Plant Protein Localization Using Discriminative and Frequent Partition-Based Subsequences

S. Vahid Jazayeri and Osmar R. Zaïane
University of Alberta
Edmonton, Canada
{jazayeri, zaiane}@cs.ualberta.ca

Abstract

The function of proteins in the living cells varies with respect to their localizations. Extracellular plant proteins are responsible for vital functions such as nutrition acquisition, protection from pathogens, communication with other soil organisms, etc. Hence, characterizing these proteins and distinguishing them from intracellular proteins is of high interest to biologists. Nonetheless, the small number of available extracellular proteins for training makes classifying them difficult and challenging. This work focuses on distinguishing extracellular proteins using partition-based subsequences, i.e., subsequences of amino acids in special partitions within the protein sequences. The use of an associative classifier in this work helps to acquire a set of accurate, small and interpretable localization rules that can be used for further biological analysis. The achievement of 98.83% F-Measure for identifying extracellular proteins shows the appropriateness of the selected features and the classification method.

1 Introduction

Proteins are one of the main structures of the living cells that conduct different processes and functions in the cell. Proteins are linear sequences of amino acids, and so far twenty amino acids have been identified in the nature. These amino acids are coded by twenty alphabetic characters [5]. Therefore, proteins can be considered as character strings of different length varying from 41 amino acids or less, for a mitochondrial protein, to 3705 or more, for an outer membrane protein. Biological experiments indicate that amino acid sequences encode information about protein structures, functions, localizations, *etc.* Many datasets of raw biological sequences are collected through genome sequencing projects, and are publicly available for researchers.

One of the important problems in the biology community is the functional classification of proteins based on their structures, localizations, or other properties. In order for proteins to accomplish a specific function, they localize in different locations inside the cell and sometimes they are transported to the extracellular space. There is a variety of localization sites within plant cells such as nucleus, mitochondria, cytoplasm, membrane, *etc.* that are generally referred to as intracellular (IC) localizations. When outside the cell, we refer to them as extra-cellular (EC). Protein sub-cellular localization is the key characteristic to study the function of proteins. In plants, EC proteins are responsible for vital functions such as "nutrition acquisition, communication with other soil organisms, protection from pathogens, and resistance to disease and toxic metals" [25]. Therefore, they are of high importance for the cells and are a target of analysis in the biology community. Herein, we particularly focus on characterizing and predicting EC proteins by learning and classifying proteins to EC or IC locations.

Localization of proteins has been a research interest for bio-informaticiens and machine learners for some time, but it is a challenging problem mainly due to the lack of training data, and when data exists, to severe imbalance in the training data. Another difficulty is the identification of appropriate features in the data to accurately localize proteins. Some have used simple distribution of amino acids (*i.e.*, protein composition), subsequences, special signatures or combinations. In this paper we introduce the idea of taking advantage of partitioning sequences of amino acids and identifying the relevant partition where some subsequences occur. These partitions appear to have discriminative power with regard to localization of proteins.

To do so, we transform the proteins that are originally represented as strings of amino acids into sets of frequent motifs extracted from these strings. Motifs are subsequences of amino acids that are frequently occurring in the collection. Then, proteins are partitioned in equal partitions and each motif in a protein is labeled by the parti-

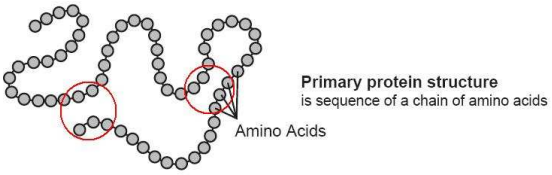


Figure 1. Structure of Protein [1]. The locations that may interact due to their close distance are circled.

tion in which it occurs. This is more complex than it appears, since each protein has to be expressed by some identified motifs, and identifying all partitions where motifs occur, given different partitioning intervals, is a hard problem. These features (*i.e.*, motif and partition pairs) are frequent subsequences associated with their discriminative partitions along the protein sequence, which we call Partition-Based Subsequence (or PBS). They constitute our input for classifying proteins.

Our inspiration comes from the following observation. Proteins are of complicated shapes in 3-dimensional space. At this level, proteins of the same class may present higher similarity than at the simple level of amino acid sequences [19]. On the other hand, it is difficult to characterize the 3-D specifications of proteins. Discovering the special regions of protein structures where frequent subsequences appear most may encode significant information about the structure of proteins. For example, EC proteins may be folded such that some regions may have biochemical effects on each other due to their close distance (as Figure 1 illustrates). Such effects may cause special patterns to be formed in these regions. This is what motivated us to discover subsequence patterns that are frequent in special regions of protein sequences.

We use an associative classifier to predict EC proteins. The reason for our choice is that associative classifiers construct an interpretable rule-based model that can be used for further biological analysis. As the popularity of SVM's [15] is increasing in the biological data mining field, we also compare our results with those of SVM. Our experiments on a biologically verified dataset, show that the localization prediction is more accurate when PBS features are used rather than simple subsequences. Moreover, it is shown that associative classifier on such feature datasets predicts the localization of proteins better than the state-of-the-art algorithms.

