Amino Acids (2009) 37:427–433
DOI 10.1007/s00726-008-0172-0

ORIGINAL ARTICLE

A complexity-based method for predicting protein subcellular location

Xiaoqi Zheng · Taigang Liu · Jun Wang

Received: 30 May 2008 / Accepted: 4 August 2008 / Published online: 22 August 2008

Abstract A complexity-based approach is proposed to predict subcellular location of proteins. Instead of extracting features from protein sequences as done previously, our approach is based on a complexity decomposition of symbol sequences. In the first step, distance between each pair of protein sequences is evaluated by the conditional complexity of one sequence given the other. Subcellular location of a protein is then determined using the *k*-nearest neighbor algorithm. Using three widely used data sets created by Reinhardt and Hubbard, Park and Kanehisa, and Gardy et al., our approach shows an improvement in prediction accuracy over those based on the amino acid composition and Markov model of protein sequences.

Keywords Protein subcellular location · Symbol sequence complexity · *k*-Nearest neighbor algorithm · Jackknife analysis

Introduction

Information on subcellular localization of proteins is important to molecular cell biology, proteomics, system biology and drug discovery (Chou and Shen 2008). Knowing the compartment in which a protein resides may give important insights as to its structure, function and interacting network in cellular metabolism. As currently practiced, experimental determination of subcellular location is mainly accomplished by the three approaches: cell fractionation, electron microscopy and fluorescence microscopy. However, these approaches are time consuming, subjective and highly variable (Murphy et al. 2000). In recent years, a great deal of automated methods to quickly identify the protein subcellular location from sequence data have emerged. These methods can be grouped into three categories. The first category is based on the existence of some sorting signals (Nakai 2000), which include signal peptides, mitochondrial targeting peptides and chloroplast transit peptides (Emanuelsson et al. 2000; Nielsen et al. 1997, 1999). Nevertheless, the performances of these methods are highly dependent on the quality of the N-terminal sequence assignment (Hua and Sun 2001). The second category investigates the whole sequence information such as the amino acid composition (Nakashima and Nishikawa 1994; Cedano et al. 1997; Chou and Elord 1999; Reinhardt and Hubbard 1998), dipeptide (Huang and Li 2004), gapped amino acid pairs (Park and Kanehisa 2003; Guo et al. 2005) and so on. These kinds of algorithms also suffer from the drawback that there is a loss of sequence order information. The third category fuses the results from different modules each of which extracts a particular feature from proteins. They integrate signal peptide information or whole sequence information with other features such as physical and chemical properties (Chou 2001; Feng and Zhang 2002), protein domain information (Chou and Cai 2002, 2003, 2004) and *n*-gram (Yu et al. 2004), etc.

All methods mentioned above need first to extract features from protein sequences, and then use some machine

X. Zheng
Department of Applied Mathematics,
Dalian University of Technology, 116024 Dalian, China

T. Liu · J. Wang
College of Advanced Science and Technology,
Dalian University of Technology, 116024 Dalian, China

J. Wang (✉)
Department of Mathematics, Shanghai Normal University,
200034 Shanghai, China
e-mail: junwang@dlut.edu.cn

learning approaches to identify their locations. The performance of these methods relies heavily on the sensitivity and selectivity of the corresponding feature vectors. But it is known that when protein sequences are decomposed into the amino acid composition, dipeptide composition or some other feature descriptors, much information for prediction is lost (especially the order information). Hence, it is expected that a higher accuracy would be gained when predicting the subcellular locations directly from sequence data.

In the present study, we bypass the process of feature extraction and only make use of a conditional complexity measure of symbol sequences. There are several methods to evaluate the intrinsic complexity of a symbol sequence, e.g., entropy measure (Sadovsky 2003), compositional complexity (Bernaola-Galván et al. 1999), linguistic complexity (Troyanskaya et al. 2002) and Lempel–Ziv (LZ) complexity (Ziv and Lempel 1977, 1978; Lempel and Ziv 1976). Of the known complexity measures, the LZ complexity measure is the most adequate one in reflecting the repeated patterns occurring in the text, and hence is adopted in this study. This kind of complexity measure was first used in protein subcellular prediction by Xiao et al. (2005), who used it as a component of pseudo amino acid composition vector, and later investigated by Diao et al. (2008). In this study, instead of comparing the corresponding sequence features, we use the conditional complexity of one sequence relative to the other sequence as a measure of distance. Then *k*-nearest neighbor (*k*-NN) algorithm is used to identify the subcellular location of a test protein given a set of training sequences.

