Frequent-Subsequence-Based Prediction of Outer Membrane Proteins *

Rong She, Fei Chen, Ke Wang, Martin Ester
School of Computing Science
Simon Fraser University
rshe, fchena, wangk, ester@cs.sfu.ca

Jennifer L. Gardy, Fiona S. L. Brinkman
Department of Molecular Biology and Biochemistry
Simon Fraser University
jlgardy, brinkman@sfu.ca

ABSTRACT
A number of medically important disease-causing bacteria (collectively called Gram-negative bacteria) are noted for the extra "outer" membrane that surrounds their cell. Proteins resident in this membrane (outer membrane proteins, or OMPs) are of primary research interest for antibiotic and vaccine drug design as they are on the surface of the bacteria and so are the most accessible targets to develop new drugs against. With the development of genome sequencing technology and bioinformatics, biologists can now deduce all the proteins that are likely produced in a given bacteria and have attempted to classify where proteins are located in a bacterial cell. However such protein localization programs are currently least accurate when predicting OMPs, and so there is a current need for the development of a better OMP classifier. Data mining research suggests that the use of frequent patterns has good performance in aiding the development of accurate and efficient classification algorithms. In this paper, we present two methods to identify OMPs based on frequent subsequences and test them on all Gram-negative bacterial proteins whose localizations have been determined by biological experiments. One classifier follows an association rule approach, while the other is based on support vector machines (SVMs). We compare the proposed methods with the state-of-the-art methods in the biological domain. The results demonstrate that our methods are better both in terms of accurately identifying OMPs and providing biological insights that increase our understanding of the structures and functions of these important proteins.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications - Data Mining

Keywords
Classification, association rule, support vector machine, subcellular localization, outer membrane protein

*Research was supported in part by a research grant from the Networks of Centres of Excellence/Institute for Robotics and Intelligent Systems, and in part by a research grant from the Natural Science and Engineering Research Council of Canada.

SIGKDD '03, August 24-27, 2003, Washington, DC, USA.

1. INTRODUCTION
Understanding the biology of disease-causing, or pathogenic, organisms can have a great impact on improving the quality of human life. With the progress of high throughput genome sequencing projects, biologists have accumulated huge amounts of raw biological sequences that are publicly available. In order to gain a better understanding of the structure and function of such sequences, one critical task facing the biology community is to correctly classify these sequences into corresponding functional families. One of the most important protein classification problems is to predict the subcellular localization of proteins [5]. For proper functioning, a protein has to be transported to the correct intra- or extra-cellular compartments in a soluble form, or attached to a membrane that surrounds the cell, hence the subcellular localization of a protein plays a key role with regard to its functions.

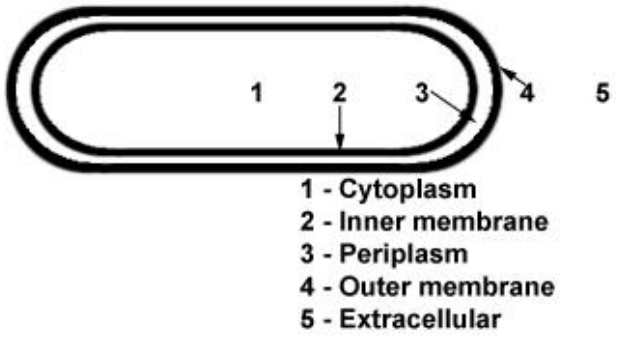

Figure 1. The Gram-negative bacterial cell. *5 primary localization sites are noted.*

Studies of a family of bacteria, collectively known as Gram-negative bacteria, have shown that such bacteria have a distinct cell structure that presents an interesting challenge for localization prediction. While many bacteria have only 3 primary localization sites, in a Gram-negative bacterial cell a protein may be resident at one of 5 primary localization sites, as illustrated in Figure 1. Proteins are synthesized in the cytoplasm and may remain there, or be transported to the inner membrane, the periplasm, the outer membrane, or the extracellular environment. Most other bacteria, as well as all animal and plant cells, do not have the additional outer membrane structure, and so methods for identification of proteins on the surface of other such cells are not directly applicable.

Biological experiments indicate that the information required to direct a protein to any localization site is primarily encoded in the

protein's amino acid sequence. For example, the presence of a string of amino acid residues in a protein that forms a structure known as a transmembrane α-helix is indicative of a protein resident at the inner membrane. Such α-helix structures have a very characteristic sequence and current algorithms for their detection in a given protein sequence are very accurate. However, the integral outer membrane proteins, characteristic of Gram-negative bacteria, do not consist of transmembrane α-helices, but rather form antiparallel β-strands that form a barrel shape. Such proteins are therefore also referred to as β-barrel outer membrane proteins (Figure 2) [20]. In this paper, we study the problem of identifying these outer membrane proteins (OMPs) from sequence information alone, and its application to localization classification.

Figure 2. A β-barrel outer membrane protein. *The central barrel shape, formed by antiparallel β-strands, rests in the outer membrane. The aromatic amino acids shown form a "girdle" to anchor the protein in the membrane. The β-strands are connected by short stretches of amino acid sequences (turns) at the inner, or periplasmic side, and longer stretches (loops) at the outer, or extracellular side.*

The prediction of OMPs is of great interest to biologists for several reasons. Many Gram-negative bacteria are medically important pathogens that cause a number of different diseases, including food poisoning, typhoid fever, E. coli, and other food- and water-borne diseases, stomach ulcers, meningitis, gonorrhea, cholera and plague. Some of these Gram-negative bacterial family members also cause diseases that affect other animals and plants of agricultural interest and others are of environmental interest for their bioremediation properties. Because OMPs are exposed on the surface of these bacterial cells, they represent primary drug and vaccine targets since it is usually easier to develop drugs against the surface of a disease-causing bacterial cell. In addition, such surface-exposed proteins are potentially useful as part of diagnostics to detect the bacteria. Such bacterial detection systems are useful for diagnosing disease and for detection of bacteria in the environment.