The rest of the paper is organized as follows: Section 2 is a review of the related work. Section 3 explains the algorithm of mining discriminative frequent partition-based subsequences. In section 4 the associative classifier for the special case of our problem is explained. Experimental results are discussed in section 5, and finally section 6 concludes the paper.

2 Related Work

Several approaches have been proposed to predict different protein localizations. These approaches differ in the features and the classification methods they have used. Generally these works can be grouped in five different categories.

For some specific cell locations, it has been shown that N-terminal signals direct proteins to their localization sites. Signals are "short subsequences of approximately 3 to 70 amino acids and can be identified by looking at the primary protein sequence" [23]. SignalP [10] and ChloroP [16] identify these signals by means of neural networks. TargetP [17] integrates the last two algorithms with some extension to predict four different localizations. The highest reported overall accuracy for these locations among these three tools is 90% which is the result of TargetP for non-plant proteins.

Textual annotations of a protein, which is available in SWISS-PROT [4], can also be used to predict protein localization. Based on lexical analysis, keywords from the textual annotations of homologous proteins are extracted. Then, a protein is represented in terms of the keywords that are contained within the annotation of the protein. Using these features, LOCKey [18] employs Multiple category classifiers and PA-SUB [23] uses different classifiers (Naïve Bayes as its default classifier) for sub-cellular localization prediction. PA-SUB with the overall accuracy of about 98% outperforms LOCKey.

It has been shown that EC and IC proteins differ in their amino acid compositions [9], *i.e.*, the relative frequency of the twenty amino acids in the sequence of a protein. Based on protein composition, predicting subcellular localization has been done by applying statistical analysis-based algorithms [9, 11, 13], SVM [20], Neural Networks [3], Markov chain models [24]. Among these approaches, the SVM-based method has reported the highest overall accuracy of 91.4% on prokaryotic proteins.

Frequent subsequences within proteins are other features used for subcellular localization. A *frequent subsequence* is a consecutive series of amino acids that appear in more than a certain number of proteins of a specific class. In this context, proteins are represented in terms of frequent subsequences that they contain. Zaïane *et al.* [25] used such features and applied SVM and boosting methods to predict EC localization. They have achieved the F-Measure of 80.4% with boosting. In another effort, they have used discriminative frequent sequential patterns as rules [25]. A *frequent sequential* pattern is of the form $*X_1 * X_2 * ... * X_n*$ where X_i is a frequent subsequence and $*$ represents a

variable-length-don't-care. The same method for localizing outer membrane proteins has been used by She *et al.* [19]. Their approach suffers from low recall although they have achieved high precision.

The last category of approaches is the combination of different methods. Zaïane *et al.* [25] have applied the boosting methods with the combination of frequent subsequences and amino acid compositions to predict EC localization. The F-Measure of their algorithm is 83.1% on plant proteins. With SVM as the learning algorithm, Li and Liu [21] predict protein locations by combining N-terminal signals and amino acid compositions. Their highest achievement is 91.9% overall accuracy on non-plant proteins. Höglund *et al.* has achieved the overall accuracy of more than 74% by combining N-terminal signals, amino acid compositions and sequence motifs [2]. PSORT [12], probably the most complete tool for predicting many different localization sites, integrates various statistical methods and classification algorithms. However, its overall accuracy is less than 66%.

The approach we propose is different from the above in some aspects. First, most of these methods are for predicting different sub-cellular localizations while we only focus on a two-class problem of EC and IC localization prediction. Second, most of them use the overall classification accuracy for evaluation, while in our case where only 4% of the data is EC, a classifier that always classifies as IC will achieve an overall accuracy of at least 96%. For evaluation we use the F-measure instead. Finally, by partition-based subsequences, we probably indirectly exploit information about the folded structures of proteins in our prediction, something that is not considered in the mentioned works.

3 Subsequence Based Feature Extraction

Proteins should be re-expressed in terms of their features extracted from the sequence of amino-acids. The features that we focus on in this paper are based on frequent subsequences or motifs because:

- "Common subsequences among related proteins may perform similar functions via related biochemical mechanisms" [19] and are of great interest to biologists.

- "Frequent subsequences capture local similarity that may relate to important functional or structural information of extracellular proteins" [22].

In mining discriminative features (patterns) for a class C of proteins, a "frequent pattern" refers to a pattern that occurs in more than a certain fraction ($MinSup$) of proteins of class C. The confidence of such a pattern (discriminative of class C) is:

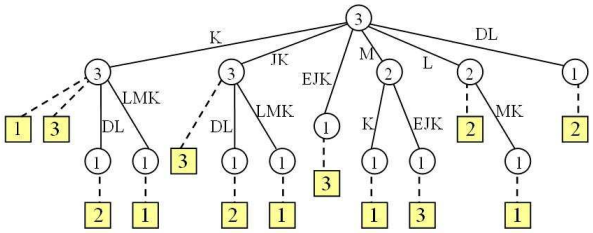


Figure 2. The GST of three strings: 1) JKLMK, 2) JKDL, 3) MEJK

$$Confidence(M) = \frac{frequency\ of\ M\ in\ class\ C}{frequency\ of\ M\ in\ both\ classes} \quad (1)$$

The less a pattern appears in the other class, the higher its confidence is. If the confidence is more than a certain threshold ($MinConf$), it is called "discriminative".