Materials and methods

Sequence data

We use three data sets to examine the performance of our method. The first was generated by Reinhardt and Hubbard (1998). This data set was widely used to examine performances of different prediction methods (Yuan 1999; Hua and Sun 2001; Huang and Li 2004; Gao et al. 2005). It was taken from SWISS-PROT release 33.0 and only included globular proteins, because the transmembrane proteins were predicted with a much higher accuracy (Boyd et al. 1998). Redundancy in this data set was reduced such that none had >90% sequence identity to any other in the set. As is shown in Table 1, this data set has 2,427 protein sequences from eukaryotic species classified into four location groups, cytoplasmic, extracellular, nuclear and mitochondrial, and 997 prokaryotic sequences are assigned to three location categories, cytoplasmic, extracellular and periplasmic. The second widely used data set was constructed by Park and Kanehisa (2003) (Table 2). This data set was extended to 12 subcellular locations: chloroplast, cytoplasmic, cytoskeleton, endoplasmic reticulum, extracellular, Golgi apparatus, lysosomal, mitochondrial, nuclear, peroxisomal, plasma membrane and vacuolar protein. Any two sequences in this data set has ≤80% sequence identity. The third data set was built by Gardy et al. (2003), extracted from SWISS-PROT release 40.29. This data set consists of 1,443 protein sequences: 1,302 proteins localized in a single subcellular site, which are 248 cytoplasmic, 268 inner membrane, 244 periplasmic, 352 outer membrane, and 190 extracellular. This data set also includes a further 141 proteins resident at multiple localization sites: 14 cytoplasmic/inner membrane, 50 inner membrane/periplasmic, and 77 outer membrane/extracellular.

Table 1 Number of sequences within each subcellular localization category of the first dataset (Reinhardt and Hubbard 1998)

Species	Subcellular location	Number of sequences
Prokaryotic	Cytoplasmic	688
	Periplasmic	202
	Extracellular	107
Eukaryotic	Nuclear	1,097
	Cytoplasmic	684
	Mitochondrial	321
	Extracellular	325

Table 2 Number of sequences within each subcellular localization category of the second data set (Park and Kanehisa 2003)

Subcellular location	No. of entries
Chloroplast	671
Cytoplasmic	1,241
Cytoskeleton	40
Endoplasmic reticulum	114
Extracellular	861
Golgi apparatus	47
Lysosomal	93
Mitochondrial	727
Nuclear	1,932
Peroxisomal	125
Plasma membrane	1,674
Vacuolar	54
Total	7,579

LZ complexity of symbol sequences and pairwise distance measure

LZ complexity, proposed by Lempel and Ziv to measure the randomness of finite sequences, is an easily computable

and universal depiction of sequence complexity (Lempel and Ziv 1976; Ziv and Lempel 1977, 1978). This complexity measure is related to the number of distinct substrings (i.e., patterns) and the rate of their occurrence along a given sequence (Lempel and Ziv 1976). In resent years, LZ decomposition and LZ complexity of symbol sequences have been applied extensively in biological data analysis, e.g., phylogenetic tree reconstruction (Otu and Sayood 2003), recognition of structural regularities and classification of protein structures (Orlov and Potapov 2004; Wang and Zheng 2008).

For symbol sequences S, T and R defined over a finite alphabet $\mathcal{A}$, let $L(S)$ be the length of S, $S(i)$ be the ith element of S and $S(i, j)$ be the subsequence of S that starts at position i and ends at position j. Let S be the concatenation of T and R, i.e., $S = TR$. Here, T is called a prefix of S and S is called an extension of T.

