of different proteins $N(\text{seq})$ | 207<sup>b</sup> |

None of proteins included here has ≥ 25% sequence identity to any other in a same subcellular location.

<sup>a</sup>Of the 207 different proteins, 165 have one subcellular location, 39 have two locations, and 3 have three locations. See Online Supporting Information A for the protein sequences.

<sup>b</sup>Virus on $X$ has been reported [24]. Furthermore, the benchmark dataset consisting of 207 viral protein sequences, of which some of the proteins may belong to two and more subcellular locations. To the best of our knowledge, iLoc-Virus is at present the best predictor able to deal with multiple-location or multiplex proteins when predicting viral protein subcellular localization. However, ML-KNN classifier used by iLoc-Virus is not optimal because it doesn’t take correlations between subcellular locations into account.

In this paper, to better reflect the characteristics of multiplex proteins, a new predictor, called Virus-ECC-mPLoc, has been developed that can be used to deal with the systems containing both singleplex and multiplex proteins by introducing a powerful multi-label learning approach which exploits correlations between subcellular locations and by hybridizing the gene ontology information with the dipeptide composition information. Our experimental results on a benchmark dataset consisting of 207 viral protein sequences show that the overall success rates thus obtained by Virus-ECC-mPLoc are 86.9% for jackknife test and 87.2% for independent data set test, which are significantly higher than that by iLoc-Virus predictor.

According to a recent comprehensive review [25], to establish a really useful statistical predictor for a protein system, we need to consider the following procedures: (i) construct or select a valid benchmark dataset to train and test the predictor; (ii) formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target concerned; (iii) introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) establish a user-friendly web-server for the predictor that is accessible to the public. Below, let us elaborate how to deal with these steps one-by-one.

2. MATERIALS AND METHODS

2.1. Datasets

In this study, we use the same dataset $X$ in iLoc-Virus [24] as the benchmark dataset for the current study. Using the dataset $X$ will make it easy to compare our new predictor with the existing one because the tested results by iLoc-Virus on $X$ have been reported [24]. Furthermore, the dataset is constructed specialized for viral proteins, where none of proteins included in $X$ has greater than or equal to 25% pairwise sequence identity to any other in a same subcellular location compared with most of the other benchmark datasets in this area.

The dataset $X$ contains 207 viral protein sequences, of which 165 belong to one subcellular location, 39 to two locations, 3 to three locations, and none to four or more locations. The dataset covers 6 subcellular locations as shown in Fig. (1), and hence can be represented as

$$X = X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5 \cup X_6,$$

where $X_i$ represents the subset for the subcellular location of “viral capsid”, $X_2$ for “host cell membrane”, $X_3$ for “host endoplasmic reticulum”, and so forth. A breakdown of the 207 viral proteins in the benchmark dataset $X$ according to their six location sites is given in Table 1. To avoid redundancy and homology bias, none of the proteins in $X$ has greater than or equal to 25% pairwise sequence identity to any other in a same subset. For convenience, hereafter let us just use the subscripts of Eq. (1) as the codes of the 6 location sites; i.e., “1” for “viral capsid”, “2” for “host cell membrane”, “3” for “host endoplasmic reticulum”, and so forth (Table 2).

Note that because some proteins may occur in two different locations, the 207 different proteins actually correspond to 252 “locative proteins” (Table 1). For the concept of locative proteins, readers are referred to [27-29] where the difference between “protein” and “locative protein” and their relationship are elaborated.

For readers’ convenience, the corresponding accession numbers and protein sequences in $X$ are given in Online Supporting Information A.