The ability of frequent subsequences to discriminate EC and IC proteins has been already studied [25]. However, we believe that the presence of a subsequence in special partitions of protein sequences might be more discriminative than the subsequence itself. For example, "$ACDE$" may be a frequent subsequence among both IC and EC proteins, thus is not distinguishing. Nonetheless, "$ACDE$" may appear in the first half of EC protein sequences while in IC proteins it may occur in the second half of the sequences. Here the association of "$ACDE$" and its respective location along proteins is a discriminative pattern. Such a pattern is called "Partition-Based Subsequence", or in short PBS. PBSs are the generalized form of simple subsequences. Simple subsequences are the PBSs whose partition is the whole protein.

If a PBS is frequent, its subsequence regardless of partition is more (or equally) frequent. Thus, to find frequent PBSs, frequent subsequences should be mined first. To mine frequent subsequences, there are algorithms based on the Generalized Suffix Tree of protein sequences (GST) [8]. Figure 2 shows the GST of three strings. As this figure shows, edges are labeled with character strings and leaf nodes are associated with an index. The concatenation of edge labels from the root to a leaf node with index i is a suffix of the ith string. Each internal node stores the frequency of the substring which is constructed by concatenating the edge labels from the root to that node. There are efficient algorithms for online construction of GST in linear time [8]. After the GST is constructed, frequent subsequences are mined through a single traversal of the tree.

Mining frequent subsequences does not guarantee that all the proteins contain at least one of the subsequences. A protein that can not be expressed by any of the frequent subsequences is called *silent*. To avoid silent proteins, $MinSup$ should be set to a low value. On the other hand, low frequency leads a huge number of subsequences, which adds complexity to the classification later. Even for small $MinSup$ values, if $MinLen$ (minimum length of motifs) is not short enough, *i.e.*, in our case less than 4, silent proteins are still observed. Because of the imbalance between the two classes, short motifs that appear in the smaller class, *i.e.*, EC, also appear in the larger class, *i.e.*, IC. Therefore, short subsequences cannot be discriminative. Considering the partitioning of protein sequences for motif occurrences is a solution to resolving silent proteins.

Since proteins differ greatly in length, the partition should be defined relative to the length (partition-based) *i.e.*, a protein sequence is divided into 2, 3 or more equal partitions. The presence of frequent subsequences in different partitions is investigated. If protein sequences are assumed to be divided into P partitions, the presence of a subsequence S in the i'th partition of proteins, where $1 \leq i \leq P$, is denoted by $S_{i/P}$. The problem is to find subsequences S with their particions, *i.e.*, values for i and P, such that $S_{i/P}$ is frequent and discriminative with respect to $MinSup$ and $MinConf$.

In other words, our approach looks at a partition of 100%, then two partitions of 50%, then three partitions of 33% and so on. To explain the algorithm of mining PBSs, a Partition-Frequency Table of a subsequence S should be defined first. In this table, the P'th row is an array of length P. The value in row P and column i indicates $frequency(S_{i/P})$. The first row of this table shows the frequency of subsequence S where each protein is considered as only one sequence (no partitioning). The last row of this table is related to partitioning proteins to a maximum number, namely $MaxPart$, which is given by the user. If $MaxPart$ is chosen to be 3, for example, each frequent subsequence possesses a Partition-Frequency Table which is filled as Figure 3 illustrates.

$frequency(S)$		
$frequency(S_{1/2})$	$frequency(S_{2/2})$	
$frequency(S_{1/3})$	$frequency(S_{2/3})$	$frequency(S_{3/3})$

Figure 3. Partition-Frequency table of a subsequence S where partitioning proteins to 1, 2, and 3 is investigated (MaxPart=3)

After this table is filled with frequencies, the partitions with enough frequency make a frequent PBS. Filling in any slot of this table for all frequent subsequences is a complex task. However, there is no need to fill the whole table. Indeed, if processed top-down, some partitions can be ignored