An extension $S = TR$ of T is called *reproducible* (denoted $T \rightarrow S$), if there exists an integer $m \leq L(T)$ such that $R(k) = S(m + k-1)$, for $k = 1,2,...,L(R)$. For example EWRRA→EWRRAWRRA with $m = 2$, and AAQHQG → AAQHQGHQ with $m = 4$. Similarly, an extension $S = TR$ of T is called *producible* (denoted $T \Rightarrow S$) if $T \rightarrow S(1,L(S)-1)$. That is to say, an extra 'different' symbol at the end of the producible extension is allowed. For example, AAQHQG ⇒ AAQHQGHQW. Thus if $T \rightarrow S$ then $T \Rightarrow S$, but the reverse is not always true.

According to the above definitions, any non-null sequence S can be built from the null sequence by iterative "producible extensions", i.e., $\varepsilon \Rightarrow S(1, h_1) \Rightarrow S(1,h_2) \Rightarrow \ldots \Rightarrow S(1, h_r)$, where $h_1 = 1$ and $h_r = L(S)$. This r-step production process of S leads to a parsing of S into such decomposition: $H(S) = S(1, h_1)\cdot S(h_1 + 1,h_2)...S(h_{r-1} + 1, h_r)$, where $H(S)$ is called the *history* of S and $H_i(S) = S(h_{i-1} + 1, h_i)$ is called the ith component of $H(S)$. Furthermore, if $S(1, h_{i-1}) \rightarrow S(1, h_i)$ is not true, the component $H_i(S)$ is called *exhaustive*. A history $H(S)$ is called exhaustive if all of its components are exhaustive (with a possible exception of the last one). Taking S=AAQHQGHQW as an example, two production processes of S are as follows

$$\epsilon \Rightarrow \mathrm{A} \Rightarrow \mathrm{AAQ} \Rightarrow \mathrm{AAQH} \Rightarrow \mathrm{AAQHQG} \Rightarrow \mathrm{AAQHQGHQ} \Rightarrow \mathrm{AAQHQGHQW}, \quad (1)$$

$$\epsilon \Rightarrow \mathrm{A} \Rightarrow \mathrm{AAQ} \Rightarrow \mathrm{AAQH} \Rightarrow \mathrm{AAQHQG} \Rightarrow \mathrm{AAQHQGHQW}. \quad (2)$$

The corresponding decompositions are

$$H_1(S) = \mathrm{A}\cdot\mathrm{AQ}\cdot\mathrm{H}\cdot\mathrm{QG}\cdot\mathrm{HQ}\cdot\mathrm{W}, \text{ and}$$

$$H_2(S) = \mathrm{A}\cdot\mathrm{AQ}\cdot\mathrm{H}\cdot\mathrm{QG}\cdot\mathrm{HQW}.$$

However, $H_1(S)$ is not exhaustive because the second to last extension AAQHQG ⇒ AAQHQGHQ in Eq. 1 is also reproducible (i.e., AAQHQG → AAQHQGHQ). While $H_2(S)$ is exhaustive as all its extensions (except for the last one) are producible instead of reproducible. It is easy to declare that the exhaustive history of any non-null sequence is unique. Define the LZ complexity of a sequence S [denote by $c(S)$] to be the number of components in the exhaustive history of S. In the above example, $c(S) = 5$. For $S' = \mathrm{EWRRAWRRAEW}$, the corresponding exhaustive history is $H(S') = \mathrm{E}\cdot\mathrm{W}\cdot\mathrm{R}\cdot\mathrm{RA}\cdot\mathrm{WRRAE}\cdot\mathrm{W}$, so $c(S') = 6$.

To evaluate the distance between sequences S and T, one can decompose one sequence treating "information" in the other sequence as free, i.e., when parsing T, we treat its substrings which occur in S as having occurred in T. Accordingly, a dissimilarity measure can be defined as

$$\mathrm{Dissim}(S, T) = c(ST) - c(S).$$

Note that $Dissim(S,T) \neq Dissim(T,S)$. To ensure the symmetry condition and eliminate the effect of different sequence lengths, the final distance between S and T is defined as

$$d(S, T) = \frac{\max\{\mathrm{Dissim(S, T)}, \mathrm{Dissim(T, S)}\}}{\max\{c(S), c(T)\}}.$$

According to above definitions, $c(SS) = c(S)$ or $c(S) + 1$. So the distance d satisfies the identity condition up to a small error term.