The ability to identify such potential targets from sequence information alone would allow researchers to quickly prioritize a list of proteins for further study. Additionally, OMPs represent a class of proteins that exhibit high similarity at the 3-dimensional level but little at the level of the amino acid sequence itself, and it remains difficult to characterize the factors that cause a protein to take its 3-dimentional shape. Being able to predict OMPs from sequence information alone will not only assist in genome annotation, functional classification, and drug discovery; determination of the relevant sequence patterns may also provide insights into the biology of this important class of proteins.

Two notable characteristics present in this practical application make the problem interesting and challenging. First, because of the medical significance of OMPs, and the lengthy time it takes to further study a prioritized drug, vaccine or diagnostic target in the laboratory, biologists want to be fairly sure about the certainty of a sequence being an OMP when it is classified as so by our classifier. That is, our goal is to maximize the precision of our outer membrane predictions while maintaining the corresponding recall at a reasonable level. This is very different from many data mining applications, where the overall classification accuracy for all classes is used as a performance measure. It is also different from typical rare-class classification problems, where an important consideration is to cover as many rare-class samples (e.g., responders in direct marketing or intruders in network intrusion detection) as possible, at the expense of covering some other-class samples, i.e., favoring recall over precision. Secondly, biologists are very interested in identifying the most significant subsequences that discriminate OMPs from non-OMPs. This requires techniques that could produce such patterns in order for biologists to make further analyses.

In this paper, we apply data mining techniques to solve the OMP prediction problem. Our approach is based on two biological observations. First, common subsequences among related proteins may perform similar functions via related biochemical mechanisms. Second, OMPs have the general structure of alternating "turns", "strands" and "loops" within the β-barrel shape, as illustrated in Figure 2. Characteristic sequence residues occur in these different regions of the protein, such as the aromatic residues are located near turns, followed by particular residues that prefer to form the "strand" structures. Thus a reasonable approach is to extract similar patterns that occur in many OMPs and search for the combinations that distinguish OMPs from non-OMPs. We exploit the notion of *frequent subsequences* studied in association rule mining to capture such similarities. A frequent subsequence is a consecutive subsequence of amino acids that occurs in many OMPs. However, frequent subsequences alone do not work because they may also occur in non-OMPs and may not generalize well to proteins not contained in the training set. We present two methods of using frequent subsequences for identifying OMPs. One uses frequent subsequences to construct classification rules for OMPs, and the other uses frequent subsequences as features for a support vector machine (SVM) [21] that will search for the hyperplane to separate the two classes. Our experiments on a biologically verified dataset show that these methods dramatically outperform the state-of-the-art methods developed by biologists for OMP prediction.

The rest of this paper is organized as follows. Section 2 briefly discusses the related work. Section 3 introduces the dataset used in our experiments, as well as the evaluation measures that we applied in our research. Section 4 describes the details of the rule-based method and Section 5 describes the SVM-based method. Section 6 discusses the experimental results and the biological significance of our methods. Section 7 concludes the paper and suggests directions for future research.

2. RELATED WORK

2.1 Work on Related Problems

Several publicly-available localization predictors exist in the biological domain, all based on different sequence features and capable of predicting different localizations. As it has been shown that intracellular and extracellular proteins differ significantly in their amino acid composition [15], methods based on protein amino acid composition have been developed to predict proteins of cytoplasmic, periplasmic, and extracellular localizations, such as neural networks [17], Markov chain models [27] and SVMs [6]. In particular, SVMs have achieved a classification accuracy of 91.4% for prokaryotic organisms.

However, these methods are limited to predicting three out of five classic subcellular localizations present in a Gram-negative bacterial cell and do not predict OMPs. In fact, even for the predictive methods that are capable of identifying OMPs, precision remains poor [4, 8, 14, 19, 25, 28]. Furthermore, the datasets used to train and evaluate these existing methods are often small and not manually curated. They are typically assembled based on third-party annotations in the SWISSPROT database [2], annotations which are not verified in some cases and may present incorrect information.

To date, little research has been done on the prediction of OMPs. Scientists have previously used neural network-based methods [4, 8], hydrophobicity analysis [19], and combinations of methods, including homology analysis and amino acid abundance [25, 28], to varying degrees of success. The most recent approach, reported by Martelli et al. [14] is, to date, the most successful attempt at OMP classification. They used a hidden Markov model (HMM) to represent the prototypes of OMPs, as it is known that each amino acid residue of a β-barrel membrane protein can be categorized into one of three types: outer/extracellular loops, transmembrane β-strands and inner/periplasmic turns. The HMM was trained on all non-redundant β-barrel OMPs whose 3-dimentional structure has been determined experimentally, however there were only 12 of such proteins, which is not a large dataset. Once the HMM was trained, classification was performed by computing the probability of the protein sequence being emitted by the model. They reported a fairly good recall of 84% for OMP prediction (called accuracy in their paper) on their test dataset.

Unfortunately, none of the above methods for OMP prediction are publicly available on the Internet. Additionally, most methods were trained on small datasets with localization information typically derived from third-party annotations – information that can be inaccurate. The methods, in many cases, were not tested on a dataset of known OMPs - rather they were used to screen genomes or a list of putative OMPs, and the number of OMPs found was reported. Thus, there is no way to critically evaluate any methods other than Martelli et al's HMM. Furthermore, all previous protein localization research evaluated the classification performance based on overall accuracy and weighted all locations equally. In our research, performance evaluation is focused on the precision of OMP prediction. In terms of this measure, the method used by Martelli et. al obtained a very low precision of 46% on their dataset, as calculated by us based on other reported measures in their publication.

