CLASSIFICATION OF PROTEIN SEQUENCES BASED ON WORD SEGMENTATION METHODS

YANG YANG[1], BAO-LIANG LU[1,2*] and WEN-YUN YANG[1]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University,*
[2]*Laboratory for Computational Biology, Shanghai Center for Systems Biomedicine,*
800 Dong Chuan Road, Shanghai 200240, China
E-mail: {alayman, bllu, ywy}@sjtu.edu.cn

Protein sequences contain great potential revealing protein function, structure families and evolution information. Classifying protein sequences into different functional groups or families based on their sequence patterns has attracted lots of research efforts in the last decade. A key issue of these classification systems is how to interpret and represent protein sequences, which largely determines the performance of classifiers. Inspired by text classification and Chinese word segmentation techniques, we propose a segmentation-based feature extraction method. The extracted features include selected words, i.e., substrings of the sequences, and also motifs specified in public database. They are segmented out and their occurrence frequencies are recorded as the feature vector values. We conducted experiments on two protein data sets. One is a set of SCOP families, and the other is GPCR family. Experiments in classification of SCOP protein families show that the proposed method not only results in an extremely condensed feature set but also achieves higher accuracy than the methods based on whole k-spectrum feature space. And it also performs comparably to the most powerful classifiers for GPCR level I and level II subfamily recognition with 92.6 and 88.8% accuracy, respectively.

1. Introduction

The development of sequencing techniques led to an exponential growth of protein sequences in the public databases. In the last decade, sequence information has been successfully applied to unveil structure, function, evolutionary relationships, etc. To understand the functional roles or structure families of proteins, a lot of computational methods have been developed to classify protein sequences and detect remote homology based on their sequence similarity.

Machine learning approaches have been widely applied to this problem, such as hidden Markov model, neural networks and support vector machines (SVMs). In recent years, much work has focused on support vector machines for protein sequence classification and achieved better results. A key issue of these methods is how to interpret and represent protein sequences, i.e., features extraction. There are typically two trends. One trend inexplicitly presents features in kernels[1,2,3]. The other implements the feature extraction and classification separately[4,5].

*Corresponding author. This work was partially supported by the National Natural Science Foundation of China via the grant NSFC 60473040.

In this work, we analyze protein sequences similarly with natural languages, aiming to seek useful features to represent protein sequences for classification. The extracted features from amino acid sequences are usually amino acid frequencies, dimer or trimer frequencies, motifs, etc. The problem with these approaches is that we often do not know which features are important for determining the property of proteins relevant for our classification. To find the most discriminant features, we adopt some text processing techniques. In Ref. 6, we selected high-frequency k-mers and conducted a segmentation to calculate the feature vectors for predicting protein subcellular localization. Here we will examine several other criteria to select informative k-mers.

The proposed method arises from Chinese word segmentation and text classification techniques. Biological sequences and text documents are both strings of consecutive characters, written in different languages with respective words. The basic units of protein sequences are 20 kinds of amino acids, while for human languages, the basic units are usually letters or syllables. We model the protein sequences as concatenations of words without any space and punctuation, and develop an automatic segmentation technique for them. Obviously, there are many differences between text and protein sequences. Protein sequences have a much smaller character set than text, but are much longer than text sentences. Moreover, the words of protein sequences are unknown to us. Thus word selection and segmentation criteria peculiar to protein sequences are necessitated.

We applied the method to two protein family classification problems. The first data set is a well-studied collection of protein families built by Jaakkola *et al.*[7], and the second one consists of G-Protein-Coupled Receptors (GPCRs). Experiments show that the proposed method not only results in an extremely condensed feature set but also achieves higher accuracy (measured by ROC_{50} scores) than methods based on whole k-spectrum feature space in classification of SCOP protein families. The GPCR proteins have high diversity, whose sequences share little similarity and are particularly difficult to classify. Former researches on this subject, using decision tree, support vector machines and HMMs, have gained extremely high classification accuracy around 90%. We show that our method is comparably to the most powerful classifiers for GPCR level I and level II subfamily recognition with highly reduced feature space.