The k-nearest neighbor algorithm

Protein subcellular localization prediction is a multi-class classification problem. Given a test sequence, its location is distinguished by a set of training data. In the present work, we use the k-nearest neighbor (k-NN) algorithm, a simple non-parametric classification algorithm (Duda et al. 2000). Given a test protein S of unknown category, this algorithm first finds the k-nearest neighbors in the training set $\{S_i\}$ ($i = 1,2,...,N$), where N is the number of training sequences. Then it assigns a prediction label to the test sample S according to the categories of its neighbors. For example, we have four categories, C_1, C_2, C_3 and C_4. Denote the number of k-NN in each category by N_1, N_2, N_3 and N_4, respectively. If

$$N_1 = \max\{N_1, N_2, N_3, N_4\},$$

the sample S should be classified to the category C_1. If there are two or more maximum numbers, one can compare their relative orders in the k-NN. Combining the k-NN algorithm with fuzzy set theory, this yields the fuzzy k-NN method. This method assigns fuzzy memberships of samples to different categories rather than a particular class as in 'k-NN', and can often improve classification performance (Bezdek et al. 1993; Leszczynski et al. 1999).

Despite its simplicity, the k-NN algorithm (and fuzzy k-NN algorithm) can give competitive performance

compared to many other methods, and it is robust to noisy training data. However, this method has the deficiency of high computation cost because we need to compute distance of each query instance to all training samples.

Jackknife test and accuracy

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, subsampling test, and jackknife test (Chou and Zhang 1995). However, as demonstrated by Chou and Shen (2007), the jackknife test has the least arbitrariness. A brief yet convincing elucidation in this regard was also given by Chou and Shen (2008), Mardia et al. (1979) . Therefore, we use jackknife test for cross validation. During the process of jackknife testing, each protein is singled out in turn as a test sample, the remaining proteins are used as training set to calculate test sample's membership and predict the category. After identifying the location of each protein using k-NN algorithm, the prediction accuracy for the category C is

$$\text{Accuracy}(C) = \frac{p(C)}{|C|},$$

and the total accuracy is

$$\text{Total accuracy} = \frac{\sum_C p(C)}{N},$$

where $|C|$ is the number of proteins in category C, $N = \sum_C |C|$ is the total number of sequences, and $p(C)$ is the number of properly predicted proteins in C. However, the above total accuracy depends heavily on the location of groups with large numbers of entries. In order to evaluate performances of small groups equally important to those of large groups, Park and Kanehisa (2003) defined the location accuracy as follows

$$\text{Location accuracy} = \frac{\sum_C \text{Accuracy(C)}}{M},$$

where M is the number of subcellular locations.

As another widely used measure of prediction accuracy, Matthew's correlation coefficient (MCC) provides a single measure of evaluating specificity and sensitivity together, where it equals one for perfect predictions and zero for random assignments (Matthews 1975).

$$MCC(C) = \frac{p(C)n(C) - u(C)o(C)}{\sqrt{(p(C)+u(C))(p(C)+o(C))(n(C)+u(C))(n(C)+o(C))}},$$

where $n(C)$ is the number of correctly predicted proteins not in location C, $u(C)$ is the number of underpredicted and $o(C)$ is the number of overpredicted proteins.

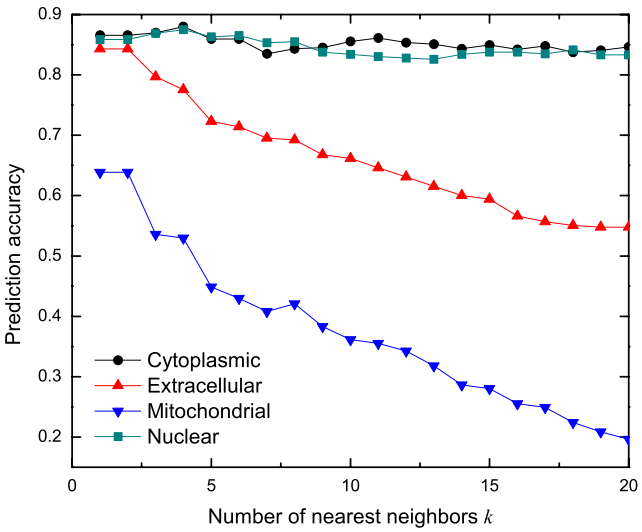

Fig. 1 Overall predictions for different subcellular locations at different values of k using our method. The prediction accuracies of cytoplasmic and nuclear proteins reach the maximum value at $k = 4$, but the overall maximum accuracy is reached at $k = 1$ and 2

