Gpos-PLoc: an ensemble classifier for predicting subcellular localization of Gram-positive bacterial proteins

Hong-Bin Shen¹ and Kuo-Chen Chou¹,²,³

¹Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, 1954 Hua-Shan Road, Shanghai 200030, China.
²Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, CA 92130, USA.
³To whom correspondence should be addressed. E-mail: kchou@san.rr.com

A statistical analysis indicated that, of the 35 016 Gram-positive bacterial proteins from the recent Swiss-Prot database, ~57% of these entries are without subcellular location annotations. In the gene ontology database, the corresponding percentage is ~67%, meaning the percentage of proteins without subcellular component annotations is even higher. With the avalanche of gene products generated in the post-genomic era, the number of such location-unknown entries will continuously increase. It is highly desired to develop an automated method for timely and accurately identifying their subcellular localization because the information thus obtained is very useful for both basic research and drug discovery practice. In view of this, an ensemble classifier called ‘Gpos-PLoc’ was developed for predicting Gram-positive protein subcellular localization. The new predictor is featured by fusing many basic classifiers, each of which was engineered according to the optimized evidence-theoretic K-nearest neighbors rule. As a demonstration, tests were performed on Gram-positive proteins among the following five subcellular location sites: (1) cell wall, (2) cytoplasm, (3) extracellular, (4) periplasm and (5) plasma membrane. To eliminate redundancy and homology bias, only those proteins which have <25% sequence identity to any other in the same subcellular location were allowed to be included in the benchmark datasets. The overall success rates thus achieved by Gpos-PLoc were >80% for both jackknife cross-validation test and independent dataset test, implying that Gpos-PLoc might become a very useful vehicle for expediting the analysis of Gram-positive bacterial proteins. Gpos-PLoc is freely accessible to public as a web-server at http://202.120.37.186/bioinf/Gpos/. To support the need of many investigators in the relevant areas, a downloadable file is provided at the same website to list the results identified by Gpos-PLoc for 31 898 Gram-positive bacterial protein entries in Swiss-Prot database that either have no subcellular location annotation or are annotated with uncertain terms such as ‘probable’, ‘potential’, ‘perhaps’ and ‘by similarity’. Such large-scale results will be updated once a year to include the new entries of Gram-positive bacterial proteins and reflect the continuous development of Gpos-PLoc.

Keywords: amphiphilic pseudo amino acid composition/fusion/gene ontology/Gram-positive/OET-KNN rule

Introduction

Bacteria are both harmful and useful to the environment, and animals, including humans. Therefore, the information about the proteins in the bacterial cell, such as their function and subcellular location, will be very useful for screening candidates in drug design, or selecting proteins for a special target. Bacteria can be divided into two groups: Gram-positive and Gram-negative. Gram-positive bacteria are those that are stained dark blue or violet by Gram staining; whereas Gram-negative bacteria cannot retain the stain, instead taking up the counter-stain and appearing red or pink.

An extensive study has been conducted in developing methods for predicting Gram-negative protein subcellular location (see, e.g. Nakai and Kanehisa, 1991; Nakai and Horton, 1999; Nakai, 2000; Gardy et al., 2003). However, so far very few reports have been seen for Gram-positive proteins in this regard.

According to the Swiss-Prot database (Bairoch and Apweiler, 2000), version 50.0 released on 30th May 2006, the number of total Gram-positive protein entries is 35 889. After excluding those annotated as ‘fragment’ or containing <50 amino acid residues, the number is reduced to 35 016, of which 15 236 are with subcellular location annotations (Item 1 of Table I). However, of the 15 236 proteins, 3118 are annotated with experimental observations (Item 2 of Table I) and 12 118 annotated with uncertain labels such as ‘probable’, ‘potential’, ‘perhaps’ and ‘by similarity’ (Item 3 of Table I). The uncertain annotations cannot be used as robust data for training a solid predictor. Actually, proteins with uncertain annotations also belong to the targets of identification either by newly developed predictors or by further experiments.

