RC24549 (W0805-030) May 6, 2008
Other (Bioinformatics)

IBM Research Report

“Localize”: An Accurate Method for Predicting a Protein's Sub-cellular Location

Aristotelis Tsirigos, Stanislav Polonsky,
Kevin C. Miranda*, Isidore Rigoutsos
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

*Center for Systems Biology
Harvard Medical School
Boston, MA 02114 USA

“Localize”: An Accurate Method for Predicting a Protein’s Sub-cellular Location

Aristotelis Tsirigos[1,┼], Stanislav Polonsky[1,┼], Kevin C. Miranda[2], Isidore Rigoutsos[1,*]

[1] *IBM Thomas J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598, U.S.A.*

[2] *Center for Systems Biology, Harvard Medical School, Boston, MA 02114, U.S.A.*

┼ *These authors have contributed equally to this work*

E-mail: atsirigo@us.ibm.com, polonsky@us.ibm.com, kmiranda@mgh.harvard.edu, rigoutso@us.ibm.com

[*] Correspondence should be addressed to I.R. (rigoutso@us.ibm.com)

Number of pages: 28

Number of figures: 8

Number of tables: 4

Number of words: 5,600 approx.

ABSTRACT

The computational prediction of a protein's sub-cellular location directly from the amino acid sequence is a well-known problem in bioinformatics. Together with structural and functional protein annotation methods, it is a valuable tool in high-throughput sequencing projects. In this work, we introduce a new method for the prediction of a protein's sub-cellular location that is pattern-based and relies on the analysis of the corresponding amino acid sequence. Our method uses a training set of amino acid sequences from which it generates both fixed- and variable-length amino acid patterns that it then uses to place unclassified proteins into one of twelve possible sub-cellular locations. Through a series of experiments, we demonstrate that the new method can achieve substantial improvements in average sub-cellular location accuracy and total accuracy over previously reported approaches. An implementation of the described method is available at: *http://cbcsrv.watson.ibm.com/localize.html*.

INTRODUCTION

Intracellular protein sorting is responsible for maintaining the correct structure and function of every cell within an organism. Organelles such as the nucleus, Golgi apparatus, endoplasmic reticulum (ER) and plasma membrane need to maintain a strict collection of resident proteins for optimal function. The importance of protein sorting is highlighted only when it breaks down and a disease state occurs [1, 2]. Protein trafficking is a highly-complex procedure involving various forms of cargo, carriers, destinations and routes. The entire process is highly dynamic and characterized by the constant movement of proteins throughout the cell.

Current hypotheses maintain that protein trafficking is dependent on bulk flow movement through the cell in combination with active sorting signals and retention signals which are present within proteins [3]. Most proteins are able to reach their destination by using one or more of signals directing general bulk flow, active sorting or retention. For example, a newly synthesized cadherin protein uses a *N*-terminal signal sequence to enter the ER, bulk flow to transverse from the ER to the Golgi apparatus and finally a basolateral sorting signal to reach the plasma membrane. At the plasma membrane it can be either retained through interaction with other cadherins, or undergo endocytosis through an as-yet-undefined mechanism [4].

The myriad of trafficking steps undertaken by E-cadherin exemplify the scale and difficulty of predicting a proteins sub-cellular localization. Experimental validation of the sub-cellular localization of an individual protein is currently a slow and labor-intensive process. Computational methods that can help speed-up the elucidation of the underlying sequence signals are thus very important.

Predicting sub-cellular localization is a well known problem in computational biology and several methods have been proposed to date that address this task. We refer the interested reader to several review articles which have already covered this subject [5-7] rather extensively.

The localization prediction methods can be broken down into three major categories. The first category relies on the use of previously-discovered biological signals, such as protein sorting and retention signals, in order to predict protein localization [8-18]. An important limitation of these methods is that they require the knowledge of such signals. Unfortunately, many of these signals continue to elude us as their identification requires time-consuming and expensive lab experiments. Consequently, the prediction of subcellular localization using this class of methods is possible for only a small fraction of all proteins of interest.

The second category comprises methods that are based on sequence homologies, identification of protein domains, or other available functional annotations. Phylogenetic profiles using BLAST or PSI-BLAST were proposed in [9, 19, 20], maximal patterns of InterPro domains in [21-24], SMART domains in [25], and functional annotations in [26-29].

The third category of methods does not make use of any such biological signals. Instead these methods rely on the observation that global sequence features, such as amino acid composition, can be specific to a sub-cellular location [30, 31]. Numerous tools belonging to this category have been developed in recent years, the most prominent of which are described in [20, 23, 24, 32-38].

In the work that we present below, we extend the *third category* by introducing a new, pattern-based method for predicting protein sub-cellular localization. The method, which is described in Section 2, combines *global* sequence properties (such as amino acid composition) with *local* sequence properties (represented by short, statistically-significant, variable-length patterns of amino acids). In Section 3, we evaluate the method's performance and compare its performance to that achieved by an earlier method on the same dataset. Section 4 concludes our paper with a discussion and directions for future work in this area.

METHODS

Overview of unsupervised pattern discovery

Our method is based on the use of a collection of amino acid patterns that are discovered in an automated manner and cover the sequence space of the training set under consideration. Such signals have been shown to capture functional and structural signals [27-29].

Typically, an unsupervised pattern discovery tool takes a set *D* of protein sequences as input and discovers a comprehensive set of patterns that appear recurrently in different subsets of sequences. For the work described below, we have used the Teiresias pattern discovery algorithm [39, 40]; the algorithm can provably find *all* patterns p in the input set *D* that satisfy the following properties:

(1) each p is composed of either literal characters, i.e. individual amino acids, or classes of amino acids (designated by their inclusion in brackets), possibly separated by a number of wild-cards characters ("dots"); a wild-card indicates that the corresponding position can be occupied by any amino acid.

(2) each p comprises at least L literal characters (or equivalence classes) in any span of $W \geq L$ positions. Then the pattern *p* is considered to be an $\langle L, W \rangle$ pattern. For example, the pattern A.C.[FY]..L is a <2,4> pattern, whereas patterns A…C.[FY]..L, A.C...[FY]..L and A.C.[FY]…L are <2,5> patterns.

(3) each p occurs at least *K* times in the set *D*. *K* is referred to as the "support" of pattern *p*.

Teiresias works in two phases which are termed scanning and convolution. Scanning is performed in order to discover all <*L, W*> patterns with length at most *W* ("seed patterns"). These seed patterns are combined during convolution to form progressively longer patterns. The extension process is guided by the contents of the processed dataset and thus terminates naturally – the algorithm imposes no upper bounds on the length of the discovered patterns.