2.2 Work Related to Our Methods

In data mining research, many schemes that use frequently occurring itemsets in classification have been studied [1, 13, 24]. However, these techniques are concentrated on transactional datasets. Lesh et al [11] presents a technique for mining frequent subsequences which satisfy some user-specified constraints. These constraints are intended to select a subset of frequent subsequences as features for classification. Their work focuses on efficient feature mining but not on building a classifier.

A growing interest in the use of SVMs in bioinformatics has emerged recently and resulted in research exploring string-based (sequence-similarity) kernels for SVM classification. Deshpande and Karypis [3] evaluate several widely used biological sequence classification algorithms: K-nearest neighbor, Markov model and SVMs, where biological sequences are modeled as vectors in the feature space of subsequences up to a given length. The feature space chosen by Vert [22] consists of all potential subsequences. Each feature is weighted by its inversed probability density evaluated under different probability models. The intuition behind this is that the rarer the subsequence is, the more its occurrence in two sequences increases the similarity between them. Leslie et al [12] considers all possible subsequences of a fixed length k as features, which are weighted by the number of occurrences in a sequence. Our feature space consists of only frequent subsequences that occur in at least some minimum fraction of OMPs. This dramatically reduces the number of features compared with the feature space comprising all potential subsequences. Later experiments also show SVMs trained in such a feature space can perform better.

3. DATASET AND EVALUATION METHODOLOGY

3.1 Dataset

To critically evaluate the effectiveness of OMP prediction methods, we created a dataset that represents the largest available set of Gram-negative bacterial proteins with experimentally determined subcellular localizations (available at http://www.psort.org/dataset). This dataset was created by extracting all Gram-negative proteins with an annotated subcellular localization site from the SWISSPROT database (http://us.expasy.org/sprot/). The annotated localization sites were then confirmed through a manual search of the literature, and those proteins with an experimentally verified localization site were added to the dataset. Being the largest dataset of its kind, and with the subcellular localization of each protein confirmed by biological experiments, this dataset is of excellent quality and provides a reliable means to evaluate different methods.

The dataset contains protein sequences of variable lengths. The longest sequence consists of 3705 amino acid residues and the shortest sequence has a length of only 50. Two classes are present in this dataset: OMPs and non-OMPs. They are referred to as the

"OM" class and "NOM" class, respectively. The distribution of these two classes is imbalanced, with 27% being "OM" and 73% being "NOM". The details are shown in Table 1.

Table 1. Gram-negative Bacterial Membrane Protein Dataset

Data	# of Sequences	% of each class	Min. Length	Max. Length	Ave. Length
OM	427	27.4%	91	3705	571.1
NOM	1132	72.6%	50	1034	256.8
Total	**1559**				**342.9**

3.2 Classifier Evaluation Methodology

The performance of a classifier is usually measured by classification *accuracy*, *precision* and *recall*. They are defined based on a confusion matrix as shown in Table 2. Because we are primarily interested in identifying OMPs, we refer to the OMPs as "positive" samples and all non-OMPs as "negative" samples here.

Table 2. Confusion Matrix in Classification

	Actual OMP	Actual non-OMP
Classified as OMP	TP (true positive)	FP (false positive)
Classified as non-OMP	FN (false negative)	TN (true negative)

Overall Accuracy: $Acc = (TP+TN) / (TP+FP+FN+TN)$ (1)

Precision of OM class: $P = TP / (TP+FP)$ (2)

Recall of OM class: $R = TP / (TP+FN)$ (3)

In our research, the performance of our predictive methods is not measured by the overall accuracy, because the majority of proteins belong to the NOM class and the overall accuracy would be influenced mainly by NOM protein prediction. Instead, since the identification of OMPs is our primary concern, our goal is high precision in OMP identification. Because of the inherent tradeoff between precision and recall in classification, our goal is to maximize the precision for OM class predictions while maintaining the recall of the same class at a reasonable level (at least 50%). Because the classification performance of the NOM class is not our main concern, all classification results are evaluated based on precision and recall of the OM class. Note that we do not use some single comprehensive measure such as F-measure (defined as $2RP/(R+P)$) for our evaluation, as we prefer high precision and it is impossible to explicitly quantify the tradeoff between precision and recall in this particular context.

4. ASSOCIATION RULE BASED CLASSIFICATION

4.1 Rule-Based Classification

In the context of protein sequence mining, frequent sequential patterns can be considered as an analogue to frequent itemsets in traditional transactional data. A pattern is frequent if it matches at least a fraction of sequences (specified by *minimum support*, or *MinSup*) in the OM class. If a pattern *P* appears frequently in OMPs, it is called an *association rule*, which in turn will serve as a classification rule $P=>OM$. This rule implies that any protein sequence matching *P* belongs to the OM class. The *confidence* of this rule is the conditional probability that a sequence belongs to the OM class given that it contains pattern *P*. Our classifier is thus built upon these association rules.

Note that since we are interested in identifying OMPs, only rules that are mined from the OM-class sequences are used. Thus we divided the original training dataset into two subsets, with each subset containing only sequences of one class. Rules are then mined only from the OM-class subset.

A *frequent pattern* has the form *X*X*…, in which each 'X' is a *frequent subsequence* made of consecutive amino acids, and each '*' is a VLDC (variable-length-don't-care) which may substitute for one or more letters when matching the pattern against a protein sequence. The reason we choose this form of pattern is that subsequences capture the local similarity that may relate to important structures or functions of OMPs, and VLDCs compress the remaining irrelevant portions. To remove trivial local similarities, we restrict frequent subsequences to those of some minimum length. Indeed, with the alphabet of only 20 amino acids, it is likely that very short subsequences will occur in sequences of both classes and such subsequences are non-discriminative with regard to classification.