Such a gap would become even wider if a similar statistical analysis was conducted based on the gene ontology (GO) database (Ashburner et al., 2000), which was established according to molecular function, biological process and cellular component. As shown in Item 5 of Table I, of the 35 016 Gram-positive proteins, only 11 436/35 016 = 32.7% have GO annotations to indicate their subcellular components; i.e. the percentage for the proteins with subcellular location annotations in the GO database is even less than that of the Swiss-Prot database. Besides, since the GO database was derived from various other databases, including Swiss-Prot, the GO annotations might be contaminated by the uncertain information from the 12 118 entries as indicated in Item 3 of Table I.

Therefore, the number of Gram-positive proteins that have reliable subcellular location annotations is 3118 (Item 2 of Table I), which is about 9% of all the Gram-positive protein entries concerned. In other words, there are (35 016 − 3118) = 31 898 Gram-positive proteins whose subcellular locations need to be identified or further confirmed.
were excluded; also, sequences with were excluded. (5) Sequences annotated with 'fragment' location annotated with 'cytoplasm and plasma membrane' were used. (3) Sequences annotated with ambiguous or uncertain terms, such as 'potential', 'probable' and 'by similarity', were excluded. (4) Sequences annotated with uncertain terms, such as 'potential', 'probable', 'probably', 'maybe', 'likely' were used; in search for plasma membrane proteins, the key words 'cytoplasm' and 'cytoplasmic' were used; in search for extracellular proteins, the key words 'extracellular' and 'secreted' were used; in search for periplasm proteins, the key words 'periplasm' and 'periplasmic' were used; in search for periplasmic space, the key words 'periplasm' and 'periplasmic' were used. (2) Because a same subcellular location (-!-SUBCELLULAR LOCATION) in the CC field might be annotated with different terms, several key words might be used for a same subcellular location. For example, in search for cytoplasm proteins, the key words 'cytoplasm' and 'cytoplasmic' were used; in search for extracellular proteins, the key words 'extracellular', 'extracellular' and 'secreted' were used; in search for periplasm proteins, the keywords 'periplasm' and 'periplasmic' were used; in search for plasma membrane proteins, the key words 'plasma membrane', 'integral membrane', 'multi-pass membrane' and 'single-pass membrane' were used. (3) Sequences annotated with ambiguous or uncertain terms, such as 'potential', 'probable', 'probably', 'maybe', 'likely' or 'by similarity', were excluded. (4) Sequences annotated by two or more locations were not included because of lack of the uniqueness. For example, proteins with subcellular location annotated with 'cytoplasm and plasma membrane' were excluded. (5) Sequences annotated with 'fragment' were excluded; also, sequences with <50 amino acid residues were removed because they might just be fragments. (6) To avoid any homology bias, a redundancy cutoff was operated by a culling program (Wang and Dunbrack, 2003) to winnow those sequences which have ≥25% sequence identity to any other in a same subcellular location. After strictly following the above six criteria, we obtained 452 Gram-positive proteins, of which 14 belonged to cell wall, 196 to cytoplasm, 108 to extracellular, 5 to periplasm and 129 to plasma membrane (Fig. 1). It is instructive to point out here that, for quite a long time, it was thought by many that there was no periplasm in Gram-positive bacteria. However, with the technique of cryo-electron microscopy, the existence of a periplasmic space between the plasma membrane and the thick peptidoglycan layer of the gram-positive bacteria was indeed observed very recently (Zuber et al., 2006), further supporting the classification scheme as illustrated in Fig. 1. The 452 Gram-positive proteins thus obtained form a dataset \( S^0 \), which is a union of the following five subsets

\[
S^0 = S^0_1 \cup S^0_2 \cup S^0_3 \cup S^0_4 \cup S^0_5.
\]

On the basis of dataset \( S^0 \), two working datasets, i.e. a learning (training) dataset \( S^L \) and an independent testing dataset \( S^T \), were constructed. In order to fully use the data in \( S^0 \) and meanwhile guarantee that \( S^L \) and \( S^T \) be completely independent of each other, the following condition was imposed

\[
S^L \cup S^T = S^0 \quad \text{and} \quad S^L \cap S^T = \emptyset,
\]

where \( \cup \), \( \cap \) and \( \emptyset \) represent the symbols for 'union', 'intersection' and 'empty set' in the set theory, respectively. Protein samples in the corresponding subsets of \( S^L \) and \( S^T \) are randomly assigned according to the following 'bracket

Table I. Breakdown of the 35 016α Gram-positive protein sequence entries from Swiss-Prot database (version 50.0, released 30th May 2006) according to the nature of their subcellular location annotation and their expression in the GO database