Method A: Predicting sub-cellular location using fixed-length patterns

First, we build a classifier for predicting protein sub-cellular locations using only the seed patterns that are generated by the scanning phase of Teiresias. The key idea is to explore the use of higher-order amino acid patterns (cf. the amino acid pairs used in [34]) in an effort to improve accuracy while at the same time discover a simple one-classifier model to perform the task. This bypasses the need for elaborate voting schemes that are necessary when multiple classifier methods are used.

The discovery step generated a total of about 5.5 million patterns that belong to one of the four different categories shown here:

- L=1 and W=1 with chemical equivalences: this category comprises 27 patterns, one pattern for each of the 20 amino acids plus one for each of the 7 chemical equivalence classes shown in Table 1. It is also referred to as single amino acid composition.
- L=2 and W=2 with chemical equivalences: this class comprises a total of $27^2 = 729$ patterns, and is also known as composition of amino acid pairs.
- L=3 and W=5 with chemical equivalences: this category comprises all patterns containing exactly 3 letters (amino acids, or equivalence classes of amino acids) possibly separated by at most two wildcards, a total of $6 \cdot 27^3 = 118{,}098$.
- L=4 and W=6 with chemical equivalences: this category comprises all patterns containing exactly 4 letters (amino acids, or equivalence classes of amino acids) possibly separated by a total of at most two wildcards, a total of $10 \cdot 27^4 = 5{,}314{,}410$.

In Table 2 we show several examples of patterns from each category.

We use these four types of patterns to decompose each input sequence effectively converting it into a feature vector: each feature corresponds to one of the discovered patterns and the feature's value is equal to the number of times the pattern is found in the protein sequence. These feature values are subsequently normalized per unit length in order to treat short as well as long proteins on an equal basis. Also, given that shorter patterns/features are expected to occur much more frequently than longer ones, we linearly scale the *feature values* across proteins and for each feature separately, so that they range from 0 to 1. This last step removes the bias towards more frequent patterns and is necessary: otherwise, more frequent patterns would have been treated as more predictive that less frequent ones. In practice, there exist long patterns which, despite occurring only once in some sequences, they can actually be used to predict protein localization much more accurately than the frequency of single amino acids, because they turn out to be specific to a given sub-cellular location.

After the preprocessing of the training and test feature vectors is complete, we train an SVM classifier [41, 42] using an RBF kernel and classify the test vectors according to the model obtained from the training phase. The highly optimized SVM package LibSVM by Chang and Lin was used to train the SVM classifier and do the final testing: see http://www.csie.ntu.edu.tw/~cjlin/libsvm for the code and reference manuals for LibSVM. Figure 1 summarizes the training and testing processes.

It is worth mentioning that for a classification problem that is characterized by millions of features, feature selection becomes an important preprocessing step. As advocated by LibSVM developers [43], we apply multi-class Fisher scoring [44] to evaluate the importance of individual features. We found that we can boost classification accuracy by selecting 25% of the top scoring features while significantly reducing classifier training and testing times.

Method B: Predicting sub-cellular location using variable-length patterns

The following observations highlight the importance of discovering variable-size patterns in the input set (training set).

- patterns that are allowed to grow in length in an unrestricted manner will be as specific as possible for the given input dataset
- long patterns can be highly significant even if they appear few times in the dataset
- highly-significant, variable-length patterns can help identify important local similarities among sequences which are destined for the same sub-cellular location; on the other hand, computations of similarities among full-length sequences can lead to artificially high (or low) values since they ignore the small-by-comparison part of the sequence which is relevant for the classification task at hand.

Figure 2 summarizes the training and testing process for Method B. In the first step, unsupervised pattern discovery is performed using both the scanning and the convolution phases of Teiresias in order to extract all patterns contained in the training set that are also maximal in composition and length. The parameters we used for this step were L=4 and W=6 with minimum support set to K=2 – no amino acid equivalences were taken into account during this step. In general, the total number of discovered patterns can be very high. Clearly, this number is affected by the choice of parameters and the use of amino acid equivalences (e.g. chemical, structural etc.). As one would intuitively expect, a larger pattern collection could potentially increase the final classification performance. However, in the presence of more patterns, the training and classification tasks would become harder to manage given that the computational resources in terms of memory, disk storage and processing power are finite.

During the second step, we compute z-scores for each discovered pattern as a function of its expected probability and its support in the database. Formally, the z-score z_p of a pattern p is computed using the following formula:

$$z_p = \frac{N_p - D \cdot \Pr(p)}{\sqrt{D \cdot \Pr(p) \cdot (1 - \Pr(p))}}$$

where N_p is the observed number of occurrences of the pattern in the given dataset, D is the size of the dataset (total number of amino acids), and Pr(p) is the expected probability of the pattern given the observed probabilities of single amino acids under the assumption of independent, identically distributed variables. Patterns that have z-scores lower than a threshold θ are discarded.

The third step ensures that, among the highly-significant patterns, only those which give rise to the same sub-cellular location are kept: in other words, we keep only the patterns that are found in protein sequences of the same sub-cellular location (zero-entropy). We note here that since this is done using the training set only there is no guarantee that the same will hold true in the test set. However, this "guilty by association" approach has been time-honored and is very typical for this kind of methods: intuitively, we do expect this to be the case most of the time.

With the completion of the third step, we now have a set of highly-significant patterns each one of which is associated with a specific sub-cellular location. We use these pattern sets as predicates that can predict the eventual sub-cellular location of the test sequences. Since there is no guarantee that these patterns will appear in a test sequence unchanged, we introduce what we refer to as a "pattern matching score" between a pattern and a protein sequence: this score is defined as the maximum fraction over all possible ungapped alignments of the total number of matched amino acids in the pattern/protein alignment divided by the total number of matched and unmatched amino acids – obviously, this score ranges between 0 and 1 inclusive.

We are now ready to assign predictions to our test sequences. This is simply done by finding, for any given test sequence, the pattern with the highest z-score which aligns best with the test sequence (i.e. leads to the highest matching score). If the matching score is greater than or equal to a threshold α, then the test sequence is assigned to the location associated with the matching pattern, otherwise it remains unassigned (inability to predict with confidence). In other words, we try each pattern in turn, in order from the highest to the lowest z-score: when a pattern is found whose matching score is not lower than

α, we stop and assign the test sequence to the sub-cellular location of the pattern at hand. Clearly, more elaborate schemes could be applied, but such an endeavor is beyond the scope of this work; our goal is to demonstrate that variable-length patterns can in fact be used effectively to improve prediction accuracy. In the Results section, after evaluating Methods A and B separately, we also evaluate the final hybrid method "Localize" that is proposed in this paper which uses Method B in conjunction with Method A in order to deal with the test sequences that are left unclassified by Method B. First Method B is applied in order to classify a test sequence and, then, if none of the patterns that Method B has at its disposal have instances in the sequence at hand, Method A is applied to make the prediction.