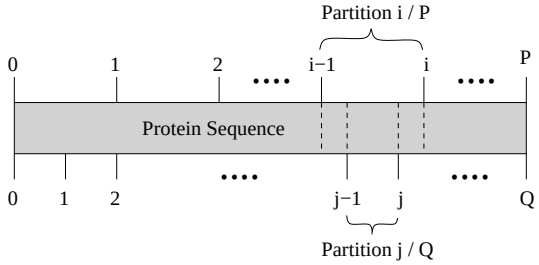


Figure 4. Illustration of Equation 2

if their subsuming partition already indicates infrequency. For example, partition 1/2 (first half) encompasses partition 2/4 (second quarter). Therefore, if a subsequence S is not frequent in the partition 1/2, it cannot be frequent in partition 2/4. Assuming that $S_{i/P}$ is not frequent, $S_{j/Q}$ is also infrequent for all smaller partitions j/Q that:

$$Q > P \;\; And \;\; \frac{i-1}{P} \leq \frac{j-1}{Q} \;\; And \;\; \frac{j}{Q} \leq \frac{i}{P} \qquad (2)$$

After a frequent PBS $S_{i/P}$ of class C is found, its occurrence in the proteins of the other class is counted and then its confidence is computed using Equation 1. If the confidence is less than a $MinConf$, the PBS is considered non-discriminative and is removed. In addition, some other filtering techniques are applied because of the large number of frequent discriminative PBSs. To filter PBSs, each protein is restricted to pick only N number of best PBSs that match with it. If a PBS is not selected by any protein, it is removed.

For selecting its N best features, a protein ranks its PBSs based on different metrics. In our approach, confidence, length and frequency are respectively the primary, secondary and final ranking metric. For example, between two PBSs with equal confidence, the longer one has a higher rank. Other metrics and priorities can be set by the user depending on the importance of feature properties.

In order to analyze the effect of different partitions on the discriminative power of features, $MaxPart$ is considered as an input parameter in this paper. However, $MaxPart$ can be set automatically: Starting from a frequent subsequence S, *i.e.*, $S_{1/1}$, if a frequent PBS $S_{i/p}$ reaches the confidence of 100%, all its sub-partitions do not need to be considered. For example, if $S_{1/2}$ is 100% confident, $S_{2/4}$ is also of the same confidence but with less frequency. Top-down partitioning continues until all the mined PBSs at the last partitioning are either infrequent or 100% confident.

As the discriminative and frequent PBSs are mined, the feature dataset is constructed in the form of a transactional dataset. Each protein is represented by a transaction with its PBSs as items. Henceforth, we refer to PBSs and proteins as items and transactions.

4 Associative Classifier

An associative classifier [14] integrates methods for association rule mining and classification. The input is a transactional dataset and the output is a set of rules of the form $X \Rightarrow C$, where X is a frequent itemset (a PBS set in our case) and C is a cell location. Thus, finding classification rules for a class C includes discovering frequent itemsets X with a support greater than a threshold ($MinSup$), and then pruning rules based on a confidence threshold ($MinConf$) and some other criteria. Support of a rule $X \Rightarrow c$ is the fraction of proteins from class C that can match X. The confidence of this rule is similar to Equation 1.

Building an Associative Classifier (Training Phase)

In our feature dataset there is a large number of proteins with long transactions. For example, with the combination of parameters that gave us the best prediction, the transactions representing proteins averaged a length of 55 and almost 10% of the transactions had a length between 350 and 550 PBSs, which is remarkably long. In such situations, so many frequent itemsets (potential rules) are mined that the classification algorithm has to consider effective means of selecting appropriate classification rules. Moreover, before rule pruning, excessive memory is required.

A discovered frequent itemset X from a class C directly corresponds to the rule $X \Rightarrow C$. As explained below each frequent itemset is potentially abridged, then the rule confidence is used to prune those rules that are less confident than $MinConf$. Other pruning strategies can be applied.

4.1 Abridging Itemsets

Itemsets could be redundant and a simplification of some itemsets can be helpful. Abridging consists of eliminating from an itemset any item that is already represented. In our context, items are motifs. In other words, If a motif is represented by another super-motif in the same itemset, the motif can be removed. If M_1 is a submotif of M_2 (M_2 is called super-motif) and is written $M_1 \preceq M_2$ if and only if all the proteins that match M_1, also match M_2. For example, $"JKLM"_{1/4} \preceq "KL"_{1/2}$: a protein containing "$JKLM$" in its first quarter has trivially contained "KL" in the first half.

The definition of sub-motif is as follows:

$Tj/Q \preceq S_{i/P} \Leftrightarrow S$ is a subsequence of T, and partition i/P surrounds partition j/Q, *i.e.*,

$$Q \geq P \;\; And \;\; \frac{i-1}{P} \leq \frac{j-1}{Q} \;\; And \;\; \frac{j}{Q} \leq \frac{i}{P}$$

Therefore, if the predicate of a rule contains two motifs M_1 and M_2 where $M_1 \preceq M_2$, the rule is simplified by removing M_2. For example $"KL"_{1/2}, "JKLM"_{1/4} \Rightarrow EC$ is simplified to $"JKLM"_{1/4} \Rightarrow EC$.