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Number</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gram-positive proteins with subcellular location annotations in Swiss-Prot database</td>
<td>15 236</td>
<td>15 236/35 016 = 43.5</td>
</tr>
<tr>
<td>2</td>
<td>Proteins in Item (1) with subcellular locations clearly observed by experiments</td>
<td>3118</td>
<td>3118/35 016 = 8.9</td>
</tr>
<tr>
<td>3</td>
<td>Proteins in Item (1) with uncertain labels, such as 'potential', 'probable' and 'by similarity'</td>
<td>12 118</td>
<td>12 118/35 016 = 34.6</td>
</tr>
<tr>
<td>4</td>
<td>Proteins that have corresponding GO numbers as defined in GO database</td>
<td>33 197</td>
<td>33 197/35 016 = 94.8</td>
</tr>
<tr>
<td>5</td>
<td>Proteins with subcellular component annotations in GO database</td>
<td>11 436</td>
<td>11 436/35016 = 32.7</td>
</tr>
</tbody>
</table>

*The original Gram-positive bacterial protein entries were 35 889, of which 873 were either annotated as 'fragment' or with <50 amino acid residues, and hence were removed for further consideration.
predicting protein structural class (Chou and Cai, 2004a) and (FunD). The FunD-discrete model is particularly effective for is represented by the functional domain composition 2002; Cai et al. 2005a; Wang et al. 2005; Xiao et al., 2005; Cedano et al., 1997; Chou and Elrod, 1999; Zhou and Doctor, 2003). In the AA-discrete model, the sample of a protein is represented by a set of discrete numbers. The simplest one is the AA-discrete model in which the sample of a protein is represented by its amino acid composition (AA) (e.g. see Nakashima and Nishikawa, 1994; Cedano et al., 1997; Chou and Elrod, 1999; Zhou and Doctor, 2003). In the AA-discrete model, all the sequence-order effects are lost. To avoid completely lose the sequence-order information, the PseAA-discrete model was introduced (Chou, 2001) that can reflect the sequence-order information (at least partially) through a set of correlation factors called ‘pseudo amino acid component’ (PseAA), and the prediction quality has been remarkably improved (see e.g. Feng, 2002; Chou and Cai, 2003b; Shen and Chou, 2005a; Wang et al., 2005; Xiao et al., 2005; Zhang et al., 2006). The FunD-D model is the one (Chou and Cai, 2002; Cai et al., 2003) in which the sample of a protein is represented by the functional domain composition (FunD). The FunD-D model is particularly effective for predicting protein structural class (Chou and Cai, 2004a) and protease type (Chou and Cai, 2006).

The current prediction predictor is called Gpos-PLoc, which was established on the basis of two cornerstones: one is the GO-PseAA discrete model and the other the fusion OET-KNN (optimized evidence-theoretic K-nearest neighbors) operating engine. The former is for formulating the protein samples by hybridizing GO (Ashburner et al., 2000) and the amphiphilic PseAA (Chou, 2005), as detailed in Chou and Shen (2006); whereas the latter is a powerful ensemble classifier formed by fusing many basic individual classifiers each of which is engineered according to the OET-KNN rule (Denoeux, 1995; Keller et al., 1985).

One of the important advantages in using the GO-PseAA discrete model is that the protein samples mapped into the GO space will be clustered in a way distinctly correlated with their subcellular locations, so as to result in a high prediction quality even if the numbers of proteins in some of the training subsets are not as large as usually required. The procedures in using the GO-PseAA discrete model to represent a protein sample can be described as follows. Mapping UniProtKB/Swiss-Prot protein entries (Apweiler et al., 2004) to the GO database, one can get a list of data called ‘gene_association.goa_uniprot’, where each UniProtKB/Swiss-Prot protein entry corresponds to one or several GO numbers. In this study, such a data file was directly downloaded from ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UNIPROT/ (released on 4th March 2006). As shown in Table III, the relationships between the UniProtKB/Swiss-Prot protein entries (accession numbers) and the GO numbers may be one-to-many, reflecting the biological reality that a particular protein may function in several processes, contain domains that carry out diverse molecular functions, and participate in multiple alternative interactions with other proteins, organelles or locations in the cell (Ashburner et al., 2000). It can be seen from Table III that, for those proteins with ‘subcellular location unknown’ annotation in Swiss-Prot database, their corresponding GO numbers in GO database are also annotated with ‘cellular component unknown’ (e.g. proteins with accession numbers P82679, Q9RC23 and Q7S1D3), and that even for some proteins whose subcellular locations are clearly annotated in Swiss-Prot database, their corresponding GO numbers in GO database are annotated with ‘cellular component unknown’ (e.g. protein with accession number P00782).