RESULTS

In this section, we will analyze separately the performance of each one of the component methods (A and B – see above). Finally, we will discuss the performance of the hybrid method that uses both A and B as its sub-components. The hybrid method is the one to which we refer as "Localize" and we compare it with three different, previously reported methods.

Method A: Predicting sub-cellular location using fixed-length patterns

Despite the fact that a lot of research has been done on computationally predicting protein sub-cellular locations, the area still lacks universally accepted reference datasets and performance measures. We thus chose to work with the dataset introduced in [34] which includes a large number of proteins classified into 12 sub-cellular location (i.e. categories). The location-specific datasets are derived from eukaryotic entries of Swiss-Prot database release 39.0 based on the content of SUB-CELLULAR LOCATION section of CC (comment) lines: the 12 sub-cellular locations that are covered include chloroplast, cytoplasm, cytoskeleton, endoplasmic reticulum (ER), Golgi apparatus, lysosome, mitochondria, nucleus, peroxisome, plasma membrane, and vacuole. The datasets are located at http://web.kuicr.kyoto-u.ac.jp/~park/Seqdata/. In this dataset, highly similar proteins of sequence similarity more than 80% percent were grouped together, and from each group of similar proteins only one sequence was selected. This is a much more stringent threshold than the 90% threshold usually employed by other techniques [20, 33, 37] and one that makes the classification task much harder. The number of entries in each category is indicated in the first column of Table 3.

In order to estimate the prediction performance of our method we employed a 5-fold cross-validation test as in [19]. The idea of the test is to split the dataset into five approximately equal subsets. One of the five subsets is used as a collection of test sequences whereas the remaining four subsets are joined together and form the training set. This process is repeated five times so that each of the five subsets in turn is used as a test collection.

The final performance is measured on the five test sets and is defined separately for each sub-cellular location i as $P_i = T_i / n_i$, where T_i is the number of sequences correctly ascribed to the i-th category (a.k.a. true positives) and n_i is the total number of sequences in this category. This last measure is often referred to as *sensitivity*.

In addition, we define two cumulative, location-independent measures. The first one, *location accuracy*, is an average of P_i over all K locations and is defined as follows: $LP = \sum_{i=1}^{K} P_i / K$. The second measure, *total accuracy*, is the fraction of correct predictions for the total of N sequences in the dataset and is defined as follows: $TP = \sum_{i=1}^{K} T_i / N$.

The two cumulative measures are complementary: TP tracks performance mainly in categories with large numbers of sequences. On the contrary, LP treats each category equally regardless of the category's size.

We set up an optimization grid in order to determine the optimal parameters *β, c* and *γ* of our SVM classifier: *β* is the percentage of top scoring features selected for training, *c* is used to control the complexity of the learned hyperplane, and, *γ* is a parameter of the RBF kernel. As shown in Figure 3, the test accuracy is computed for each parameter triplet (*c,γ,β*) using the 5-fold cross-validation process. Using this process, we determined that the maximum test accuracy was achieved when *β=25%*, *c=64* and *γ=0.0001*, and the resulting value for the total accuracy of our SVM classifier was *82.4%*. This represents a very significant performance improvement over PLOC whose total accuracy is *78.2%*.

This performance improvement is particularly notable if one considers the following:

a) we obtain it using a single classifier (cf. for example the 5 classifiers used by PLOC);

b) there is a single value for the parameter *γ* of the RBF kernel (vs. use of a mixture of two different *γ* values by PLOC);

and, most importantly,

c) our approach obviates the need for the use of a voting scheme that combines the results from multiple classifiers.

Analogously, our achieved location accuracy performance is *62.1%* which again represents a considerable improvement over PLOC's performance of *57.9%*. Table 3 details the results of our method for each category separately.

Method B: Using variable-length patterns

Figure 4 highlights the tradeoff between accuracy and coverage at various z-score threshold levels when Method B is used alone: as the threshold for pattern selection increases the number of sequences in the test set that will be covered by those patterns decreases, but the prediction accuracy does increase as a result. For example, at z-score threshold $log\theta=20$, only 63% of the 7579 proteins are classified but the classification accuracy reaches the impressive level of 93.5%; however, if we attempt to cover more proteins by lowering the threshold to $log\theta=15$, although almost all proteins are covered (96%), the classification accuracy drops sharply.

These findings suggested that instead of trying to cover all sequences using the individual patterns of Method B, a hybrid method that combined the best characteristics from Method A and Method B would be a better choice.

"Localize": Hybrid Method B/A -- Using fixed- and variable-length patterns to further improve accuracy

The hybrid scheme that we advocate works as follows. We first use Method B to classify a test sequence. If none of the patterns that Method B has at its disposal have instances in the sequence at hand, then Method A is brought to bear.

The same training and testing approach used for Method A was applied in order to evaluate the hybrid Method B/A. A grid search was set up in order to determine the optimal z-score threshold θ and the pattern matching cutoff α for our Method B classifier, while, for Method A, we simply used the optimal parameters obtained from the previous optimization of Method A alone. The optimization process for determining the optimal parameters of α and θ is summarized in Figure 5. Ideally, we would have attempted a joint optimization over all 5 parameters of the two methods, which conceivably would have increased performance even further. However, this optimization over 5 parameters would have required a tremendous amount of computational resources.

This hybrid scheme works very well as can be seen from the results shown in Table 3. The hybrid Method B/A approach exhibits markedly better total (=84.4%) and location (=70.8%) prediction accuracies when compared to the corresponding PLOC values (78.2% and 57.9% respectively). These numbers reflect an improvement of 6.2% and 12.9% respectively. This is especially important for the location accuracy as it implies that better predictions can now be made for the under-represented categories. Indeed, we achieve an almost 3–fold improvement for the "Golgi apparatus" category and a 2–fold improvement for the "peroxisome" and "vacuole" categories when compared to PLOC. Analogous performance improvements are achieved for all remaining location categories as can be seen in Figure 6.

Finally, in order to demonstrate that discovering statistically significant patterns is rather different than simple homology searches, we compared our final method with a method which performs localization prediction of an uncharacterized protein by assigning it to the localization of the closest BLAST hit in the rest of the database. Using the same 5-fold cross-validation we compared the two methods and the results are shown in Figure 7, in which it can be seen clearly that our method outperforms the BLAST-based search method by a huge margin. The main conclusion here is that, after removing highly similar proteins, simple homology searches result in proteins with different localizations.