2. Method

In English text, spaces help to separate the words and understand the sentences well, while Chinese text contains no spaces, only punctuations indicating the pause or end of a sentence. The automatic analysis of Chinese text has been studied for tens of years. The first step of analysis is Chinese automatic segmentation, which is to separate the character strings into meaningful words or phrases. This is an important and basic step for Chinese information processing, such as information retrieval and handwriting recognition.

In addition, unlike syllabic languages, such as English or German, each single Chinese character can be treated as a word. Some researchers argue that the smallest unit in Chinese language is not a word but a single character. Similarly, considering each single amino acid is the smallest unit in protein sequence, we assume each amino acid as a single-character

word, and k-mers can be regarded as so called multi-word lexemes in natural language.

The proposed method consists of three major steps as shown in Fig. 1. Firstly, a dictionary is built by collecting all the 20 amino acids and certain number of meaningful k-mers according to some criterion. Secondly, a segmentation algorithm is adopted and the corresponding matching process is conducted on the dictionary. Lastly, sequences are converted into feature vectors based on the segmentation results.

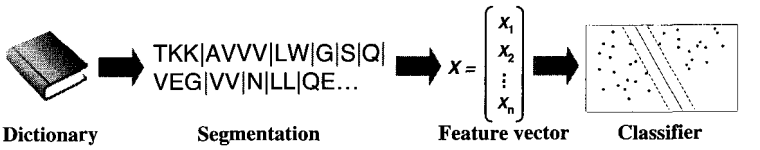

Figure 1. Flowchart of the method

2.1. *Dictionary Building*

In text, words are minimal independent and meaningful language units, and language text usually has a predefined dictionary or called lexicon, i.e., word list. However, protein sequences are written in an unknown language to us at the present state, whose words are not delineated. Any combination of letters with arbitrary length and within the given alphabet may be a word. So we first need to build a dictionary, which is the basis of segmentation.

We use statistical method to find out useful words and build dictionary based on the training data set. Firstly, a maximum word length $MaxLen$ should be set, which specifies the set of k-mers from which words are selected. Through the experiments, we find the number *four* is the best upper bound of k. Adding k-mers whose lengths are longer than four does not improve accuracy but largely increases the computational complexity. This is also mentioned in Ref. 8 that four is the typical longest distance between local interactions between amino acids.

Therefore, every k-mer with k no bigger than $MaxLen$ will be checked based on certain criterion. An intuition is that the most frequently presented strings are usually words, thus k-mers' appearance times are calculated. k-mers which are widely presented in the corpus are put into the dictionary. Considering every word will be used as a feature for classifying protein sequences, the selected words should have some discriminating ability. High-frequency words may have a balanced distribution in different classes, thus other criteria such as tf-idf and entropy value[9] could be used to choose discriminating words. These three feature ranking criteria are described in details as follows.

1) Frequency: we record the frequency for each k-mer appearing in the training sequence set and preserve a predefined proportion of the most high-frequency k-mers. The basic assumption behind this criterion is that high-frequency substrings should be segmented out and used as features, while rare strings are non-informative for classification and have little influence on global performance. Let $f_{t,s}$ be the frequency of k-mer t in

sequence s, w_t be the weight of t, and N be the size of the training set. The weight is given by the equation below:

$$w_t = \sum_{s=1}^{N} f_{t,s}. \tag{1}$$

2) tf-idf value: this criterion takes into account the distribution of each k-mer throughout all sequences in the training set. According to its definition in text categorization, tf-idf is calculated for a term in a single document. The value is in proportion to the number of occurrences of the term in the document, i.e., the tf (term frequency) part; and in inverse proportion to the number of documents in the training set for which the term occurs at least once, i.e., the idf (inverse document frequency) part. Here we refine it as the following equation. Let $w_{t,s}$ be the tf-idf value for a k-mer t in sequence s, and n_t be the number of sequences in which t appears.

$$w_{t,s} = f_{t,s} \times log\frac{N}{n_t}. \tag{2}$$

The weight of t, w_t, is defined as the maximum value of $w_{t,s}$:

$$w_t = \max_{s \in \mathcal{T}} w_{t,s}, \tag{3}$$

where $\mathcal{T}$ denotes the whole data set.