Also, because the current GO database is not complete yet, some protein entries (such as ‘P0A5B7’, ‘O53077’ and ‘P0A5Q2’) have no corresponding GO numbers, i.e. no mapping records at all in the GO database, and hence are not included in the data list of gene_association.goa_uniprot.

Furthermore, the GO numbers do not increase successively and orderly. For easier handling, some reorganization and compression procedure was taken to renumber them. For example, after such a procedure, the original GO numbers GO:0000001, GO:0000002, GO:0000003, GO:0000004, GO:0000006, ..., GO:0051912 would become GO_compress:0000001, GO_compress:0000002, GO_compress:0000003, GO_compress:0000004, GO_compress:0000005, ..., GO_compress:0009918, respectively. The GO database thus obtained is called GO_compress database, whose dimensions were reduced from 51 912 in the original GO database to 9918. Each of the 9918 entities in the GO_compress database served as a base to define a protein sample.

**Table II.** Number of Gram-positive proteins in each of the five subcellular locations for the learning and testing datasets, respectively

<table>
<thead>
<tr>
<th>Subcellular location</th>
<th>Learning dataset $S^L$</th>
<th>Testing dataset $S^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Cell wall</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>(2) Cytoplasm</td>
<td>79</td>
<td>117</td>
</tr>
<tr>
<td>(3) Extracellular</td>
<td>61</td>
<td>47</td>
</tr>
<tr>
<td>(4) Periplasm</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>(5) Plasma membrane</td>
<td>65</td>
<td>64</td>
</tr>
<tr>
<td>Total</td>
<td>220</td>
<td>232</td>
</tr>
</tbody>
</table>

**Method**

The sequential model and discrete model are often used for predicting protein subcellular location. In the sequential model, the sample of a protein is represented by its amino acid sequence, and the sequence similarity search-based tools like BLAST are used to conduct prediction. However, this approach fails to work when the query protein does not have significant homology to proteins of known location. In the discrete model, the sample of a protein is represented by a set of discrete numbers. The simplest one is the AA-discrete model in which the sample of a protein is represented by its amino acid composition (AA) (e.g. see Nakashima and Nishikawa, 1994; Cedano et al., 1997; Chou and Elrod, 1999; Zhou and Doctor, 2003). In the AA-discrete model, all the sequence-order effects are lost. To avoid completely lose the sequence-order information, the PseAA-discrete model was introduced (Chou, 2001) that can reflect the sequence-order information (at least partially) through a set of correlation factors called ‘pseudo amino acid component’ (PseAA), and the prediction quality has been remarkably improved (see e.g. Feng, 2002; Chou and Cai, 2003b; Shen and Chou, 2005a; Wang et al., 2005; Xiao et al., 2005; Zhang et al., 2006). The FunD-D model is the one (Chou and Cai, 2002; Cai et al., 2003) in which the sample of a protein is represented by the functional domain composition (FunD). The FunD-D model is particularly effective for predicting protein structural class (Chou and Cai, 2004a) and protease type (Chou and Cai, 2006).
Unfortunately, the current GO numbers failed to give a complete coverage in the sense that some proteins might not belong to any of the GO numbers as mentioned above. Although the problem will gradually become trivial or eventually be solved with the continuous development of the GO database, to tackle such a problem right now, a hybridization approach was introduced by fusing the GO approach and the amphiphilic pseudo amino acid composition (PseAA) approach (Chou, 2005; Chou and Cai, 2005), as described below.