Evaluation using a lower similarity threshold

In order to further ensure that our method improves the accuracy of protein localization prediction, we evaluated its performance on an even less "redundant" dataset, consisting of a set of proteins with maximum pairwise similarity no more that 50% and we compared our results to the ones obtained using the fuzzy k-nearest neighbors method in [36]. The results for the 11 sub-cellular localization used in [36] are shown in Table 4 and in Figure 8, and demonstrate an improvement of 10.5% in total accuracy and 8.3% in location accuracy compared to the fuzzy k-nearest neighbors method.

CONCLUSION

We have presented a new method that allows us to confidently predict sub-cellular protein locations directly from a protein's amino acid sequence. It is based on the unsupervised discovery of fixed-length as well as variable-length patterns. Our method results in a significantly-improved ability to predict a protein's eventual location directly from amino acid sequence. When compared with the state-of-the-art amino-acid-composition-based tool PLOC, we demonstrate improvements of total accuracy by 6.2% and of location accuracy by 12.9% respectively.

Despite significant computational advances over the years, the problem of sub-cellular protein localization is still far from solved for eukaryotic organisms. And, even though we have demonstrated that our method achieves significant prediction gains, we believe that it is only prudent for practitioners to use the output from all available prediction tools before drawing any conclusions.

Our future work will concentrate on the analysis of factors which limit the performance of the various methods. In this regard, one important improvement, we believe, is likely to result from the use of organism-specific datasets. A significantly harder variation of this problem would require that one address the case of proteins with *multiple* locations and that one predict *all* intermediate such locations.

REFERENCES

1. Aridor, M. and L.A. Hannan, *Traffic jam: a compendium of human diseases that affect intracellular transport processes.* Traffic, 2000. **1**(11): p. 836-51.
2. Aridor, M. and L.A. Hannan, *Traffic jams II: an update of diseases of intracellular transport.* Traffic, 2002. **3**(11): p. 781-90.
3. Mellman, I., *Endocytosis and molecular sorting.* Annu Rev Cell Dev Biol, 1996. **12**: p. 575-625.
4. Bryant, D.M. and J.L. Stow, *The ins and outs of E-cadherin trafficking.* Trends Cell Biol, 2004. **14**(8): p. 427-34.
5. Nakai, K., *Protein sorting signals and prediction of subcellular localization.* Adv Protein Chem, 2000. **54**: p. 277-344.
6. Emanuelsson, O., *Predicting protein subcellular localisation from amino acid sequence information.* Brief Bioinform, 2002. **3**(4): p. 361-76.
7. Schneider, G. and U. Fechner, *Advances in the prediction of protein targeting signals.* Proteomics, 2004. **4**(6): p. 1571-1580.
8. Bendtsen, J.D., et al., *Improved prediction of signal peptides: SignalP 3.0.* Journal Of Molecular Biology, 2004. **340**(4): p. 783-795.
9. Nakai, K. and M. Kanehisa, *A knowledge base for predicting protein localization sites in eukaryotic cells.* Genomics, 1992. **14**(4): p. 897-911.
10. Boden, M. and J. Hawkins, *Prediction of subcellular localization using sequence-biased recurrent networks.* Bioinformatics, 2005. **21**(10): p. 2279-2286.
11. Chen, Y.J., et al., *Secreted protein prediction system combining CJ-SPHMM, TMHMM, and PSORT.* Mammalian Genome, 2003. **14**(12): p. 859-865.
12. Nair, R. and B. Rost, *Mimicking cellular sorting improves prediction of subcellular localization.* Journal Of Molecular Biology, 2005. **348**(1): p. 85-100.
13. Zhang, Z.M. and W.J. Henzel, *Signal peptide prediction based on analysis of experimentally verified cleavage sites.* Protein Science, 2004. **13**(10): p. 2819-2824.
14. Small, I., et al., *Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences.* Proteomics, 2004. **4**(6): p. 1581-1590.
15. Hiller, K., et al., *PrediSi: prediction of signal peptides and their cleavage positions.* Nucleic Acids Research, 2004. **32**: p. W375-W379.
16. Heddad, A., M. Brameier, and R.M. MacCallum, *Evolving regular expression-based sequence classifiers for protein nuclear localisation*, in *Applications Of Evolutionary Computing*. 2004. p. 31-40.
17. Drawid, A. and M. Gerstein, *A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome.* J Mol Biol, 2000. **301**(4): p. 1059-75.
18. Emanuelsson, O., et al., *Predicting Subcellular Localization of Proteins Based on their N-terminal Amino Acid Sequence.* J Mol Biol, 2000. **300**(4): p. 1005-1016.
19. Marcotte, E.M., et al., *Localizing proteins in the cell from their phylogenetic profiles.* Proc Natl Acad Sci U S A, 2000. **97**(22): p. 12115-20.
20. Bhasin, M. and G.P.S. Raghava, *ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST.* Nucleic Acids Research, 2004. **32**: p. W414-W419.
21. Scott, M.S., et al., *Refining protein subcellular localization.* PLoS Comput Biol, 2005. **1**(6): p. e66.

22. Scott, M.S., D.Y. Thomas, and M.T. Hallett, *Predicting subcellular localization via protein motif co-occurrence.* Genome Res, 2004. **14**(10A): p. 1957-66.
23. Cai, Y.D. and K.C. Chou, *Nearest neighbour algorithm for predicting protein subcellular location by combining functional domain composition and pseudo-amino acid composition.* Biochemical And Biophysical Research Communications, 2003. **305**(2): p. 407-411.
24. Cai, Y.D. and K.C. Chou, *Predicting subcellular localization of proteins in a hybridization space.* Bioinformatics, 2004. **20**(7): p. 1151-1156.
25. Mott, R., et al., *Predicting protein cellular localization using a domain projection method.* Genome Res, 2002. **12**(8): p. 1168-74.
26. Nair, R. and B. Rost, *LOCnet and LOCtarget: sub-cellular localization for structural genomics targets.* Nucleic Acids Research, 2004. **32**: p. W517-W521.
27. Nair, R. and B. Rost, *Sequence conserved for subcellular localization.* Protein Sci, 2002. **11**(12): p. 2836-47.
28. Nair, R. and B. Rost, *Inferring sub-cellular localization through automated lexical analysis.* Bioinformatics, 2002. **18 Suppl 1**: p. S78-86.
29. Lu, Z., et al., *Predicting subcellular localization of proteins using machine-learned classifiers.* Bioinformatics, 2004. **20**(4): p. 547-56.
30. Nakashima, H. and K. Nishikawa, *Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies.* J Mol Biol, 1994. **238**(1): p. 54-61.
31. Reinhardt, A. and T. Hubbard, *Using neural networks for prediction of the subcellular location of proteins.* Nucleic Acids Res, 1998. **26**(9): p. 2230-6.
32. Gao, Q.B., et al., *Prediction of protein subcellular location using a combined feature of sequence.* Febs Letters, 2005. **579**(16): p. 3444-3448.
33. Garg, A., M. Bhasin, and G.P.S. Raghava, *Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search.* Journal Of Biological Chemistry, 2005. **280**(15): p. 14427-14432.
34. Park, K.J. and M. Kanehisa, *Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs.* Bioinformatics, 2003. **19**(13): p. 1656-1663.
35. Xiao, X., et al., *Using complexity measure factor to predict protein subcellular location.* Amino Acids, 2005. **28**(1): p. 57-61.
36. Huang, Y. and Y. Li, *Prediction of protein subcellular locations using fuzzy k-NN method.* Bioinformatics, 2004. **20**(1): p. 21-28.
37. Matsuda, S., et al., *A novel representation of protein sequences for prediction of subcellular location using support vector machines.* Protein Sci, 2005. **14**(11): p. 2804-13.
38. Reczko, M. and A. Hatzigeorgiou, *Prediction of the subcellular localization of eukaryotic proteins using sequence signals and composition.* Proteomics, 2004. **4**(6): p. 1591-1596.
39. Rigoutsos, I. and A. Floratos, *Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm.* Bioinformatics, 1998. **14**(1): p. 55-67.
40. Rigoutsos, I. and A. Floratos, *Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm (vol 14, pg 55, 1998).* Bioinformatics, 1998. **14**(2): p. 229-229.
41. Vapnik, V.N., *The Nature of Statistical Learning Theory*. 2 ed. 1999: Springer. 314
42. Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. 1 ed. 2000: Cambridge University Press. 189
43. Chen, Y.-W. and C.-J. Lin, *Combining SVMs with various feature selection strategies.* , in *Feature extraction, foundations and applications*. 2005, to be published.