Results

Tests have been done with various values of nearest neighbors k (from 1 to 20), and prediction accuracies for the first data set are shown in Fig. 1. As is shown, for different values of k, prediction accuracy of cytoplasmic proteins is similar to that of nuclear proteins, and their common maximum value (88.0 and 87.5% for cytoplasmic and nuclear proteins, respectively) is achieved at $k = 4$. But on the whole, their prediction accuracy does not vary significantly with the increase of k. While for extracellular and mitochondrial proteins, the proportion of successful prediction presents a significant decrease with the increase of k, i.e., from 84.3 to 54.8% for extracellular proteins and from 63.9 to 19.6% for mitochondrial proteins. One possible reason for this phenomenon is finite set effect. Small sample sets, extracellular and mitochondrial proteins in the first data sets, are more prone to be affected by the noise of large k. In summary, the overall accuracy of prediction for all protein locations achieves the maximum value at $k = 1$ and 2 (Note that our method produces the same result at $k = 1$ and 2).

In order to examine the performance of our method, we made comparisons with some classical methods on the same data set (Table 3). These are the neural network method (Reinhardt and Hubbard 1998), which is based on the amino acid composition of protein sequences, Markov chain model (Yuan 1999), support vector machine (SVM) (Hua and Sun 2001) and Fuzzy k-NN method (Huang and Li 2004), which is based on frequencies of dipeptides (denoted by Dipeptide-FKNN in Table 4). From Table 4

Table 3 Overall success prediction rates for the first data set by different algorithms and test methods

Location	Neural network[a] accuracy (%)	Markov model[b]		SVM[a]		Dipeptide-FKNN[c]		Our method	
		Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy(%)	MCC	Accuracy (%)	MCC
Cytoplasmic	55	78.1	0.60	76.9	0.64	86.7	0.76	86.5	0.73
Extracellular	75	62.2	0.63	80.0	0.78	83.7	0.87	84.3	0.85
Mitochondrial	61	69.2	0.53	56.7	0.58	60.4	0.63	63.9	0.62
Nuclear	72	74.1	0.68	87.4	0.75	92.0	0.83	85.9	0.76
Total accuracy	66	73.0	–	79.4	–	85.2	–	82.9	–
Location accuracy	65.8	70.9	–	77.8	–	80.7	–	80.2	–

[a] Based on the amino acid composition (Reinhardt and Hubbard 1998; Hua and Sun 2001)

[b] Based on a Markov model of amino acid sequences (Yuan 1999)

[c] Based on the frequencies of dipeptides (Huang and Li 2004).

Table 4 Comparison of our method with the SVM-based method using the second data set

Subcellular location	SVM[a]	Our method	
		k-NN[b]	Fuzzy k-NN[c]
Chloroplast	72.3	87.0	86.4
Cytoplasmic	72.2	80.5	81.6
Cytoskeleton	58.5	80.0	77.5
Endoplasmic reticulum	46.5	79.8	78.9
Extracellular	78.0	83.3	84.0
Golgi apparatus	14.6	63.8	61.7
Lysosomal	61.8	74.2	73.1
Mitochondrial	57.4	63.7	62.9
Nuclear	89.6	79.0	84.4
Peroxisomal	25.2	66.4	62.4
Plasma membrane	92.2	75.1	86.7
Vacuolar	25.0	64.8	66.7
Total accuracy (%)	78.2	77.7	81.6
Location accuracy (%)	57.9	74.8	75.5