Our algorithm comprises three stages:

1. Find frequent subsequences above some minimum length (specified by *MinSup* and *MinLgh*; both are determined empirically).
2. Find frequent patterns, using frequent subsequences and VLDCs, that match at least *Minsup* fraction of OM class sequences.
3. Build the classifier using frequent patterns.

4.1.1 Stage 1

To find frequent subsequences, we made use of an efficient implementation of generalized suffix tree (GST) [23] with some simple modifications. Suffix trees have been extensively used in string matching and are shown to be an effective data structure for finding common subsequences that runs in linear time [7, 10]. Since each protein sequence is essentially a string of letters, generalized suffix trees can be easily applied to mine frequent subsequences among protein sequences. Interested readers are referred to the above-mentioned documents for details.

4.1.2 Stage 2

Starting from frequent subsequences, we build frequent patterns by looking at the support of each candidate pattern constructed by concatenating two or more frequent subsequences with VLDCs. To deal with the explosive number of candidates based on the number of subsequences, several optimization heuristics and techniques have been used in order to speed up this process.

Our goal is to predict OM sequences with high precision, i.e., whenever we classify a sequence as OM, more than 90% confidence in the prediction is required. Therefore the patterns we need to extract to use as classification rules should be highly confident. Thus we set the minimum confidence (*MinConf*) to a fairly high level (greater than 85%). Intuitively, this means that frequent patterns found in OM sequences should appear very infrequently in NOM sequences. Hence we have another

constraint called *MaxSup*, which is the maximum allowed support level of a candidate pattern in the NOM training sequences. Whenever a candidate pattern does not satisfy these constraints, it is pruned immediately, i.e., it will not be used to build further patterns with more subsequences. Further experiments have shown that this has drastically reduced the running time while producing satisfactory results.

As we will show in the next stage, our final set of classification rules is built following the *MCF principle* (most-confident-first) [24], i.e., rules are ranked first according to their confidence, then according to their support level, then their general-specific relationship if applicable, and finally according to the lexicographical order. Hence, if two rules *r* and *r'* have the same confidence and rule *r* is more specific than *r'*, then rule *r* is redundant, since *r* cannot have higher support than *r'* and will be always ranked after *r'*, any protein sequence that matches rule *r* must also match *r'* and will always be classified using rule *r'*. Based on this observation, if a frequent subsequence itself has confidence of 100% - i.e., it appears only in OM sequences - any pattern that is built upon this subsequence will also have confidence 100% and thus is redundant with regard to classification. Therefore no further patterns will be constructed on top of such subsequences; instead, these subsequences will be used immediately as classification rules in the next stage.

4.1.3 Stage 3

In this stage, we must decide which ruleset to select from among all rules mined in previous steps. Using the procedures presented in [24], rules are first ranked following the *MCF principle* and a tree structure is built with each node in the tree representing one rule. The error-based pruning based on pessimistic error estimation [16] is used in order to prune overfitting rules. The general principle of pruning is that if the classifier without a certain rule has the same or lower estimated error rate than the classifier with this rule, the rule is pruned.

The remaining rules are used as classification rules. Any unseen case will be matched against these OM rules. A default rule (ϕ=>*NOM*) is added to the last to cover all cases that cannot be covered by any previous OM rules. Only when all OM rules have failed would a protein sequence be predicted as NOM by the default rule.

4.2 Refined Rule-Based Classification

Our experiments showed that the classifier built with the method described in Section 4.1 has very good performance in terms of OM class precision (well over 90%), however, the corresponding recall is low (around 40%). This is due to the high confidence of our classification rules, which have captured some features that are very specific to OM sequences; on the other hand, they cover only a portion of OM protein sequences and reject many other members of this class. As a further effort to identify more OM sequences and increase recall, we built a second level of classifier on top of the existing classifier.

At the second level, all training sequences that are covered by the default rule in the first classifier are used as a new training dataset. We apply the same pattern-mining process to this new dataset, and build a second-level classifier with newly mined rules. Now, for any unseen protein sequence, we first apply the first-level classifier and search for the presence of an OM rule. If a rule is found within the sequence, then the sequence is classified as OM; if no rule is found, the second-level classifier is used and only when both classifiers do not yield matching OM rules will a sequence be classified as NOM. Since the second-level classifier is essentially a refinement aiming to improve the classification performance of the first-level classifier, we will refer to it as *Refined Rule-Based* classification, or *RRB*. In contrast, the rule-based classification implementing only the first level is termed *Single-level Rule-Based* classification, or *SRB*.

In our experiments, this approach was shown to have effectively improved the recall level to 60%, while maintaining the precision at 90%. Theoretically, such a refinement may be repeatedly done in more than two levels. However, when classification is done across multiple levels, the remaining training data that can be used at the next level is reduced quickly. When there is no sufficient training data left, no further improvement can be done.

5. SUPPORT VECTOR MACHINE BASED CLASSIFICATION

With the increased interest in techniques of classifying biological sequences, Deshpande and Karypis [3] evaluated several widely-used sequence classification algorithms and showed that classifiers based on Support Vector Machines (SVMs) [21] are able to achieve higher accuracy than others such as Markov model based classifiers and K-nearest neighbour based classifiers. Thus we explored the use of SVMs for our outer membrane classification problem. In addition, traditional classifiers like C4.5 [16] work successfully in feature spaces with moderate dimensions. However, they are not as robust as SVMs in a high dimensional feature space. We will compare See5 [18] (an improved version of C4.5) with SVMs through experiments in Section 6.