44. Chen, Y.-W., *Combining SVMs with various feature selection strategies*, in *Department of Computer Science and Information Engineering*. 2005, National Taiwan University.

FIGURE LEGENDS

Figure 1: **Training/testing for method based on fixed-length pattern discovery (Method A).**

Figure 2: **Training/testing for method based on variable-length pattern discovery (Method B).**

Figure 3: **5-fold cross-validation for Method A.**

Figure 4**: Tradeoff between accuracy and coverage using Method B as standalone.**

Figure 5: **5-fold cross-validation for hybrid Method B/A.**

Figure 6: **Comparison of our method with PLOC [34].**

Figure 7: **Comparison of our method with simple best-hit using BLAST.**

Figure 8: **Comparison of our method with fuzzy k-NN method [36].**

Figure 1: Training/testing for method based on fixed-length pattern discovery (Method A).

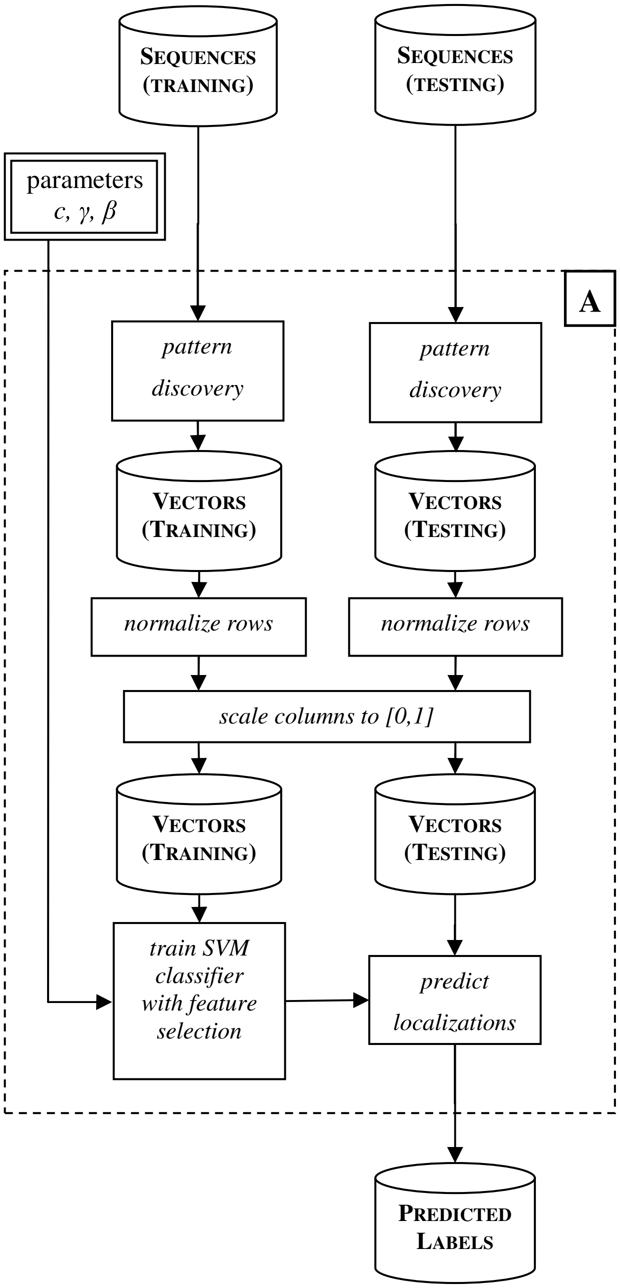

Figure 2: Training/testing for method based on variable-length pattern discovery (Method B).