[a] Using dipeptide frequency as sequence feature

[b] Results are got at $k = 1$

[c] Results are got at $k = 13$ and $m = 1.04$

we can see that the total prediction accuracy of our method is 82.9%, which is higher than that of the neural network method (66%), Markov chain model (73%) and support vector machine (79.4%). However, this prediction accuracy is slightly less than that using Fuzzy k-NN method—our method works better for mitochondrial class, but achieves worse results for nuclear proteins. Note that compared to the nuclear class, the mitochondrial class has less number of proteins, so it is relatively harder to predict. In other words, our complexity-based method has better prediction potential for relatively "hard" cases, when the training set is limited. Moreover, in contrast to the Fuzzy k-NN method, our approach is fully automatic, i.e., it has no free parameter and does not need to extract features from protein sequences. The above comparisons indicate that our method has reached a satisfactory performance despite its simplicity.

Prediction accuracy for the second data set is shown in Table 4. As can be seen, the total accuracy of our method is similar to that of the SVM method. Compared to the SVM-based method, our method performs better at nearly all subcellular locations except for nuclear and plasma membrane proteins. Especially for some small size classes, e.g., endoplasmic reticulum (from 46.5 to 79.8%), Golgi apparatus (from 14.6 to 63.8%), peroxisomal (from 25.2 to 66.4%) and vacuolar (from 25.0 to 64.8%). As previously stated, these small size classes are much harder to predict compared to the two large classes (nuclear and plasma membrane proteins). But for the location accuracy, our method shows a significant improvement, i.e., from 57.9% (SVM) to 74.8% (k-NN). In order to increase the prediction accuracy, Fuzzy k-NN algorithm at different values of k and fuzzy strength parameter m was tried on the second data set. At $k = 13$ and fuzzy strength parameter $m = 1.04$, our method gets the maximum prediction accuracy (Table 4). As can be seen, our method based on fuzzy k-NN algorithm achieves a significant improvement at both total accuracy (81.6%) and location accuracy (75.5%).

In Table 5, we compare the performances of our method and three widely used predictive tools, i.e., PSORT I (Nakai and Kanehisa 1991), PSORT-B (Gardy et al. 2003) and SubLoc (Hua and Sun 2001), for the third data set. The maximum prediction accuracy of our method is got for fuzzy k-NN algorithm at $k = 33$ and fuzzy strength parameter $m = 1.06$. As can be seen in this table, the overall prediction accuracy of our method reaches 79.8%, which is 18.9% higher than that of PSORT I, 5.0% higher than that of PSORT-B, and 1.3% higher than that of SubLoc. Similar observation can be made for the location accuracy. Roughly speaking, our method does not show significantly better performance than PSORT-B and

Table 5 Predictive performances of different approaches in the prediction of subcellular localization for Gram-negative bacteria

Location	PSORT I		PSORT-B		SubLoc[a]		Our method[b]	
	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC
Cytoplasmic	75.4	0.58	69.4	0.79	75.0	0.74	77.8	0.64
Inner membrane	95.1	0.64	78.7	0.85	82.8	0.89	78.4	0.83
Periplasmic	66.4	0.55	57.6	0.69	68.9	0.71	67.6	0.67
Outer membrane	54.5	0.47	90.3	0.93	89.1	0.86	94.6	0.78
Extracellular	–	–	70.0	0.79	69.5	0.78	72.6	0.78
Total accuracy	60.9	–	74.8	–	78.5	–	79.8	–
Location accuracy	58.3	–	73.2	–	77.1	–	78.2	–

[a] Based on the amino acid composition and SVM techniques (Hua and Sun 2001)

[b] Results are got at $k = 33$ and $m = 1.06$

SubLoc. But note that PSORT-B is a multimodular method, i.e., it comprises six modules examining the query sequence specifically for different characteristics such as amino acid composition, similarity to proteins of known localization, presence of a signal peptide, transmembrane α-helices, and motifs corresponding to specific localizations. Then, it constructs a Bayesian network to generate a final probability value for each localization site. While the SubLoc needs to select the proper kernel function parameters and the regularization parameter. So compared to the PSORT-B and SubLoc, our method is more straightforward and simple. It is interesting to note that SubLoc uses the amino acid composition as the only input vectors, but it shows a better overall performance than the more complicated multimodular PSORT-B. This surprisingly good predictive performances support previous observations that amino acid composition is indeed a good discriminator for subcellular localization.