2.2. Feature Extraction

To develop a powerful method for statistically predicting protein subcellular localization, one of the most important things is to extract core and essential features of protein samples that are closely correlated with their subcellular locations. To avoid losing many important information hidden in protein sequences, the pseudo amino acid composition (PseAAC) was proposed [30, 31] to replace the simple amino acid composition (AAC) for representing the sample
of a protein. For a brief introduction about Chou’s PseAAC, visit the Wikipedia web-page at http://en.wikipedia.org/wiki/Pseudo_amino_acid_composition. For a summary about its recent development and applications, see a comprehensive review [32]. Ever since the concept of PseAAC was proposed by Chou [30] in 2001, it has rapidly penetrated into almost all the fields of protein attribute prediction, such as identifying bacterial virulent proteins [33], predicting homologous proteins [34], predicting protein secondary structure content [35], predicting super-secondary structure [36], predicting protein quaternary structure [39], predicting enzyme family and sub-family classes [40-42], predicting protein subcellular location [43], predicting subcellular localization of apoptosis proteins [44-47], predicting protein subnuclear location [48], predicting protein sub mitochondria locations [49-50], identifying cell wall lytic enzymes [51], identifying risk type of human papillomaviruses [52], identifying DNA-binding proteins [53], predicting G-Protein-Coupled Receptor Classes [54, 55], predicting protein folding rates [56], predicting outer membrane proteins [57], predicting cyclin proteins [58], predicting GABA(A) receptor proteins [59], identifying bacterial secreted proteins [60], identifying the cofactors of oxidoreductases [61], identifying lipase types [62], identifying protease family [63], predicting Golgi protein types [64], classifying amino acids [65], among many others. In the present study, we adopted Gene Ontology and Dipeptide Composition feature extraction methods to generate features of protein examples, which are widely used in many existing protein subcellular localization systems [24, 27-29, 66-73]. For reader’s convenience, a brief introduction on Gene Ontology and Dipeptide Composition is given below.

**GO (Gene Ontology)** GO database [74] was established according to the molecular function, biological process, and cellular component. Accordingly, protein samples defined in a GO database would be clustered in a way better reflecting their subcellular locations [26, 75]. So far, there are two main approaches to extract features from GO database space. However, in order to incorporate more information, instead of only using 0 and 1 elements as done in [66], here let us use another better approach [68] as described below.

**Step 1** Compression and reorganization of the existing GO numbers. The GO database (version 94 released 08 April 2011) contains many GO numbers. However, these numbers do not increase successively and orderly. For easier handling, some reorganization and compression procedure was taken to renumber them. The GO database obtained through such a treatment is called GO_compress database, which contains 18,844 numbers increasing successively from 1 to the last one.

**Step 2** According to Eq. (6) of a recent comprehensive review [25], the general form of Chou’s PseAAC can be formulated as

\[ P = \left[ \psi_1, \psi_2, \ldots, \psi_\Omega \right]^T, \quad (2) \]

where \( T \) is a transpose operator, while the subscript \( \Omega \) reflects the dimension of the vector and its value as well as the components \( \psi_1, \psi_2, \ldots, \psi_\Omega \) will be defined by the feature extractions as elaborated below. Using Eq. (2) with \( \Omega = 18,844 \), the protein P can be represented as

\[ P_{GO} = \left[ f_1^G, f_2^G, \ldots, f_u^G, \ldots, f_{18844}^G \right]^T, \quad (3) \]

where \( f_u^G (u = 1,2,\ldots,18,844) \) are defined via the following steps.

**Step 3** Use BLAST [76] to search the homologous proteins of the protein P from the Swiss-Prot database (version 55.3), with the expect value \( E \leq 0.001 \) for the BLAST parameter.

**Step 4** Those proteins which have \( \geq 60\% \) pairwise sequence identity with the protein P are collected into a set, \( X_{P-homo} \), called the “homology set” of P. All the elements in \( X_{P-homo} \) can be deemed as the “representative proteins” of P, sharing some similar attributes such as structural conformations and biological functions. Because they were retrieved from the Swiss-Prot database, these representative proteins must each have their own accession numbers.

**Step 5** Search the GO database at http://www.ebi.ac.uk/GOA/ to find the corresponding GO number(s) [77] for each of the accession numbers collected in Step 4, followed by converting the GO numbers thus obtained to their GO_compress numbers as described in Step 1.

**Step 6** The elements in Eq. (2) is given by

\[ f_u^G = \frac{N(rep)}{\sum k=1 N(rep)} g(u,k) \quad (u = 1,2,\ldots,18844), \quad (4) \]

where \( N(rep) \) is the number of representative proteins in \( X_{P-homo} \), and

\[ g(u,k) = \begin{cases} 1, & \text{if the k-th representative protein hits the u-th GO_compress number} \\ 0, & \text{otherwise} \end{cases} \]

(5)

Note that the GO feature extraction method may become a naught vector or meaningless under any of the following situations: (1) the protein P does not have significant homology to any protein in the Swiss-Prot database, i.e., \( X_{P-homo} = \emptyset \) meaning the homology set \( X_{P-homo} \) is an empty one; (2) its representative proteins do not contain any useful GO information for statistical prediction based on a given training dataset.