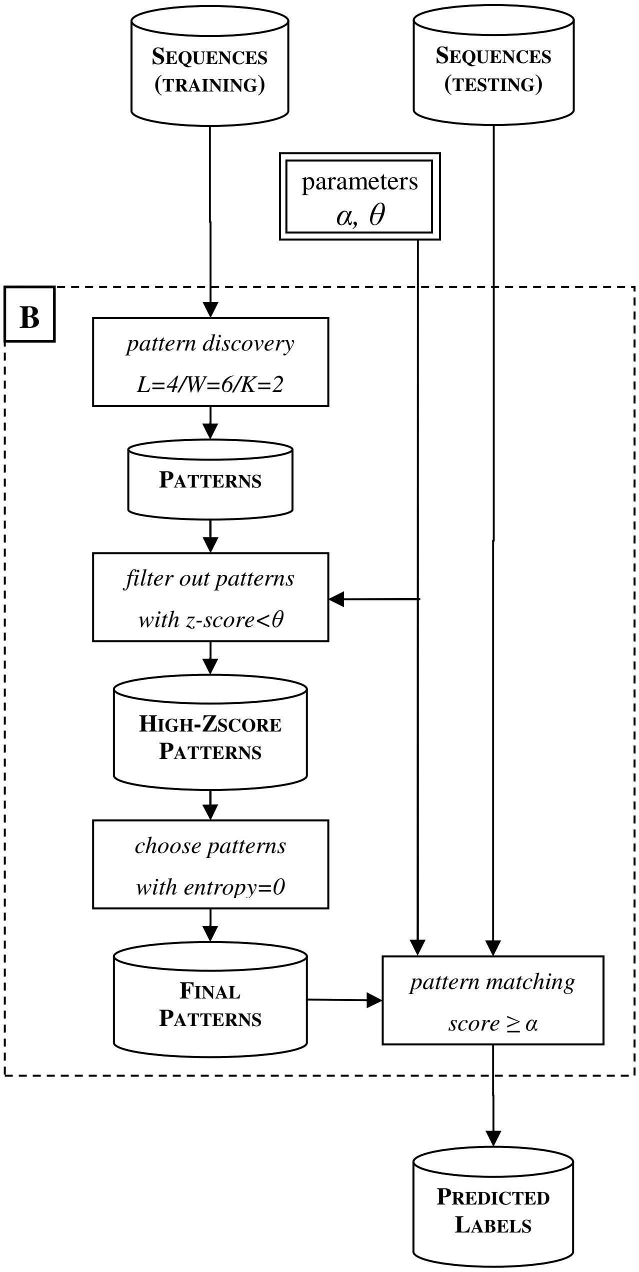
SEQUENCES (TRAINING)
SEQUENCES (TESTING)
parameters α, θ
B
pattern discovery L=4/W=6/K=2
PATTERNS
filter out patterns with z-score<θ
HIGH-ZSCORE PATTERNS
choose patterns with entropy=0
FINAL PATTERNS
pattern matching score ≥ α
PREDICTED LABELS

Figure 3: 5-fold cross-validation for Method A.

```
INPUT: 7579 protein sequences and their known locations

PARTITION input data into 5 folds
CHOOSE parameters ($c$,$\gamma$,$\beta$) from optimization grid
  FOR fold j = 1 to 5
    Obtain predictions using Method A (Figure 1) on fold $j$ with parameters ($c$,$\gamma$,$\beta$)
  END FOR
  Compute total accuracy based on the predictions from all 5 folds
END CHOOSE
Select the parameters ($c$,$\gamma$,$\beta$) that yield the maximum accuracy
```

Figure 4: Tradeoff between accuracy and coverage using Method B as standalone.

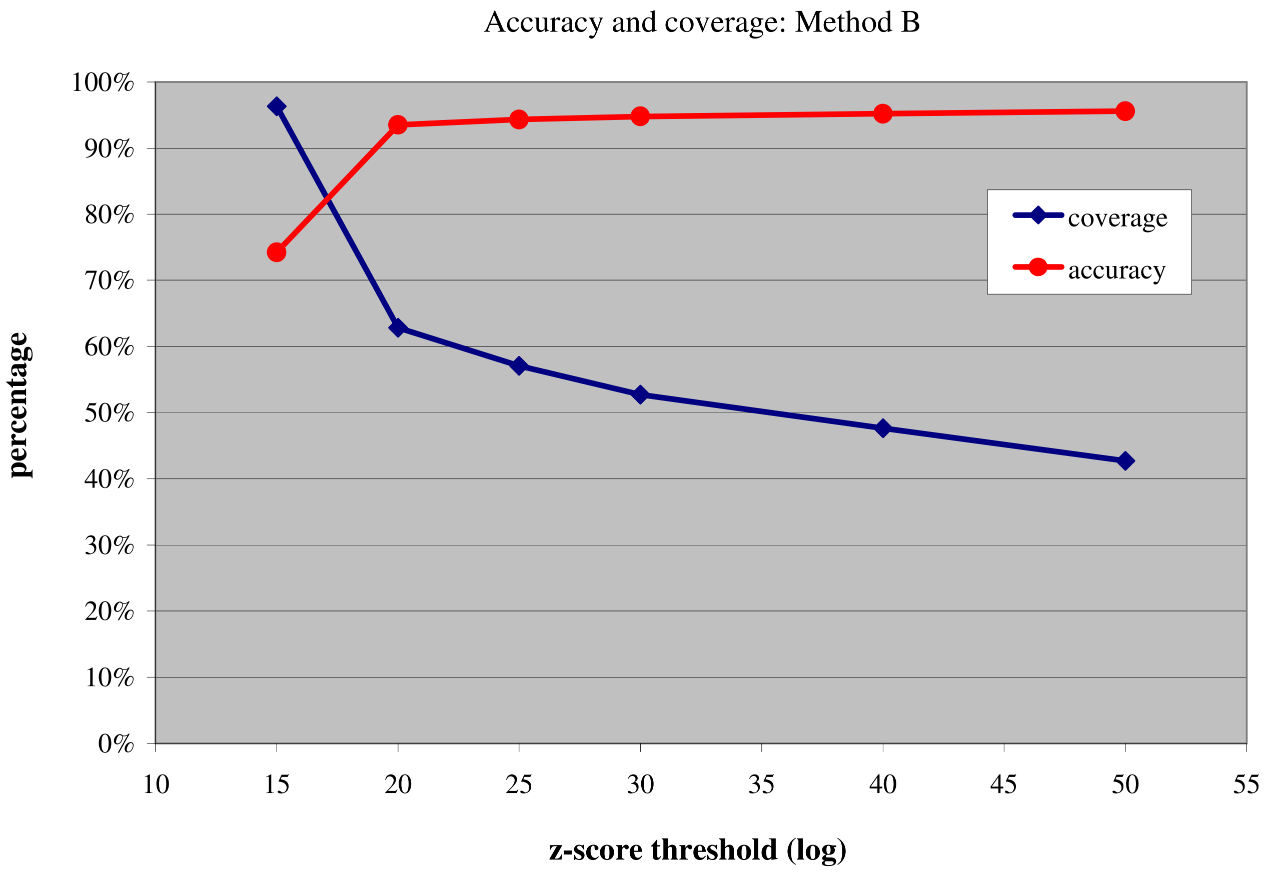

Figure 5: 5-fold cross-validation for hybrid Method B/A.

```
INPUT: 7579 protein sequences and their known locations
PARTITION input data into 5 folds
CHOOSE parameters ($\alpha$,$\theta$) from optimization grid
  FOR fold j = 1 to 5
    Obtain predictions using Method B (Figure 2) on fold $j$ with parameters ($\alpha$,$\theta$)
    Obtain predictions for unclassified instances using Method A
  END FOR
  Compute total accuracy based on the predictions from all 5 folds
END CHOOSE
Select the parameters ($\alpha$,$\theta$) that yield the maximum accuracy
```

Figure 6: Comparison of our method with PLOC [34].