4.2 Computing the Confidence of a Rule

The confidence of $X \Rightarrow C$, where X is an itemset, depends on the frequency of X in both classes. The frequency in class C is available as soon as X is mined as a frequent itemset of class C. The important issue is counting its frequency in the other class. For fast and efficient computation of this frequency, the following approach is used:

1. All the items (PBSs) are coded to unique numbers. The feature dataset is then transformed in transactions of numerical items.

2. Items in each transaction are sorted in the increasing order.

3. Transactions of each class are inserted into two Trie structures, namely $Trie\{EC\}$ and $Trie\{IC\}$. The internal nodes in such a trie store the items. The direct path from the root to a leaf node is equivalent to an itemset. For computational efficiency, the number of leaves of the sub-trie rooted at a node m is also stored in node m.

4. Whenever an itemset X of class C is generated, we sort X in the increasing order of its items.

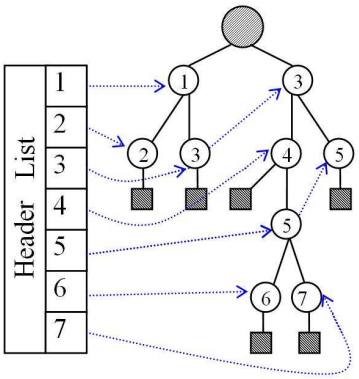
5. Find all the nodes n of $Trie\{other\ class\}$ that match the first item in X. Rooted at nodes n, traverse (Depth-First) the sub-tries to find the matches with X. Whenever X matches the trie at a node m, the algorithm counts the number of leaves of the sub-trie rooted at node m (which is stored in node m) and stops traversing deeper down the node m, and tries matching X with other branches. For fast finding of Nodes n, which match the first item of X, we use a list, called header list, which contains all the items in the trie. By following the pointers from an item I in the header list, we can find all the occurrences (matches) of item I in the trie. Figure 5 shows how a trie represents transactions.

4.3 Pruning the Rules

The minimum confidence requirement ($MinConf$) prunes many rules. However, the number of confident rules is still large and some other pruning techniques are required:

Localization	Transaction
IC	$1, 2$
IC	$1, 3$
IC	$3, 4$
IC	$3, 4, 5, 6$
IC	$3, 4, 5, 7$
IC	$3, 5$

(a) Feature dataset of EC proteins



(b) $Trie\{IC\}$

Figure 5. The Trie representation of IC protein transactions

1. If the confidence of a rule, $X \Rightarrow C$, reaches 100%, any expansion of its predicate results in a rule with 100% confidence too, *i.e.,* $X \cup Y \Rightarrow C, conf = 100\%$, where X and Y are two disjoint itemsets and C is a class label. In this case, keeping the first rule suffices, and any other expanded rule is not useful. This technique prunes many rules on the fly especially using a depth-first search of the itemset lattice. Indeed, expansions of a rule fall in the deeper levels of the lattice, and thus we can stop going deeper in the recursion path as soon as we finds a 100% confident rule.

2. If R is a rule with confidence $conf$, all the sub-rules of R with confidence less than $conf$ should be pruned. R_1 is a sub-rule of R_2 (R_2 is called super-rule) and is written $R_1 \sqsubseteq R_2$ if and only if any protein that matches R_1 can also match R_2 (*i.e.,* R_2 is more general), further, R_1 and R_2 should imply the same class. In other words if R_1 is:

$$n_1, n_2, \ldots n_i \Rightarrow C \ \ with\ conf = \alpha$$

and R_2 is:

$$m_1, m_2, \ldots m_j \Rightarrow C \ \ with\ conf = \beta$$

Then $R_1 \sqsubseteq R_2$ if and only if:

   (a) $i \geq j$, *i.e.,* the length of a sub-rule can not be less than that of its super-rule.

   (b) For each item m_b $(1 \leq b \leq j)$, there must be an item n_a $(1 \leq a \leq i)$ such that $n_a \preceq m_b$. *i.e.,* at least one sub-motif of each m_b must be found in the sub-rule.

   Therefore, R_2 is a more general rule and if $\alpha \leq \beta$ then R_2 is much worth keeping rather than R_1. For example suppose R_1 is:

$$"JKLM"_{2/4}, "PQRST"_{4/5} \Rightarrow EC \ \ with\, conf = 60\%$$

   and R_2 is:

$$"KL"_{1/2} \Rightarrow EC \ \ with\, conf = 90\%$$

   R_1 should be removed and R_2 kept because $"JKLM"_{2/4} \preceq "KL"_{1/2}$.

When a new rule is to be added in a rule set, this rule has to be compared to all the older rules. Any older rule that is a removable sub-rule of the new rule, is removed. If the new rule is a removable sub-rule of any older rule then it is not added in the set. The data structure used for this rule set is also a Trie similar to Figure 5.

Evaluating Associative Classifier (Testing Phase)

Given an unknown protein P, a set of rules can match P when the antecedent of a rule applies for the PBSs representing P. These rules can localize the protein as EC or IC. To decide between the two classes, the average confidence of the matching rules of each class is considered. The class with the highest average confidence is the class label assigned to P. There is an exceptional case for which confidence averaging is not used. Whenever a rule with 100% confidence matches a test protein P, P is assigned the class label of that rule as long as there is no other 100% confident rule of the other class.