SVMs assume all data to be represented as vectors in some feature space. Given a labelled set of training data from two classes (the positive and the negative class), SVMs find a hyperplane that correctly separates the training data of the two different classes while maximizing the distance of either class from the hyperplane (maximizing the margin). SVMs can also deal with linearly non-separable data sets by either using a kernel function *K* to map the original data vectors into a much higher dimensional space where the data points are linearly separable, or by using soft margin separation hyperplanesthat allows some degree of training error in order to obtain a large margin. A parameter *C* is introduced to control the trade-off between training error and size of the margin.

The direction of the maximal margin hyperplane is determined through a set of support vectors *SV*. A new sample $\vec{x}$ is classified depending on the sign of the following decision function:

$$f(\vec{x}) = \sum_{\vec{x}_i \in SV} \lambda_i y_i K(\vec{x}_i, \vec{x}) + b$$

where $K(\vec{x}_i, \vec{x})$ is the kernel function and y_i is the class label of the support vectors. *SV*, their weights λ_i and the parameter b are learned by the SVM.

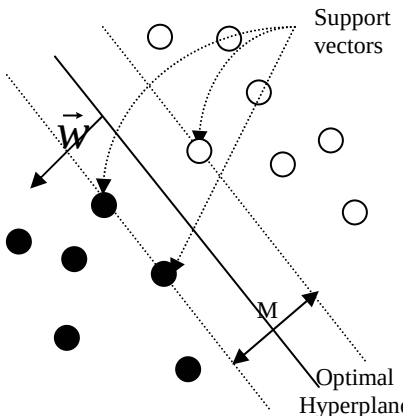

Figure 3. A linear SVM for a two-dimensional training set.

Figure 3 illustrates the basic concepts of SVMs, in particular the maximum margin separating hyperplane and the support vectors. M is the margin between the two classes. $\vec{w}$ is the norm vector of the separating hyperplane.

5.1 Feature Extraction

To apply the SVM-based method, the fist step is to transform the protein sequences, which are strings of letters, into a vector representation suitable for SVMs.

The error bounds of SVMs, E ($\vec{w}$), are defined in [26] as

$$E(\vec{w}) \approx \| \vec{w} \|_2^2 \max_i \| \vec{x}_i \|_2^2$$

where $\vec{w}$ is the norm vector of the separating hyperplane and $\vec{x}_i$ denotes the *i*th training data vector. This indicates that the error bounds are restrained by both the margin between the two classes (given by the inverse of the L2-norm of $\vec{w}$) and the maximal L2-norm of the data vectors. Obviously, if more features are used to describe protein sequences, the average distance between sequence pairs in the feature space becomes larger. As a result, this increases the chance of a larger margin. On the other hand, in general the maximal L2-norm of the data also increases with an increasing number of features. Therefore the feature space should be carefully chosen to allow an appropriate trade-off between achieving a large margin and keeping the maximal L2-norm of the data small.

We chose to use subsequences that occur frequently in OM sequences as our feature space. These frequent subsequences represent statistically significant features with regard to OM sequences, while resulting in substantially lower dimensions compared with the feature space of all potential subsequences. Unlike the rule-based method in Section 4, the SVM-based method uses only the minimum support *MinSup* to extract frequent subsequences. In other words, we do not restrict the minimum length of subsequences and the maximum support in the NOM class. The idea is to let the SVMs select important features.

5.2 Sequence Transformation

The protein sequences are then mapped into the resulting feature space. Each sequence is represented as an *n*-dimensional vector, where *n* is the number of frequent subsequences. A binary representation is used, i.e. if a frequent subsequence occurs in the protein sequence, the value of the corresponding feature is 1, otherwise it's 0.

5.3 Building SVM Classifiers

SVMs are trained in the transformed feature space with different kernel functions and different values for the trade-off parameter *C* to construct classifiers. Classical kernel functions include:

- Linear Kernel Function:

$$K(\vec{x}_i, \vec{x}) = \vec{x}_i \bullet \vec{x}$$

- Polynomial Kernel Function:

$$K(\vec{x}_i, \vec{x}) = (\vec{x}_i \bullet \vec{x} + 1)^d$$

- Radial Basic Function (RBF):

$$K(\vec{x}_i, \vec{x}) = \exp(-\gamma \| \vec{x}_i - \vec{x} \|^2)$$

We will explore the choice of different kernel functions and values of parameter C in Section 6.

6. EXPERIMENTAL EVALUATION

We evaluated the proposed data mining methods on the dataset introduced in section 3. We performed 5-fold cross validation, wherein each run takes one of the 5 folds as the test set and the remaining 4 folds as the training set. We report the average precision and recall over the 5 runs. In addition, to ensure absolute fairness, all compared methods are evaluated using exactly the same folding. In other words, the 5 folds are generated only once and are used by all methods.

6.1 SRB and RRB Classification

Since biological interest is focused on OM-class rules, we mine frequent subsequences from the subset of data that contains only OM-class proteins and use relatively high minimum confidence (95%, 90% and 85%). Maximum support in the NOM class is set at the same level of *MinSup* in OM class. Tables 3 and 4 show the precision and recall achieved by SRB at different parameter settings. Keep in mind that our goal was to maximize precision while obtaining reasonably high recall. To this end, the best result achieved by SRB is precision 97% and recall 42% (with parameters *MinSup*=0.8%, *MinConf*=95%, *MinLgh*=7), as boldfaced in Table 4. To get an idea of the size of the final classifier, the number of classification rules is also shown in the tables.