Under such a situation, let us consider using the dipeptide composition method to extract features for the protein P, as described below.

**DC (Dipeptide Composition)** Dipeptide composition represents the occurrence frequency of each two adjacent
amino acid residues. It is used to describe the global information about each protein sequence in the form of 420-dimensional (420-D) feature vector. An advantage of DC over amino acid composition is that it uses some sequence-order information. Dipeptide composition will generate 420 components for each protein sequence, the first 20 components are the conventional amino acid composition (AAC); the following 400 components are the fractions of 400 dipeptides, i.e. AA, AC, AD, ..., YV, YW, YY; the 400 components are calculated using the following equation

\[ \text{fraction of dip}(i) = \frac{\text{total number of dip}(i)}{\text{total number of all possible dipeptides}}, \]

where dip(i) is the i-th dipeptide of the 400 dipeptides, i=1, 2, ..., 400.

2.3. Prediction Algorithm: Ensemble of Classifier Chains

To enhance the success rate, the powerful ECC (Ensemble of Classifier Chains) classifier [78] was adopted to perform prediction. Below, let us briefly introduce the Ensemble of Classifier Chains classifier.

Consider the problem of classifying N proteins into M subcellular locations, which can be formulated as

\[ Y = \{\lambda_1, \lambda_2, ..., \lambda_M\}. \tag{7} \]

The available information is assumed to consist in a training dataset

\[ T = \{ (P_1, Y_1), (P_2, Y_2), ..., (P_N, Y_N) \}, \tag{8} \]

where the N proteins \( P_i \) (i = 1, 2, ..., N) and their corresponding class labels \( Y_i \) (i = 1, 2, ..., N) is a subset of \( Y \) of Eq. (7). This indicates that each protein may belong to two or more subcellular locations.

Binary relevance (BR) [79] is a popular approach to convert a multi-label learning problem into a number of independent binary classification ones. Specifically, a separate classifier \( h_\mu \) is learned for each class label \( \lambda_\mu \) in \( Y \), where each protein associated with class label set \( Y \) will be regarded as positive example when class label \( \lambda_\mu \in Y \) while regarded as negative example when class label \( \lambda_\mu \notin Y \). For the classification of a new protein, BR outputs the union of the class labels that are predicted by the \( M \) classifiers. BR is conceptually simple and easy to implement, whereas may be less effective since it don’t take label correlations into account.

However, in contrast to BR, individual classifiers of CC (Classifier Chain) have to be trained sequentially. Furthermore, classifiers are linked along a chain where each classifier is responsible for prediction of presence or absence of class label \( \lambda_\mu \in Y \). The feature space of each classifier in the chain is extended with the 0/1 label associations of all previous classifiers. In other words, a chain \( C_1, ..., C_M \) of binary classifiers is constructed. Each classifier \( C_\mu \) in the chain is responsible for predicting the binary association of class label \( \lambda_\mu \) given the feature space, incremented by all prior binary relevance associations in the chain \( \lambda_1, ..., \lambda_{\mu-1} \).

The chaining method passes label information between classifiers, allowing CC to take into account label correlations and thus overcoming the label independence problem of BR method. However, the order of the chain itself clearly has an effect on accuracy. In [78], the issue is solved by using an ensemble framework with a different random chain ordering for each iteration.

In contrast to the traditional single-label ensemble learning, ECC is an ensemble of multiple multi-label methods, i.e. the CC method. Following the typical strategy of ensemble learning, ECC also has two steps, in which the first is to train \( M \) CC classifiers \( C_1, C_2, ..., C_M \) and the second is to combine their predictions. In the first step, each \( C_k \) is trained with both a random chain ordering and a random subset of original training data set. In the second step, multi-label predictions of each \( C_k \) model are summed by label so that each label gets some votes, and then, we use a threshold to select the most possible labels which form the final multi-label prediction. Specifically, each \( C_k \) model predicts a vector \( y_k = (l^k_1, ..., l^k_M) \in \{0,1\}^M \). The sums are stored in a vector \( W = (l_1, ..., l_M) \in R^M \) such that \( l_j = \sum_{k=1}^{M} l^k_j \). Hence each \( l_j \in W \) represents the sum of the votes for the \( j \) th label.