In few cases, a test protein cannot match any rule from any class. Such a protein is called *Undecided* and is de facto classified as EC, the rare class. The reason is that the large community of IC proteins is more likely to include enough samples from different distributions of IC proteins. Thus, IC proteins are well learnt by the classifier and are not left undecided. In contrast, the small volume of EC proteins lacks enough samples to lead the classifier to learn the patterns of EC proteins. Hence, it is more likely for an undecided protein to be EC.

To strengthen our argument, we clustered all the proteins in our training set (after stripping them from their labels) and identified outliers. The pattern of an outlier is very infrequent as it belongs to no cluster and is not learnt by the classifier. Revealing the labels demonstrated that most of these outliers were in fact EC. Thus, undecided proteins should be, and are, classified as extracellular.

5 Experimental Results

In this section, we evaluate the discriminative power of partition-based subsequences to predict the two subcellular localizations: EC and IC. The main classification algorithm that has been the focus of this work is the associative classifier. However, because of the growing interest in SVM and its strong ability to classify high dimensional data, we compare our results with those of SVM.

5.1 Dataset and Evaluation Methodology

We performed our method on a plant protein dataset from Proteome Analyst Project [23] at the University of Alberta. The dataset is constructed from SWISS-PROT. After cleaning the data, *i.e.,* removing repetitive or defected proteins which contain nonexistent amino acids, 3149 proteins remained, 4% of which are EC proteins.

To evaluate the performance of classifiers, *Overall Accuracy* is often used. However, this is usually inappropriate particularly with imbalanced data. In our case with 96% of proteins being IC, a classifier that always classifies as IC achieves the overall accuracy of 96% while no EC proteins are correctly classified. Instead, we use precision, recall and F-measure with respect to EC (*i.e.,* the target class). Based on the confusion matrix shown in table 1, Precision(P), Recall(R) and F-Measure (a harmonic average of precision and recall) of EC prediction are defined as:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad FMeasure = \frac{2PR}{P + R}$$

	Actually EC	Actually IC
Predicted as EC	TP	FP
Predicted as IC	FN	TN

Table 1. Confusion Matrix

To have a more reliable evaluation, all the classification experiments are based on a 3-fold cross validation. The dataset is divided into three equal parts (folds) such that the distribution of EC and IC proteins in each fold does not change. Each run takes two folds for training and the other fold for testing. In the end, the f-measures from each run are averaged as EC prediction f-measure. To be fair, the exact same folds are used for both classifiers: Ours and SVM.

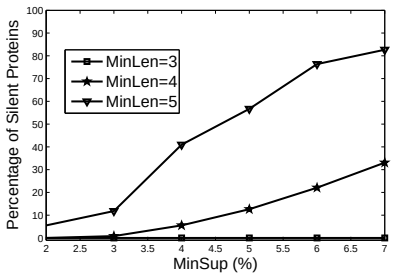
5.2 Mining Frequent Partition-Based Subsequences (PBS)

Mining frequent PBSs depends on the parameters N, $MinConf$, $MinSup$, $MaxPart$ (max. number of partitions) and $MinLen$ (min. length of subsequences). A proper setting for these parameters should prevent *silent* proteins. $MinLen$ and $MinSup$ are the most sensitive parameters, because lengthiness and high frequency could overshadow short and less frequent motifs that are actually discriminating and expressing those silent proteins. For the other parameters, we use the following setting:

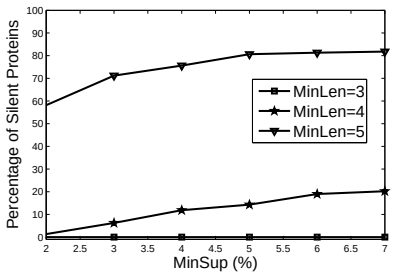
- $N = 1$: Each protein selects its top best PBS. The larger the N, the longer the transactions in the feature dataset and the harder their classification.
- $MinConf = 50\%$: If the confidence of a PBS is less than 50%, it is useless because it is more frequent in the other class.
- $MaxPart = 10$: $MaxPart$ was initially set to 10 (*i.e.,* partitioning to 1 (no partitioning), 2, ... up to 10). Based on further experiments that are discussed in section 5.3, we found out that 9 is a good value. However, $MaxPart$ of 9 and 10 generate such very close results that the experiments for $MaxPart = 9$ is not illustrated in the paper.