**Step 1:** Search a protein sample in the GO_compress database, if there is a hit corresponding to the \(i\)th GO_compress number, then the \(i\)th component of the protein in the 9918-D (dimensional) GO_compress space is assigned 1; otherwise, 0. Thus, the protein can be formulated as

\[
P = \begin{bmatrix} g_1 & g_2 & \cdots & g_i & \cdots & g_{9918} \end{bmatrix}^T,
\]

where \(T\) is the transpose operator, and

\[
g_i = \begin{cases} 1, & \text{hit found in GO_compress,} \\ 0, & \text{otherwise.} \end{cases}
\]

**Step 2:** If no hit (i.e. no record in the GO_compress database) is found whatsoever, then the protein should be defined in the \((20 + 2\lambda)\)-D amphiphilic PseAA space (Chou, 2005), as given below

\[
P = \begin{bmatrix} p_1 \\ \vdots \\ p_{20} \\ p_{20+1} \\ \vdots \\ p_{20+2\lambda} \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_{20} \\ p_{20+1} \\ \vdots \\ p_{20+2\lambda} \end{bmatrix}, \quad \Lambda = 20 + 2\lambda, \quad (6)
\]

where \(p_1, p_2, \ldots, p_{20}\) are associated with the amino acid composition reflecting the occurrence frequencies of the 20 native amino acids in the protein (Chou and Zhang, 1993, 1994), and \(p_{20+1}, p_{20+2}, \ldots, p_{20+2\lambda}\) are the 2\(\lambda\) correlation factors that reflect its sequence-order pattern through the amphiphilic feature. The protein representation as defined by equation (6) is called the ‘amphiphilic pseudo amino acid composition’ or PseAA, which has the same form as the conventional amino acid composition but contains more components and information. The components in equation (6) can be easily derived according to equations (2)–(6) of Chou (2005).

Suppose there are \(N\) proteins (\(P_1, P_2, \ldots, P_N\)) which have been classified into \(M\) subsets (subcellular locations). For the current case, we have \(M = 5\). Now, for a query protein \(P\), how can we identify which subset it belongs to? Subsequently we shall use the OET-KNN rule (Cover and Hart, 1967; Keller et al., 1985; Denoeux, 1995) to deal with this problem. The key of OET-KNN algorithm is to predict a query protein \(P\) belonging to the subset that has the highest evidence derived from the \(K\)-nearest neighbors of \(P\). For reader’s convenience, a brief introduction about OET-KNN
classifier and its key equations are given in Appendix 1. There are many different definitions to measure the ‘nearness’ for the OET-KNN classifier, such as Euclidean distance, Hamming distance (Mardia et al., 1979) and Mahalanobis distance (Mahalanobis, 1936; Pillai, 1985; Chou, 1995). Here, we use the following equation to measure the nearness between protein \( P \) and \( P_i \):

\[
\delta(P, P_i) = 1 - \frac{P \cdot P_i}{||P|| \cdot ||P_i||},
\]

where \( P \cdot P_i \) is the dot product of the two vectors, and \( ||P|| \) and \( ||P_i|| \) their modulus, respectively. According to equation (7), when \( P = P_i \) we have \( \delta(P, P_i) = 0 \), indicating the ‘distance’ between the two proteins is zero and hence they have perfect or 100% similarity.

Using the OET-KNN rule, the predicted result will depend on the selection of the parameter \( K \), the number of the nearest neighbors to the query protein \( P \). Because the predicted results by the OET-KNN algorithm (Cover and Hart, 1967; Keller et al., 1985; Denoeux, 1995) depend on the selection of parameter \( K \), hereafter we shall use OET-NN(\( K \)) to represent the symbol of OET-KNN, implying that the predicted result is the function of \( K \), the number of the nearest neighbors concerned for the query protein \( P \).

During the course of prediction, the following self-consistency principle should be followed. If a query protein could be defined in the 9918-D GO_compress space (equation (4)), then the prediction should be carried out based on those proteins in the training dataset that could be defined in the same 9918-D space. If the query protein in the 9918-D GO_compress space was a naught vector and hence must be defined instead in the \((20 + 2\lambda)\)-D or \(\Lambda\)-D PseAA space (equation (6)), then the prediction should be conducted according to the self-consistency principle that all the proteins in the training dataset be defined in the same \(\Lambda\)-D space as well. Accordingly, the current hybridization predictor actually consists of two sub-predictors: (a) the OET-NN(\( K \))-GO predictor that operates in the 9918-D GO_compress space, and (b) the OET-NN(\( K, \Lambda \))-PseAA predictor that operates in the \(\Lambda\)-D amphiphilic PseAA space. The former is the function of \( K \), whereas the latter the function of both \( K \) and \( \Lambda \). For a given learning dataset, selection of different \( K \) and \( \Lambda \) would result in different outcomes. To get the optimal success rate, one has to test the results by using different numbers of \( K \) and \( \Lambda \) one by one. However, it is both time-consuming and tedious to do so. To solve such a problem, the following two fusion processes are introduced for the OET-NN(\( K \)) and OET-NN(\( K, \Lambda \)) classifiers, respectively.