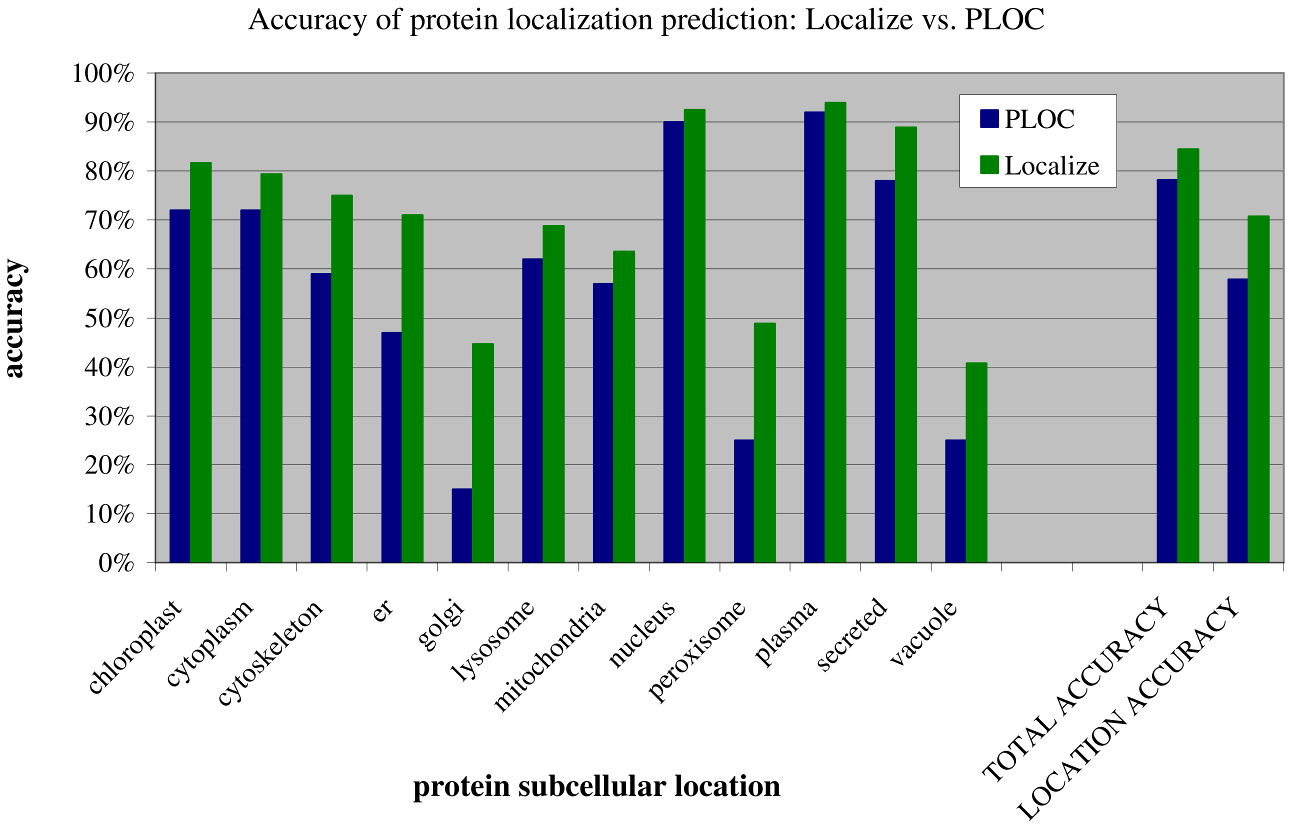

Figure 7: Comparison of our method with simple best-hit using BLAST.

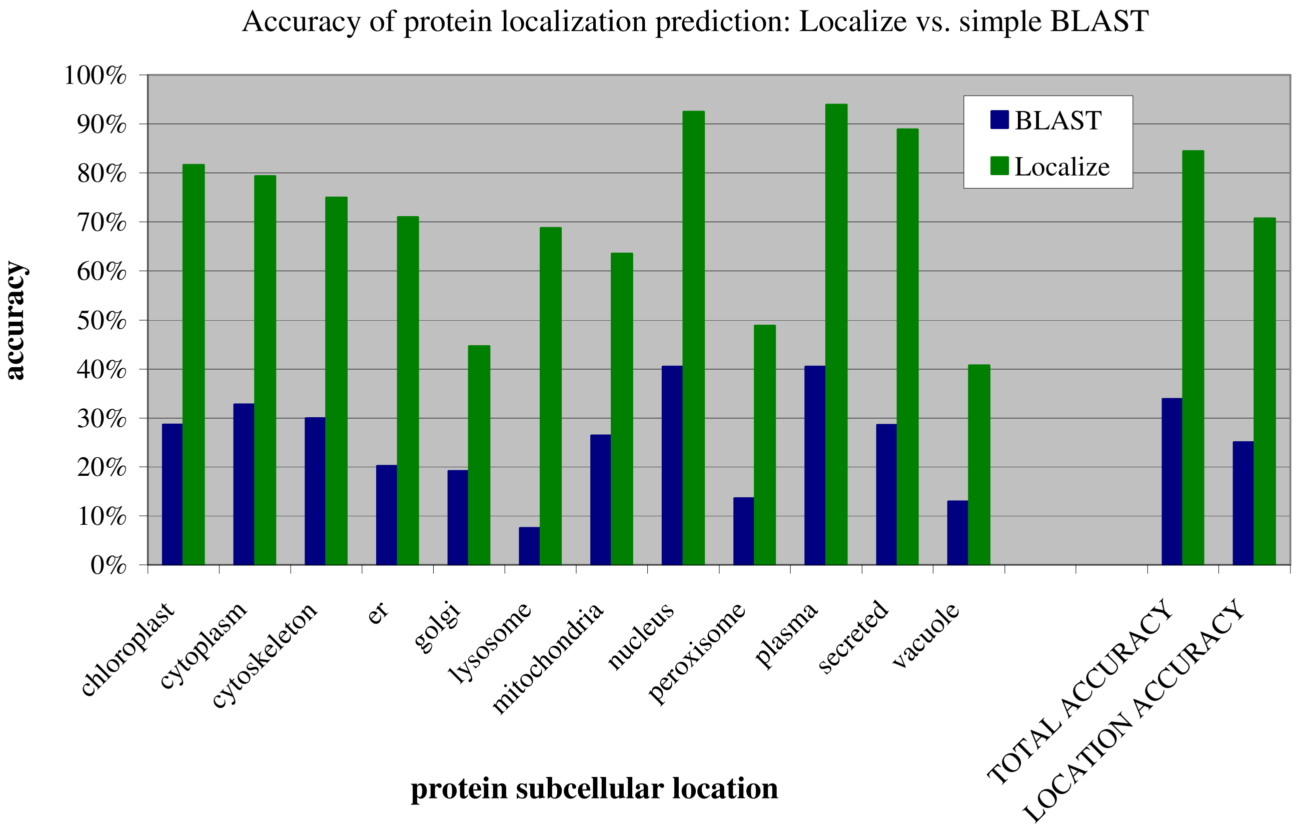

Figure 8: Comparison of our method with fuzzy k-NN method [36].