Conclusion and discussion

Traditional development of the algorithms for predicting protein subcellular location generally focuses on investigating new and effective mathematical descriptors of protein sequences, e.g., the amino acid composition (Nakashima and Nishikawa 1994), pseudo amino acid composition (Chou 2001), amino acid pair and gapped amino acid pair compositions (Park and Kanehisa 2003), physicochemical properties (Lu et al. 2004; Xie et al. 2005), N-terminal sorting signals (Emanuelsson et al. 2000) and some calculated structural information (Andrade et al. 1998). In the present study, we propose a prediction which makes use of only a complexity measure of symbol sequences. The k-nearest neighbor method (or fuzzy k-NN method) is used to predict subcellular locations of three widely used data sets. The jackknife test shows that our method presents high prediction accuracy as compared to the other methods described. We hope our method could play a complementary role to existing experimental and computational methods for protein subcellular localization.

Compared to the ongoing machine learning methods, our method has the following characteristics. First, it searches for the most similar protein (or a group of proteins) in the training set instead of small set of parameters. Second, along with the first characteristic, our method can avoid the bias of selecting sequence features. Third, sequence-order effects are incorporated into prediction. Fourth, it indirectly uses specific signals common for proteins with fixed subcellular location (as fragments in LZ decomposition). More explicitly, when using LZ complexity-based distance, proteins within a fixed subcellular location are relatively 'closer' to each other as they are supposed to share some common sorting signals. However, our method also suffers from the disadvantage of high computational load relative to word statistic-based methods (computational complexity of the LZ decomposition algorithm is $O(n^2)$, where n is the sequence length). Therefore, furthermore attention should be paid on some other complexity measures of symbol sequences mentioned in the main text.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China.

References

Andrade MA, O'Donoghue SI, Rost B (1998) Adaptation of protein surfaces to subcellular location. J Mol Biol 276:517–525

Bernaola-Galván P, Carpena P, Román-Roldán R, Oliver JL (1999) Compositional complexity of DNA sequence models. Comput Phys Commun 121(1):136–138

Bezdek JC, Hall LO, Clarke LP (1993) Review of MR image segmentation techniques using pattern recognition. Med Phys 20:1033–1048

Boyd D, Schierle C, Beckwith J (1998) How many membrane proteins are there? Protein Sci 7:201–205

Cedano J, Aloy P, Pérez-Pons JA, Querol E (1997) Relation between amino acid composition and cellular location of proteins. J Mol Biol 266:594–600

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

Chou KC, Cai YD (2002) Using functional domain composition and support vector machines for prediction of protein subcellular location. J Biol Chem 277:45765–45769

Chou KC, Cai YD (2003) A new hybrid approach to predict subcellular localization of proteins by incorporating Gene ontology. Biochem Biophys Res Commun 311:743–747

Chou KC, Cai YD (2004) Prediction of protein subcellular locations by GO-FunD-PseAA predictor. Biochem Biophys Res Commun 320:1236–1239

Chou KC, Elrod DW (1999) Protein subcellular location prediction. Protein Eng 12:107–118

Chou KC, Shen HB (2007) Review: recent progresses in protein subcellular location prediction. Anal Biochem 370:1–16

Chou KC, Shen HB (2008) Cell-PLoc: a package of web-servers for predicting subcellular localization of proteins in various organisms. Nat Protoc 3:153–162

Chou KC, Zhang CT (1995) Review: prediction of protein structural classes. Crit Rev Biochem Mol Biol 30:275–349

Diao Y, Ma D, Wen Z, Yin J, Xiang J, Li M (2008) Using pseudo amino acid composition to predict transmembrane regions in protein: cellular automata and Lempel–Ziv complexity. Amino Acids 34(1):111–117

Duda RO, Hart PE, Stork DG (2000) Pattern classification, 2nd edn. Wiley, New York

Emanuelsson O, Nielsen H, Brunak S, Von Heijne G (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J Mol Biol 300:1005–1016