### One-dimensional fusion

It is for generating an ensemble classifier by fusing many individual basic OET-NN(\( K \)) classifiers each having a different specified value of \( K \), as formulated by

\[
\text{OET-NN}(K)^{\text{GO}} = [\text{OET-NN}(1)] \lor [\text{OET-NN}(2)] \lor \cdots \lor [\text{OET-NN}(\Omega)],
\]

where the symbol \( \lor \) denotes the fusing operator and OET-NN(\( K \)) the ensemble classifier formed by fusing OET-NN(1), OET-NN(2), \ldots, and OET-NN(\( \Omega \)). Here \( \Omega = 10 \) because preliminary tests indicated that the success rate obtained by the OET-NN(\( K \)) classifier trained by the current learning dataset was lower when \( K > 10 \).

The process of how the ensemble classifier OET-\( K \text{NN}(\text{GO}) \) works is as follows. Suppose the predicted classification results for the query protein \( P \) by the 10 individual classifiers in equation (8) are \( C_1, C_2, \ldots, C_{10} \) respectively; i.e.

\[
\{C_1, C_2, \ldots, C_{10}\} \in \{S_1, S_2, \ldots, S_5\},
\]

where \( \in \) is a symbol in the set theory meaning ‘element of’, and \( S_1, \ldots, S_5 \) represent the five subsets defined by the five subcellular locations studied here (Fig. 1), and the voting score for the protein \( P \) belonging to the \( k \)th subset is defined by

\[
Y^\text{GO}_k = \sum_{i=1}^{10} w_i \Delta(C_i, S_k) \quad (k = 1, 2, \ldots, 5),
\]

where \( w_i \) is the weight and was set at 1 for simplicity, and the delta function in equation (10) is given by

\[
\Delta(C_i, S_k) = \begin{cases} 1 & \text{if } C_i \in S_k, \\ 0 & \text{otherwise}, \end{cases}
\]

thus, the query protein \( P \) is predicted belonging to the subset (subcellular location) with which its score of equation (10) is the highest.

### Two-dimensional fusion

It is for generating an ensemble classifier by fusing many individual basic OET-NN(\( K, \Lambda \)) classifiers each having different specified values of \( K \) and \( \Lambda \). Owing to the similar reason as mentioned above in setting the value of \( \Omega \) for equation (8), let us consider \( K = 1, 2, \ldots, 10 \), and \( \Lambda = 20, 22, \ldots, 60 \); i.e.

\[
\{K\} = \{1, 2, \ldots, 10\}, \quad \{\Lambda\} = \{20, 22, \ldots, 58, 60\}.
\]

Thus, the ensemble classifier obtained by the two-dimensional fusion process can be formulated as

\[
\text{OET-NN}(\text{Pse}) = \left[\text{OET-NN}(1,20)\lor\text{OET-NN}(1,22)\lor\cdots\lor\text{OET-NN}(10,60)\right],
\]

where the fusion operator \( \lor \) has the same meaning as that of equation (8). The detailed process of how the ensemble classifier OET-\( K \text{NN}(\text{Pse}) \) works is as follows. Suppose the predicted classification results for the query protein \( P \) by the \( 10 \times 21 = 210 \) individual classifiers in equation (13) are

\[
\begin{bmatrix}
C_{1,20} & C_{1,22} & \cdots & C_{1,60} \\
C_{2,20} & C_{2,22} & \cdots & C_{2,60} \\
\vdots & \vdots & \ddots & \vdots \\
C_{10,20} & C_{10,22} & \cdots & C_{10,60}
\end{bmatrix} \in \{S_1, S_2, \ldots, S_5\},
\]
where $S_1$, $S_2$, $S_3$, $S_4$, $S_5$ have the same meanings as in equation (9), i.e. represent the five subsets defined by the five subcellular locations studied here (Fig. 1), and the voting score for the query protein $P$ belonging to the $k$th subset is defined by