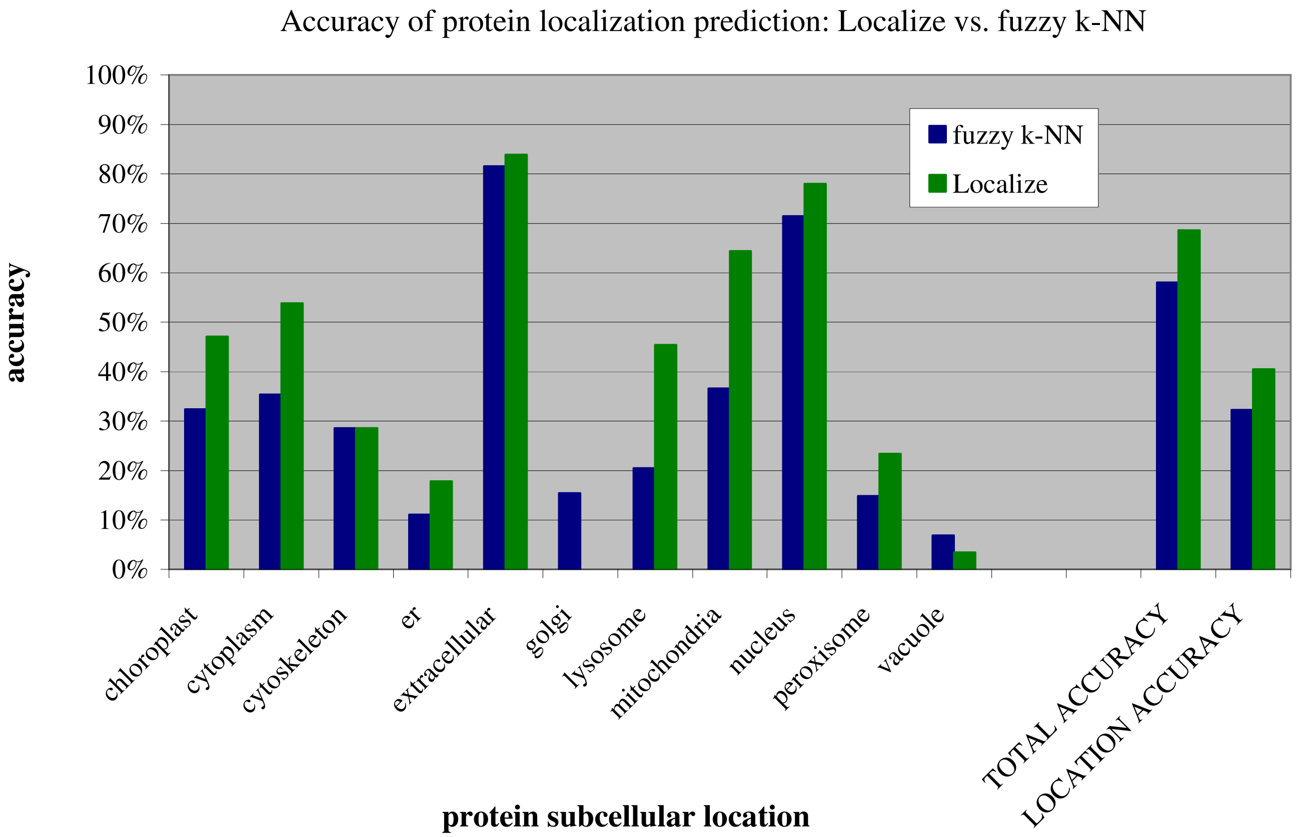

Table 1: Chemical equivalence classes for amino acids.

Equivalence class members	Symbol
A, G	[AG]
D, E	[DE]
F, Y	[FY]
K, R	[KR]
I, L, M, V	[ILMV]
N, Q	[NQ]
S, T	[ST]

Table 2: Examples of patterns.

Category	Examples
L=1/W=1	A Q [AG] [ILMV]
L=2/W=2	AE ST [DE]A T[ILMV] [AG][NQ] [KR][ILMV]
L=3/W=5	ADY AD.Y A.DY A..DY A.D.Y AD..Y [AG][DE][FY] [AG]..D[ILMV]
L=4/W=6	ADYV AD.V.Y [AG].[DE].[FY]A

Table 3: Comparison of prediction accuracy (sensitivity) for the 12 sub-cellular locations. The PLOC data is taken from [34].

Location (no. of entries)	Hybrid Method B/A	Method A	PLOC
Chloroplast (671)	**81.7%**	79.0%	72.0%
Cytoplasm (1245)	**79.4%**	77.9%	72.0%
Cytoskeleton (41)	**75.0%**	72.5%	59.0%
ER (114)	**71.1%**	58.8%	47.0%
Golgi apparatus (48)	**44.7%**	14.9%	15.0%
Lysosome (93)	**68.8%**	52.7%	62.0%
Mitochondria (727)	**63.5%**	60.7%	57.0%
Nucleus (1932)	**92.5%**	91.3%	90.0%
Peroxisome (125)	**48.8%**	36.0%	25.0%
Plasma membrane (1677)	**94.0%**	94.1%	92.0%
Secreted (862)	**89.0%**	87.6%	78.0%
Vacuole (54)	**40.7%**	20.4%	25.0%
Total accuracy, TP	**84.4%**	**82.3%**	**78.2%**
Location accuracy, LP	**70.8%**	**62.1%**	**57.9%**

Table 4: Comparison of prediction accuracy (sensitivity) for the 11 sub-cellular locations. The fuzzy k-NN data is taken from [36].

Location (no. of entries)	Hybrid Method B/A	fuzzy k-NN
Chloroplast (225)	**47.1%**	32.4%
Cytoplasm (622)	**53.9%**	35.4%
Cytoskeleton (7)	**28.6%**	28.6%
ER (45)	**17.8%**	11.1%
Extracellular (915)	**83.9%**	81.6%
Golgi apparatus (26)	**0.0%**	15.4%
Lysosome (44)	**45.5%**	20.5%
Mitochondria (424)	**64.4%**	36.6%
Nucleus (1185)	**78.0%**	71.5%
Peroxisome (47)	**23.4%**	14.9%
Vacuole (29)	**3.4%**	6.9%
Total accuracy, TP	**68.6%**	**58.1%**
Location accuracy, LP	**40.5%**	**32.3%**