We then normalize \( W \) to \( W^{\text{norm}} \), which represents a distribution of scores for each label in [0, 1]. A threshold is used to choose the final multi-label set \( Y \) such that \( \lambda_j \in Y \) where \( l_j \geq t \) for threshold \( t \). Here we simply set the threshold to be 0.5. Hence the relevant labels in \( Y \) represent the final multi-label prediction.

The entire predictor thus established is called Virus-ECC-mPLoc, which can be used to predict the subcellular localization of both singleplex and multiplex viral proteins. To provide an intuitive picture, a flowchart is provided in Fig. (1) to illustrate the prediction process of Virus-ECC-mPLoc.

3. RESULTS AND DISCUSSIONS

As mentioned in the above section, the benchmark dataset used in this study is X (cf. Online Supporting Information A), which is the same benchmark dataset constructed in [24] for iLoc-Virus.

To evaluate the proposed new prediction algorithm of this study, we compare it with iLoc-Virus [24]. Actually, for
such a dataset containing both single-location and multiple-location viral proteins distributed among 6 subcellular location sites, so far only two existing predictor, i.e., Virus-mPLoc [29] and iLoc-Virus [24], had the capacity to deal with it. Furthermore, iLoc-Virus outperforms Virus-mPLoc. Therefore, to demonstrate the power of the current predictor, it would suffice to just compare Virus-ECC-mPLoc with iLoc-Virus.

In statistical prediction, there are three commonly used methods, that is, the independent data set test, subsampling (e.g., K-fold cross validation) test, and jackknife test, which are often used for testing the accuracy of a statistical prediction method. Among the three methods, the jackknife test is deemed the most objective because it can always yield a unique result for a given benchmark data set, as elucidated in two comprehensive reviews [26, 80]. Therefore, the jackknife test has been increasingly and widely adopted by investigators to examine the power of various prediction methods. In the present study, we use both independent data set test and jackknife test to evaluate the power of Virus-ECC-mPLoc.

Table 2 and 3 report the detailed results on the 6 viral subcellular locations obtained with iLoc-Virus and Virus-ECC-mPLoc on the aforementioned benchmark dataset $X$ by the independent data set test and the jackknife test. For a fair algorithmic comparison between Virus-ECC-mPLoc and iLoc-Virus, we use the same GOA database that is described in this study to extract GO features. As we can see from Table 2 and 3, for such a stringent dataset, the overall success rate achieved by Virus-ECC-mPLoc is 86.9% for jackknife test which is about 5% higher than that by iLoc-Virus, while the overall success rate achieved by Virus-ECC-mPLoc is 87.2% for independent data set test which is about 6% higher than that by iLoc-Virus.

Note that during the process of the independent data set test and the jackknife test by iLoc-Virus and Virus-ECC-mPLoc, the false positives (over-predictions) and false negatives (under-predictions) were also taken into account to reduce the scores in calculating the overall success rate. As for the detailed process of how to count the over-predictions and under-predictions for a system containing both single-location and multiple-location proteins, see Eqs. 43-48 and Fig. (4) in a comprehensive review [26].

To provide a more intuitive and easier-to-understand measurement, let us introduce a new measure, the so-called “exact match” success rate, to reflect the accuracy of a predictor, as defined by

$$\Lambda = \frac{\sum_{i=1}^{N} \Delta(i)}{N},$$

where $\Lambda$ represents the exact match rate, $N$ the number of total proteins investigated, and
According to the above definition, for a protein belonging to, say, three subcellular locations, if only two of the three are correctly predicted, or the predicted result contains a location not belonging to the three, the prediction score will be counted as 0. In other words, when and only when all the subcellular locations of a query protein are exactly predicted without any underprediction or overprediction, can the prediction be scored with 1. Therefore, the exact match measure is much more strict and harsh than the measure used previously [24] in measuring the success rate. However, even if using such a stringent criterion on the same benchmark dataset, the overall exact match success rates achieved by Virus-ECC-mPLoc are 82.5% for jackknife test and 81.0% for independent data set test, which are about 9% and 6% higher than that by iLoc-Virus.