$$Y^\text{Pse} P_k = \sum_{i=1}^{10} \sum_{j=1}^{30} w_{i,j} \Delta(C_{i,j}, S_k) \quad (k = 1, 2, \ldots, 5), \quad (15)$$

where $w_{i,j}$ is the weight and was set at 1 for simplicity, the delta function in equation (15) is given by

$$\Delta(C_{i,j}, S_k) = \begin{cases} 1 & \text{if } C_{i,j} \in S_k, \\ 0 & \text{otherwise}, \end{cases} \quad (16)$$

thus, the query protein $P$ is predicted belonging to the subset (subcellular location) for which its score of equation (16) is the highest.

Finally, it should be pointed out that, although using GO database to predict protein subcellular location has been explored by previous investigators (Chou and Cai, 2003a, 2004b), the predictors formulated there has much less power than the current predictor owing to the following reasons. (a) The GO approach in Chou and Cai (2003a, 2004b) was operated by the nearest neighbor rule with $K = 1$ only, which is much less powerful than ensemble classifier as formulated in equations (8) and (13). (b) The dimension of the GO database space in Chou and Cai (2003a, 2004b) is 1930, but the dimension of GO database space here is 9918, indicating the need to catch up with the rapid development in GO. Besides, it is through Tables I and III presented here that the relationship between GO and Swiss-Prot is more clearly elucidated than in the previous papers (Chou and Cai, 2003a, 2004b).

**Results and discussion**

For the proteins listed in the Online Supplementary Materials A and B, we obtained the following results according to Steps 1–2 of Methods section: (a) of the 220 Gram-positive proteins in the learning dataset, 211 got hits in the GO_compress database, and hence were defined in the 9918-D GO_compress space (equations (4) and (5)), and the remaining 9 proteins defined in the A-D PseAA space (equation (6)); (b) of the 232 proteins in the testing dataset, 223 got hits and were defined in the 9918-D GO_compress space, and the remaining 9 protein was defined in the A-D PseAA space. Although for the benchmark datasets studied here, the number of proteins that could be meaningfully defined in the 9918-D GO_compress space was overwhelming, this does not mean that there is no need to include the OET-$\text{NN}^\text{Pse}$ predictor because, as shown in Table I, currently there still are $\sim$5.4% of Gram-positive proteins that have no any corresponding GO number. Therefore, in practical application, cases do exist where the query proteins may not be meaningfully defined in the GO system.

Although such a problem will be eventually solved with the continuous development of the GO database, it would be harmless and makes the predictor more complete to keep the OET-$\text{NN}^\text{Pse}$ classifier in the system since the prediction process is logically operated according to the following criterion or hierarchy: if a query protein can be defined in the 9918-D GO_compress space, then the classifier OET-$\text{NN}^\text{GO}$ is used to predict its subcellular location; otherwise, the classifier OET-$\text{NN}^\text{Pse}$ is used.

The prediction quality was tested by jackknife cross-validation and independent dataset validation. The jackknife test is thought one of the most rigorous and objective methods for cross-validation in statistics (see Chou and Zhang, 1995 for a comprehensive review) and have been increasingly used by investigators (Zhou, 1998; Feng, 2001; Zhou and Assa-Munt, 2001; Feng, 2002; Luo et al., 2002; Liu et al., 2005; Wang et al., 2005; Guo et al., 2006; Sun and Huang, 2006; Wen et al., 2006; Xiao et al., 2006; Zhang et al., 2006) in examining the accuracy of various prediction methods. Therefore, the power of a predictor should be measured by the success rate of jackknife test. The independent dataset test performed here was just for a demonstration of practical application.

The success rates obtained by jackknife and independent dataset tests for each of the 5 Gram-positive bacterial protein subcellular localization sites are given in Table IV, from which we can see that, for those subcellular locations with $\sim$60 protein samples, the success rates are quite high, and that the overall success rates by the jackknife test and independent dataset test are 82.7% and 84.1%, respectively. Therefore, it is expected that, with more data available to improve the learning dataset, particularly the subsets with $<20$ protein samples, the success rates will be further enhanced.