Feng ZP, Zhang CT (2002) A graphic representation of protein sequence and predicting the subcellular localizations of prokaryotic proteins. Int J Biochem Cell Biol 34:298–307

Gao QB, Wang ZZ, Yan C, Du YH (2005) Prediction of protein subcellular location using a combined feature of sequence. FEBS Lett 579:3444–3448

Gardy JL, Spencer C, Wang K, Ester M, Tusnady GE, Simon I, Hua S, deFays K, Lambert C, Nakai K, Brinkman FS (2003) PSORT-B: improving protein subcellular Iocalization prediction for Gram-negative bacteria. Nucleic Acids Res 31:3613–3617

Guo J, Lin YL, Sun ZR (2005) A novel method for protein subcellular localization: combining residue-couple model and SVM. Proc APBC 2005:117–129

Hua SJ, Sun ZR (2001) Support vector machine approach for protein subcellular localization prediction. Bioinformatics 17:721–728

Huang Y, Li YD (2004) Prediction of protein subcellular locations using fuzzy *k*-NN method. Bioinformatics 20:21–28

Lempel A, Ziv J (1976) On the complexity of finite sequence. IEEE T Inform Theory 22:75–81

Leszczynski K, Cosby S, Bissett R, Provost D, Boyko S, Loose S, Mvilongo E (1999) Application of a fuzzy pattern classifier to decision making in portal verification of radiotherapy. Phys Med Biol 44:253–269

Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, Poulin B, Anvik J, Macdonell C, Eisner R (2004) Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20:547–556

Mardia KV, Kent JT, Bibby JM (1979) Multivariate analysis. Academic Press, London

Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 405:442–451

Murphy RF, Boland MV, Velliste M (2000) Towards a systematics for protein subcellular location: quantitative description of protein localization patterns and automated analysis of fluorescence microscope images. Proc Int Conf Intell Syst Mol Biol 8:251–259

Nakai K (2000) Protein sorting signals and prediction of subcellular localization. Adv Protein Chem 54:277–344

Nakai K, Kanehisa M (1991) Expert system for predicting protein localization sites in Gram-negative bacteria. Proteins 11:95–110

Nakashima H, Nishikawa K (1994) Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. J Mol Biol 238:54–61

Nielsen H, Engelbrecht J, Brunak S, Von Heijne G (1997) A neural network method for identification of prokaryotic and eukaryotic signal perptides and prediction of their cleavage sites. Int J Neural Sys 8:581–599

Nielsen H, Brunak S, Von Heijne G (1999) Machine learning approaches for the prediction of signal peptides and other protein sorting signals. Protein Eng 12:3–9

Orlov YL, Potapov VN (2004) Complexity: an internet resource for analysis of DNA sequence complexity. Nucleic Acids Res 32:628–633

Otu HH, Sayood K (2003) A new sequence distance measure for phylogenetic tree construction. Bioinformatics 19:2122–2130

Park KJ, Kanehisa M (2003) Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 19:1656–1663

Reinhardt A, Hubbard T (1998) Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Res 26:2230–2236

Sadovsky MG (2003) The method to compare nucleotide sequences based on minimum entropy principle. Bull Math Biol 65:309–322

Troyanskaya OG, Arbell O, Koren Y, Landau GM, Bolshoy A (2002) Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity. Bioinformatics 18(5):679–688

Wang J, Zheng X (2008) Comparison of protein secondary structures based on backbone dihedral angles. J Theor Biol 250:382–387

Xiao X, Shao S, Ding Y, Huang Z, Huang Y, Chou KC (2005) Using complexity measure factor to predict protein subcellular location. Amino Acids 28:57–61

Xie D, Li A, Wang M, Fan Z, Feng H (2005) LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nucleic Acids Res 33:105–110

Yu CS, Lin CJ, Hwang JK (2004) Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Sci 13:1402–1406

Yuan Z (1999) Prediction of protein subcellular locations using Markov chain models. FEBS Lett 451:23–26

Ziv J, Lempel A (1977) A universal algorithm for sequential data compression. IEEE T Inform Theory 23:337–343

Ziv J, Lempel A (1978) Compression of individual sequences via variable-rate coding. IEEE T Inform Theory 24:530–536