Figure 6 demonstrates the effect of $MinSup$ and $MinLen$ on the number of silent proteins in EC and IC classes. Note that $MinSup$ starts from 2% because less than this value is equivalent to the absolute frequency of 1 for the 127 proteins of EC class, which is meaningless. Based on Figure 6, at $MinLen = 3$ no silent protein is found for different $MinSup$ values. Selecting the best $MinSup$, where $MinLen$ is fixed to 3, is based on the length of subsequences. The longer subsequences convey more information and are preferred. Figure 7 plots the average length of EC and IC frequent subsequences in the mined PBSs. It shows that increasing $MinSup$ results in shorter subsequences as they are more frequent than long ones. At $MinSup = 2\%$, longer subsequences are mined, however, we choose $MinSup = 3\%$ because 2% is a very low support (would mean an absolute frequency of 2 for EC). Moreover, at $MinSup = 3\%$ the average length of subsequences in both classes agree.

Therefore, $MinSup = 3\%$ and $MinLen = 3$ are the best settings. With the parameters set as mentioned, a feature dataset with very long transactions is produced. Table 2 shows some statistics about the feature dataset.



(a) EC Silent Proteins



(b) IC Silent Proteins

Figure 6. Percentage of silent proteins for different min. support and min. length

Total # of PBSs	Min-Trans-Len	Max-Trans-Len	Ave-Trans-Len
738	1	553	54.24

Table 2. The number of PBSs, and the size of transactions in th feature dataset

	MinSup of Rules			
	2%	3%	4%	5%
Average Precision (%)	97.71	97.71	83.39	80.06
Average Recall (%)	100	100	100	100
Average F-Measure (%)	98.83	98.83	90.69	88.74

Table 3. Evaluation of associative classifier wtih different minimum supports (3-fold cross validation)

5.3 Associative Classifier

We used an efficient, publicly available implementations of Eclat algorithm [6] to mine frequent itemsets. Eclat uses a depth-first traversal of the itemset lattice.

The feature dataset we mentioned in 5.2 is used for classification. The result of the classifier is expected to depend on parameters $MinSup$ and $MinConf$. However, the experiments showed that $MinConf$ does not influence the accuracy of EC prediction. The reason is that PBSs are so discriminative that each individual PBS of a class is a rule of size 1 with the confidence equal or very close to 100%. Based on our pruning techniques, 100% confident rules do not expand to longer rules, and the other individual PBS rules with confidence close to 100%, do not reach a higher confidence in association with other PBSs in most cases. Thus, the expansion of such rules is also pruned. What remains is the population of rules of size 1 as well as very few longer rules, all with very high confidences. This is why $MinConf$ is ineffective. Table 3 shows that regardless of $MinConf$, the minimum support of 2-3% results in 98.83% EC prediction accuracy (F-Measure).

However, we cannot claim that the best parameter values are found. In fact, the effect of $MaxPart$ on the prediction of EC proteins is not considered yet. We built different feature datasets with $MaxPart$ ranging from 1 to 14. Then, the classification on the dataset derived from each $MaxPart$ setting is evaluated. Figure 8 shows the comparison of the highest F-Measure of the classifiers trained on the datasets related to different $MaxPart$ values. According to the diagram, as more partitions along protein sequences are considered, the power of PBSs to discriminate the two classes becomes higher. Moreover, beyond 9 partitions, we observe no increase in the F-Measure.

Note that with $MaxPart = 1$, no partitioning is performed on the sequences. Therefore, PBSs are exactly the same as discriminative frequent subsequences regardless of their location along the sequences. For such features, the F-Measure of 96.16% is achieved, which clearly outperforms the F-Measure of 83.1% of the state-of-the-art algorithm for EC localization prediction [25]

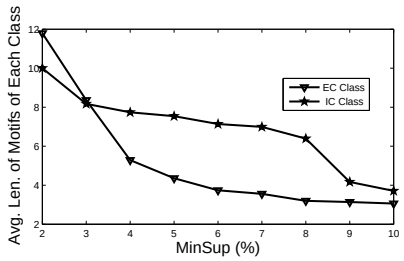


Figure 7. Average length of subsequences in the PBSs mined for proteins of each class

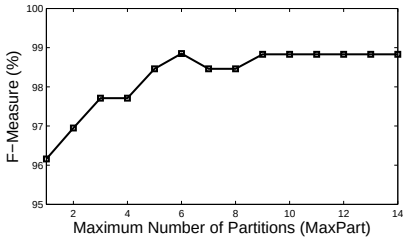


Figure 8. The effect of different partitionings of PBS mining on the prediction accuracy (F-Measure)

5.4 SVM-based Classifier

With data represented as vectors in the multi-dimensional feature space, SVM [15] finds the hyperplane that best separates instances of two classes. The hyperplane divides the feature space into two sub-spaces each for one class. Unknown data is simply classified based on the sub-space it is located in.

To use SVM, our feature dataset, which obtained the best result with our approach, is transformed from transactional to a relational dataset with a fixed dimensionality (738). This is simply done by creating a matrix in which each column represents a PBS, and each row represents a protein as a binary vector.

We used LIBSVM, an available implementation of SVM [7]. In SVM, there are two important parameters to be set: the kernel function and the parameter C (Cost). Table 4 shows the F-Measures obtained from different parameter settings. The highest F-Measure of 84.87% is achieved by using Radial Basis Function kernel with $C = 1000$. Note that polynomial kernel function (degree 2 and 3) was also tried but it could never predict EC proteins.