**CONCLUSION**

Prediction of protein subcellular localization is a challenging problem, particularly when the system concerned contains both singleplex and multiplex proteins. In this paper, we have proposed a novel multi-label predictor, called Virus-ECC-mPLoc, for predicting viral protein subcellular locations based on the powerful ECC algorithm and a hybrid of GO and DC feature extraction methods, which has been demonstrated very powerful for handling the multiplex proteins. Since user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful predictors [81], here we have provided a web-server for the method presented in this paper at http://levis.tongji.edu.cn:8080/bioinfo/Virus-ECC-mPLoc/.

**Table 2.** A Comparison of the Independent Data Set Test Success Rates by iLoc-Virus and the Current Virus-ECC-mPLoc on the Benchmark Dataset $X$ (cf. Online Supporting Information A) that Covers 6 Location Sites of Viral Proteins in which None of the Proteins Included has $\geq 25\%$ Pairwise Sequence Identity to any other in a Same Location

<table>
<thead>
<tr>
<th>Code</th>
<th>Subcellular Location</th>
<th>Success Rate by Independent data set test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>iLoc-Virus$^a$</td>
</tr>
<tr>
<td></td>
<td>Viral capsid</td>
<td>100.0%</td>
</tr>
<tr>
<td>1</td>
<td>Host cell membrane</td>
<td>82.3%</td>
</tr>
<tr>
<td>2</td>
<td>Host endoplasmic reticulum</td>
<td>76.8%</td>
</tr>
<tr>
<td>3</td>
<td>Host cytoplasm</td>
<td>78.3%</td>
</tr>
<tr>
<td>4</td>
<td>Host nucleus</td>
<td>88.1%</td>
</tr>
<tr>
<td>5</td>
<td>Secreted</td>
<td>71.5%</td>
</tr>
<tr>
<td>6</td>
<td>Overall</td>
<td>81.1%</td>
</tr>
</tbody>
</table>

$^a$ The predictor from [24].

$^b$ The predictor proposed in this paper.

**Table 3.** A Comparison of the Jackknife Success Rates by iLoc-Virus and the Current Virus-ECC-mPLoc on the Benchmark Dataset $X$ (cf. Online Supporting Information A) that Covers 6 Location Sites of Viral Proteins in which None of the Proteins Included has $\geq 25\%$ Pairwise Sequence Identity to any other in a Same Location

<table>
<thead>
<tr>
<th>Code</th>
<th>Subcellular location</th>
<th>Success rate by the jackknife test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>iLoc-Virus$^a$</td>
</tr>
<tr>
<td>1</td>
<td>Viral capsid</td>
<td>100.0%</td>
</tr>
<tr>
<td>2</td>
<td>Host cell membrane</td>
<td>78.8%</td>
</tr>
<tr>
<td>3</td>
<td>Host endoplasmic reticulum</td>
<td>75.0%</td>
</tr>
<tr>
<td>4</td>
<td>Host cytoplasm</td>
<td>79.3%</td>
</tr>
<tr>
<td>5</td>
<td>Host nucleus</td>
<td>88.1%</td>
</tr>
<tr>
<td>6</td>
<td>Secreted</td>
<td>75.0%</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>82.1%</td>
</tr>
</tbody>
</table>

$^a$ The predictor from [24].

$^b$ The predictor proposed in this paper.

$$
\alpha(i) = \begin{cases} 
1, & \text{if all the subcellular locations of the } i \text{-th protein are} \\
0, & \text{otherwise}
\end{cases}
$$

(10)
The current approach represents a new strategy to deal with the multi-label biological problems, and hence may become a useful vehicle in the area of bioinformatics and proteomics.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of China under the Grant No. 60873129 and 61005006, and the Open Projects Program of National Laboratory of Pattern Recognition in China.

SUPPLEMENTARY MATERIAL

Supplementary material is available on the publishers Web site along with the published article.

REFERENCES