The RRB classifier is built on top of the SRB classifier that gave us the best results. Similar to the first level classifier, frequent patterns are mined from the leftover OM training data. These patterns are then used to build the next-level classifier. Here the *Minsup* level cannot be too low because our remaining training data is significantly reduced relative to the first level data. Also we decrease the *MinConf* level to 75%, since we are looking for rules that improve our recall level. In addition, the maximum support in the NOM class is set to be higher (10%) for the same reason.

Table 5 shows the classification results of the RRB classification. As expected, with the RRB classifier precision is reduced whereas recall is increased to the 60% level. However, precision can still

be maintained at the 90% level. This shows that the idea of refinement works quite well, as expected.

6.2 SVM Classification

The SVM^{light} implementation [9] was chosen because it is quite well-known and has been used extensively in previous research. Since an SVM can be trained with many different choices of input parameters, we have carried out separate experiments to investigate the impact of the *MinSup* parameter and the SVM parameters.

6.2.1 Different MinSup

To investigate the pure impact of *MinSup*, we first trained SVMs based on feature spaces that consist of all frequent subsequences mined at different *MinSup* values (from 0.8% to 15%), using the linear kernel function and setting the parameter *C* (the trade-off between training error and margin) to 1.

Table 6 reports the performance of the SVM classifier. The numbers of frequent subsequences, i.e. the numbers of features used by the SVMs, are also shown. We observe that the performance is quite stable across the whole table. Comparing the performance of SVMs trained at *MinSup* of 15% and 0.8%, we see that the number of features increases by a factor of more than 100, while the recall and precision values change only by 2% and 11%, respectively. The recall achieved is at least 79% and the lowest precision is 86%. These observations demonstrate that the influence of the feature space's dimensionality on the performance of SVMs is rather small.

On the other hand, the performance of the SVM-based classifier does vary at different *MinSup* values. With the decrease of *MinSup*, the number of frequent subsequences - i.e. the number of features - is increased dramatically, as is the margin between the

Table 3. Single-level Rule-Based Classification (I)

Min. Length	MinSup in OM class = MaxSup in Non-OM class														
	MinSup (%)	1			2			3			4			5		
	MinConf (%)	85	90	95	85	90	95	85	90	95	85	90	95	85	90	95
4	Precision (%)	56	58	59	56	58	61	57	62	73	61	74	85	72	92	99
	Recall (%)	80	78	77	79	77	74	81	74	55	72	59	41	50	36	30
	# of rules	144	138	131	142	135	122	112	93	53	64	42	19	20	13	10
5	Precision (%)	78	78	78	79	82	84	91	91	92	100	100	100	100	100	100
	Recall (%)	69	68	68	55	51	50	22	21	20	14	14	14	13	13	13
	# of rules	160	158	157	90	79	74	19	19	15	7	7	6	5	5	5

Table 4. Single-level Rule-Based Classification (II)

Min. Length	MinSup in OM class = MaxSup in Non-OM class				
	MinSup (%)	0.8	1	2	3
6	Precision (%)	94	96	96	100
	Recall (%)	50	42	22	12
	# of rules	141	86	21	4
7	Precision (%)	**97**	97	97	100
	Recall (%)	**42**	33	19	12
	# of rules	77	39	12	4

Table 5. Refined Rule-Based Classification

	SRB	RRB				
		MinSup at 2nd-level (%) (MinConf: 75%, MaxSup: 10%)				
		1.5	2	3	4	5
Precision (%)	97	**90**	91	94	97	97
Recall (%)	42	**60**	58	51	44	42
# of rules in classifier	77	159	139	98	80	77

Table 6. SVM Classification at different MinSup (linear kernel, C=1)

MinSup (%)	0.8	1	2	3	4	5	6	7	8	9	11	13	15
Precision (%)	97	**98**	95	94	92	92	91	92	91	90	88	88	86
Recall (%)	79	**81**	82	82	83	82	83	82	83	82	82	81	81
# frequent subsequences	115028	53879	14058	6611	5042	4252	3561	3111	2733	2403	1858	1458	1124

Table 7. SVM Classification using different kernels and parameter C (MinSup=5%)

	Linear Kernel		Polynomial Kernel (d=2)		Radial Basis Function Kernel (γ =0.005)	
	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Default C	74	95	71	97	72	90
C = 1	82	92	78	96	75	91
C=10	82	92	78	96	78	92
C=100	82	92	78	96	78	92
C=1000	82	92	78	96	78	92

two classes. Thus, the precision is slowly increased while the recall remains stable. But note that the best performance is not achieved at the smallest *MinSup* value. When the number of features exceeds a certain level, the performance of SVMs is degraded. This is due to the fact that from some number of dimensions on, the loss caused by an increased maximal L2-norm of data exceeds the benefit from enlarging the maximal margin.

6.2.2 Different SVM training parameters

As described in Section 5, two parameters - the kernel function *K* and the parameter *C* - are used in SVM training. In this set of experiments, we chose the feature space of all frequent subsequences with *MinSup* of 5% and trained SVMs with linear kernel, polynomial kernel of degree of 2 and RBF kernel with γ =0.005. In addition, we investigated different values of the parameter *C* (default *C*, 1, 10, 100, 1000). The default *C* set by SVM^{light} is equal to the inverse of average squared Euclidian length of training data. In our experiments, this value was much smaller than 1.

The results of these experiments are summarized in Table 7. Linear kernels in general produced higher recall values while polynomial kernels produced higher precision values. Variations of the parameter *C* have little effect on the performance of the SVMs. In particular, for values of *C* larger than 10, the performance of SVMs remains unchanged.

6.2.3 Summary of SVM classification

In general, the performance of all SVMs trained in different feature spaces and with different parameter settings is very good and rather robust. The best SVM achieves a precision of 98% and a recall of 81%, when setting *MinSup*=1%, *C*=1 and using the linear kernel function, as boldfaced in Table 6. Our results show that the best results can be obtained when choosing all frequent subsequences with a relatively small *MinSup* value, however, the performance decreases when choosing more subsequences. SVM classification seems to be very robust in terms of all other parameters.