	F-Measure		
	Linear Kernel	Sigmoid Kernel	Radial Basis Function kernel
C = 1	85.1	4.58	0
C = 10	84.49	4.58	4.58
C = 100	84.49	4.58	71.70
C = 1000	84.49	4.58	84.87

Table 4. SVM Classification using different Kernels

6 Conclusion

In this paper, we proposed a new discriminative feature for predicting extracellular proteins. Partition-Based Subsequences have strong ability to discriminate between the proteins of different localizations. Moreover, they seem to encode more information about the structure of proteins by showing the regions along the protein sequences where special subsequences appear most. We applied an associative classifier on the feature datasets. With some short, interpretable and highly confident rules, the F-Measure of 98.83% for predicting extracellular proteins was achieved, significantly above the current state-of-the-art. In our case, associative classifier outperforms SVM on the same feature space with a large difference in the F-Measure. The application and evaluation of our approach on other protein localizations remain as a future work

References

[1] National human genome research institute. http://en.wikipedia.org/wiki/Image:Protein-structure.png.

[2] Höglund A., Dönnes P., Blum T., Adolph H. W., and Kohlbacher O. Multiloc: prediction of protein subcellular localization using n-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*, 22(10):1158–1165, 2006.

[3] Reinhardt A. and Hubbard T. Using neural networks for prediction of the subcellular location of proteins. *Nucleic acids research*, 26(9):2230–2236, 1998.

[4] Boeckmann B., Bairoch A., Apweiler R., Blatter M. C., Estreicher A., Gasteiger E., Martin M. J., Michoud K., O'Donovan C., Phan I., Pilbout S., and Schneider M. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res*, 31(1):365–370, January 2003.

[5] Lukas K. Buehler and Hooman H Rashidi. *Bioinformatics Basics: Applications in Biological Science and Medicine*. CRC Taylor and Francis, 2nd edition edition, 2005.

[6] Borgelt C. Efficient implementations of apriori and eclat. In *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI03)*, Melbourne, Florida, USA, 2003. software available at http://fuzzy.cs.uni-magdeburg.de/ borgelt/software.html.

[7] Chang C. C. and Lin C. J. Libsvm : a library for support vector machines, 2001. software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[8] Gusfield D. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, January 1997.

[9] Nakashima H. and Nishikawa K. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *Journal of Molecular Biology*, 238(1):54–61, 1994.

[10] Nielsen H., Engelbrecht J., Brunak S., and Von Heijne G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10(1):1–6, 1997.

[11] Cedano J., Aloy P., Perez-Pons J. A., and Querol E. Relation between amino acid composition and cellular location of proteins. *Journal of Molecular Biology*, 266:594–600, 1997.

[12] Nakai K. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics*, 14:897–911, 1992.

[13] Chou K.C. and Elrod D.W. Protein subcellular location prediction. *Protein Engineering*, 12(2):107–118, 1999.

[14] Antonie M-L., Zaïane O.R., and Coman A. *Mining Multimedia and Complex Data*, volume Lecture Notes in Artificial Intelligence 2797, chapter Associative Classifiers for Medical Images, pages 68–83. Springer-Verlag, 2003.

[15] Cristianini N. and Shawe-Taylor J. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[16] Emanuelsson O., Nielsen H., and Von Heijne G. Chlorop, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science*, 8:978–984, 1999.

[17] Emanuelsson O., Nielsen H., Brunak S., and Von Heijne G. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *Journal of Molecular Biology*, 300:1005–1016, 2000.

[18] Nair R. and Rost B. Inferring sub-cellular localization through automatic lexical analysis. In *Proceedings of the tenth International Conference on Intelligent Syetems for Molecular Biology*, pages 78–86. Oxford University Press, 2002.

[19] She R., Chen F., Wang K., Ester M., Gardy J. L., and Brinkman F. S. L. Frequent-subsequence-based prediction of outer membrane proteins. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–445, New York, NY, USA, 2003. ACM.

[20] Hua S. and Sun Z. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001.

[21] Li Y. and Liu J. Predicting subcellular localization of proteins using support vector machine with n-terminal amino composition. In *Advanced Data Mining and Applications*, Wuhan, China, 2005.

[22] Wang Y. A database for proteomic analysis of extracytosolic plant proteins. Master's thesis, Department of Computing Science, University of Alberta, Fall 2004.

[23] Lu Z. Predicting protein sub-cellular localization from homologs using machine learning algorithms. Master's thesis, Department of Computing Science, University of Alberta, 2002.

[24] Yuan Z. Prediction of protein subcellular locations using markov chain models. *FEBS Letters*, 451(1):23–26, 1999.

[25] Zaïane O. R., Wang Y., Goebel R., and Taylor G. J. Frequent subsequence-based protein localization. In *BioDM*, pages 35–47, 2006.