**Conclusion**

With the explosion of newly found protein sequences entering into protein databanks, prediction of protein subcellular locations has become increasingly important. In this article, the GO-discrete model was introduced to represent the sample of a protein. On the basis of such a frame of representation, the ensemble classifier Gpos- PLoc was developed for predicting the subcellular location of Gram-positive bacterial proteins. Gpos-PLoc was formed by fusing many basic classifiers with each engineered by the OET-KNN rule. The high success rates indicate that proteins, if represented

---

**Table IV.** The success rates by jackknife and independent dataset tests for each of the five Gram-positive bacterial protein subcellular localization sites (Fig. 1).

<table>
<thead>
<tr>
<th>Subcellular localization</th>
<th>Jackknife test$^a$</th>
<th>Independent dataset test$^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Cell wall</td>
<td>6/11 = 54.6%</td>
<td>1/3 = 33.3%</td>
</tr>
<tr>
<td>(2) Cytoplasm</td>
<td>63/79 = 79.8%</td>
<td>94/117 = 80.3%</td>
</tr>
<tr>
<td>(3) Extracellular</td>
<td>49/61 = 80.3%</td>
<td>36/47 = 76.6%</td>
</tr>
<tr>
<td>(4) Periplasm</td>
<td>2/4 = 50.0%</td>
<td>0/1 = 0.0%</td>
</tr>
<tr>
<td>(5) Plasma membrane</td>
<td>62/65 = 95.4%</td>
<td>64/64 = 100%</td>
</tr>
<tr>
<td>Overall</td>
<td>18/220 = 82.7%</td>
<td>195/232 = 84.1%</td>
</tr>
</tbody>
</table>

$^a$Jackknife cross-validation test was performed on the 220 proteins of S$^1$ in the Online Supplementary Material A, where none of proteins has $\geq$ 25% sequence identity to any other in the same subcellular location.

$^b$Prediction was performed on the 232 independent proteins of S$^5$ in the Online Supplementary Materials B using the rule parameters derived from the 220 proteins of S$^1$ in the Online Supplementary Material A. None of proteins in S$^1$ has $\geq$25% sequence identity to any others in the same subcellular location within either S$^1$ or S$^5$; or vice versa.
Appendix I: The optimized evidence-theoretic K-nearest neighbors (OET-KNN) classifier

For reader’s convenience, a brief introduction of the OET-KNN classifier is given below. For further explanation, refer to Shen and Chou (2005b). Let us consider a problem of classifying $N$ entities into $M$ classes (subcellular locations), which can be formulated as

$$F = \{\Phi_1, \Phi_2, \ldots, \Phi_\mu, \ldots, \Phi_M\}.$$  \hspace{1cm} (A1)

The available information is assumed to consist in a training dataset

$$\mathcal{N} = \{(P_1, \theta_1), \ldots, (P_N, \theta_N)\}$$ \hspace{1cm} (A2)

where the $N$ entities $P_i (i = 1, 2, \ldots, N)$ and their corresponding pattern (class) labels $\theta_i (i = 1, 2, \ldots, N)$ take values in $F$ of equation (A1). According to the KNN (K-nearest neighbors) rule (Cover and Hart, 1967), an unclassified entity $P$ is assigned to the class represented by a majority of its K-nearest neighbors of $P$. Owing to its good performance and simple-to-use feature, the KNN rule, also named as ‘voting KNN rule’, is quite popular in pattern recognition and classification.

The ET-KNN (evidence theoretic K-nearest neighbors) rule is a pattern classification method based on the Dempster–Shafer theory of belief functions (Denoeux, 1995). In the classification process, each neighbor of a pattern under consideration and aggregated using the Dempster’s rule of combination (Shafer, 1976). A decision is made by assigning a pattern to the class with which the belief or credibility function of equation (A6) has the maximum value; i.e. if

$$\mathbb{R}(P, \Phi_\mu) = \max \{\mathbb{R}(P, \Phi_1), \mathbb{R}(P, \Phi_2), \ldots, \mathbb{R}(P, \Phi_M)\}.$$  \hspace{1cm} (A7)

where $\mu = 1, 2, \ldots, M$ and the operator Max means taking the maximum one among those in the brackets, then the class $\Phi_\mu$ is the class predicted for the query protein $P$.

References