6.3 Cross Comparison

Currently, the state-of-the-art OMP prediction algorithm is the one developed by researchers in the biology field [14], using an HMM based on highly specialized biological domain knowledge. For a more complete comparison, we have supplied our dataset to the authors of [14] to obtain classification results of their method. Note that their method used OMPs with known 3D structures to build the prediction model and, therefore, did not use our training set. Instead, the model that they have built from the 12 OMPs with atomically resolved 3D structures as described in [14] was applied directly to classify our test set. The results they reported on our dataset were a precision of 64% and a recall of 71% (averages of the 5-fold cross validation).

In addition, because decision-tree algorithms could also produce classifiers in the form of rulesets, the decision-tree algorithm See5 [18] was also selected for comparison. See5 is an improved version of the prestigious C4.5 algorithm [16] with even higher accuracy. For the purpose of this research, we obtained an evaluation license for See5. Since See5 cannot be run on sequence databases directly, we first transformed our sequence data into table-formatted data, using frequent subsequences in a fashion similar to the SVM method. The best results achieved were a precision of 95% and a recall of 40%, obtained with subsequences mined at *MinSup* 0.8%, *MinConf* 95% and *MinLgh* 6.

The comparison of all five methods is shown in Figure 4. The SVM-based method significantly outperforms all other methods. It achieved both the highest precision and the highest recall. In particular, compared to RRB - the second best performer - SVMs reduce the classification error for OM proteins from 10% to 2%, i.e., by a factor of 5, while at the same time increasing the recall by 20%. Such drastic improvement confirms the exciting

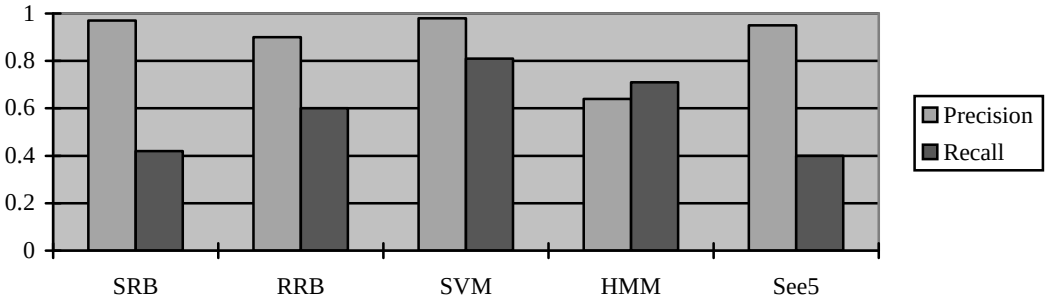

Figure 4. Comparison of Five Outer-Membrane Protein Classifiers

capability of SVMs for use in biological sequence classification. Meanwhile, the second best result was obtained by the refined rule-based classification method (RRB), which also performs much better (a classification error of 10% instead of 36%) than the current state-of-the-art method in the biological domain. This success demonstrates the strength of the frequent subsequence-based approach of protein classification. In addition, the improvement of RRB over SRB shows that the performance of rule-based classifiers can be improved by applying the framework of further refinement.

On the other hand, although SVM classification achieved the best overall performance, there are certain advantages to using rule-based classifiers. These classifiers are much easier for users to analyze and interpret, whereas no biological analysis can be easily done with the incomprehensible decision functions that are learned by SVMs.

In our application, biologists are very interested in identifying the most significant subsequences capable of discriminating OMPs from non-OMPs, as such subsequences may lead to new biological insights about this class of proteins. However, SVMs do not explicitly assign weights to all features. Instead, they perform a kernel convolution between test data and support vectors for the sake of efficient classification in high dimensional feature spaces. This holds even if a simple kernel, say the linear kernel, is used, which allows us to explicitly express the decision function as a linear combination of features. Our experimental evaluation showed that SVMs tend to use a large majority of all features, and generally the weight distribution among features is more or less uniform, i.e. with very small variance. Therefore, it is very hard to extract the most significant features from SVM classifiers.

In contrast, the classification rules used in the rule-based classifiers can be easily extracted. Initial analysis of these rules maps them to both β-strands and the short periplasmic turn regions. This may point to the importance of these regions, particularly the turns, in helping the protein to assume the correct conformation and to properly insert into the membrane. The failure of frequent subsequences to map to the extracellular loop regions supports the idea that these surface-exposed regions can be highly variable in both sequence identity and length. Disease-causing bacteria, for example, are thought to vary such surface sequences as a mechanism to evade detection by our immune system. This is one reason why β-barrel proteins are difficult to identify from sequences alone. However, as more frequent subsequences are generated through data mining experiments, mapping them to known structures will continue to provide insights into the structural biology of this class of proteins, and may assist scientists in developing 3-dimensional models for proteins which cannot be analyzed experimentally. In addition, the identification of conserved sequences that are found in the surface-exposed regions would be primary targets for new diagnostics and drugs, permitting even better prioritization of targets for further pharmaceutical study.

7. CONCLUSIONS

In this paper, we presented two new methods that make use of frequent subsequences to deal with the OMP classification problem. We created a dataset that is the largest available set of Gram-negative bacterial proteins with experimentally determined subcellular localizations. All methods were experimentally evaluated on this dataset. Both methods significantly outperformed the state-of-the-art classifier developed in the biological domain, demonstrating the remarkable strength of the frequent subsequence-based approach of protein classification. In particular, the performance achieved by our SVM-classifier is by far the best result for OMP classification that has been reported. It outperforms all other competing methods significantly and exhibits outstanding potential for biological sequence classification. In addition, the idea of refinement of rule-based classification has also been shown to perform relatively well and should be investigated in other applications. Finally, rule-based classifiers have been shown to provide biologists with useful biological patterns that improve their understanding of OMPs. Both approaches will aid research of important disease-causing bacteria, including the organisms responsible for conditions as diverse as food poisoning, water-borne diseases, ulcers, and meningitis. The combination of the classification power of the SVM approach and the biological insights gained from the rule-based approaches will allow researchers to screen genomes for novel OMPs and, potentially, to generate models of their secondary and tertiary structures.

Currently all of our classification methods only make use of the primary sequence information. We did not consider any secondary structure or additional properties of proteins at this time, for example, β-strand barrel and turn sizes, polarity of different amino-acids, etc. However, biological studies have indicated that

such characteristics are strongly correlated with the functions of proteins. We feel that future research may take these into account in order to build classifiers that integrate more key characteristics in the model. Meanwhile, we are also interested in exploring ways of extracting symbolic information from SVMs. The motivation behind this is to keep the good generalization property of SVMs and at the same time build a more understandable classifier.

8. ACKNOWLEDGEMENTS

We wish to thank Dr. P.L. Martelli (University of Bologna, Italy) for his valuable support in running his OMP prediction algorithm on our dataset and providing us with the classification results. We also thank C. Spencer (Simon Fraser University, Canada) for helpful comments and assistance in mapping rules to 3D structures.

9. REFERENCES

[1] Ali K., Manganaris S. and Srikant R., Partial classification using association rules, KDD'97, p115-118, 1997.

[2] 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., Schneider M. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, Nucleic Acids Res. 31:365-370, 2003.

[3] Deshpande M. and Karypis G., Evaluation of Techniques for Classifying Biological Sequences, PAKDD'02, 2002.

[4] Diederichs K., Freigang J., Umhau S., Zeth K. and Breed J., Prediction by a neural network of outer membrane β-strand topology, Protein Science, 7, p2413-2420, 1998.

[5] Eisenhaber F. and Bork P., Wanted: subcellular localization of proteins based on sequences, Trends in Cell Biology, 8, p169-170, 1998.

[6] Hua S. and Sun Z., Support vector machine approach for protein subcellular localization prediction, Bioinformatics, 17(8), p721-728, 2001.

[7] Hui L., Color Set Size Problem with Applications to String Matching, Combinatorial String Matching, Lecture Notes in Computer Science, 644, p230-243, Springer-Verlag, 1992.

[8] Jacoboni I., Martelli P., Fariselli P., De Pinto V. and Casadio R., Prediction of the transmembrane regions of β-barrel membrane proteins with a neural network-based predictor, Protein Science, 10, p779-787, 2001.

[9] Joachims T., Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, 2002. software downloadable at http://svmlight.joachims.org/

[10] Landau G. and Vishkin U., Fast Parallel and Serial Approximate String Matching, Journal of Algorithms, 10(2):157-169, 1989.

[11] Lesh N., Zaki M.J. and Ogihara M., Mining Features for Sequence Classification, 5th ACM SIGKDD, 1999.

[12] Leslie C., Eskin E. and Noble W., The spectrum kernel: A string kernel for SVM protein classification, Proceedings of the Pacific Symposium on Biocomputing, p564-575, 2002.

[13] Liu B., Hsu W. and Ma Y., Integrating classification and association rule mining, KDD'98, New York, NY, 1998.

[14] Martelli P., Fariselli P., Krogh A. and Casadio R., A sequence-profile-based HMM for predicting and discrimating β barrel membrane proteins, Bioinformatics, 18(1) 2002, S46-S53, 2002.

[15] Nakashima H. and Nishikawa K., Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, Journal of Molecular Biology, 238, p54-61, 1994.

[16] Quinlan J., C4.5: programs for machine learning, Morgan Kaufmann Publishers, 1993.

[17] Reinhardt A. and Hubbard T., Using neural networks for prediction of the subcellular location of proteins, Nucleic Acids Research, 26(9), p2230-2236, 1998.

[18] Rulequest Research, Information on See5/C5.0, at http://www.rulequest.com/see5-info.html

[19] Schirmer T. and Cowan S., Prediction of membrane-spanning β-strands and its application to maltoporin, Protein Science, 2, p1361-1363, 1993.

[20] Schulz G., β-barrel membrane proteins, Curr. Opin. Struct. Biology, 10, p443-447, 2000.

[21] Vapnik V., The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[22] Vert J.-P., Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings, Proceedings of the Pacific Symposium on Biocomputing, p649-660, 2002.

[23] Wang J., Chirn G., Marr T., Shapiro B., Shasha D. and Zhang K., Combinatorial Pattern Discovery for Scientific Data: Some Preliminary Results, SIGMOD-94, Minnesota, USA, 1994.

[24] Wang K., Zhou S. and He Y., Growing Decision Tree on Support-less Association Rules, KDD'00, Boston, MA, USA, 2000.

[25] Wimley W., Toward genomic identification of β-barrel membrane proteins: Composition and architecture of known structures, Protein Science, 11, p301-312, 2002.

[26] Yang M.-H., Roth D. and Ahuja N.: A Tale of Two Classifiers: SNoW vs. SVM in Visual Recognition. ECCV (4): 685-699, 2002

[27] Yuan Z., Prediction of protein subcellular locations using Markov chain models, FEBS Lett., 451, p23-26, 1999.

[28] Zhai Y. and Saier M., The β-barrel finder (BBF) program, allowing identification of outer membrane β-barrel proteins encoded within prokaryotic genomes, Protein Science, 11, p2196-2207, 2002.