3) Entropy value: it is based on information theoretic ideas and is the most complex criterion on computation. We refine this criterion as the following equation, and assign the maximum value of $w_{t,s}$ to the weight of t as equation (3).

$$w_{t,s} = \log(f_{t,s} + 1.0) * (1 + \frac{1}{\log N} \sum_{s \in \mathcal{T}} [\frac{f_{t,s}}{n_t} \log \frac{f_{t,s}}{n_t}]), \tag{4}$$

Each of the three methods has some drawbacks. The k-mers selected by frequency are more likely words because they widely spread in the sequence data. However, some words irrelevant independent of the classification task are selected, like "the" or "and" in English text. They are non-informative but with extremely high frequencies. On the contrary, tf-idf and entropy measures tend to select particularly infrequent words. For example, suppose that a word appears many times in one or two sequences, its tf-idf value and entropy value could be very high. Such words also have little impact on classification due to their rare occurrences in total.

To avoid encountering unknown words, all 20 amino acids should be included in the dictionary. Domain knowledge can be also incorporated easily by adding signals or motifs into the dictionary. Here we consider motifs in protein sequences, which are amino-acid sequence patterns with certain biological significance, usually consisting of groups of conserved residues adjacent or near to each other. Motifs are rightly short strings with biological significance which are also available in the public database. We downloaded the motif sequence patterns from PROSITE database[10]. The PROSITE database has two kinds of

records, patterns and profiles, to describe motifs. We only make use of the former one because it has fixed sequence patterns represented by regular expressions. Such motifs allow one or several amino-acids in a position and also a fixed or a variable number of non-fixed amino acids. That is to say, a motif pattern could match multiple sequence substrings. For example, C[DN][FY] can match CDF, CNF, CDY and CNY.

2.2. *Segmentation*

Segmentation is the process of matching sequences with words in the dictionary. There are a lot of methods for Chinese automatic segmentation[11]. Maximum Matching (MM) algorithm is the most basic method, which bases on the principle of long word preference. Scanning from the head of a character string to the end, the algorithm always matches the longest word at the current position, then skips the word and continues matching for the remaining string. A variation called reverse maximum matching (RMM) scans from the end to the head of sentences. Obviously, maximum matching obtains local-optimal solution using the greedy heuristic searching. Despite the defect, MM works very well in Chinese word segmentation system given a complete dictionary, and the algorithm is simple and fast. Therefore it is widely used in Chinese information processing.

There are thousands of characters usually used in Chinese and every sentence has tens of them at most, while protein sequences usually have hundreds of letters, which are composed of only 20 amino acids. What is more, words of protein sequences are unknown, to say nothing of a complete dictionary. Thus, there may be many more ways to segment the sequences into words. To find the best way of segmentation, we first eliminate a large portion of ways by numbers of segments generated, thus only those which have the least segments remain. That is to say, long words are preferred to be matched. This is based on the consideration that longer strings contain more sequence information and are more meaningful. In language text, short words are usually auxiliary, like "a", "in", "of" in English.

However, there might be multiple ways of segmentation satisfying the least segments requirement. We assign a weight for every word in the dictionary to measure its importance, and add a maximum weight product criterion to ensure unique best segmentation. After finishing the segmentation process, appearance time of each word in the sequence is recorded. The original sequence can be converted into a vector with the dimensionality of dictionary size.

Algorithm 1 describes the process for searching the optimal way of segmentation for a protein sequence. Given a dictionary $\mathcal{D}$ and a sequence S whose length is N, $segNo[1 \cdots N]$ and $wordLen[1 \cdots N]$ are two arrays recording the number of segments which have been identified from each amino acid to the end of S, and the length of word segmented from each position, respectively. They are initialized as zero arrays. $maxLen$ stands for the maximum length of words. Beginning from the start of the sequence, Search($\mathcal{D}$, S, 1) is conducted at first.

Algorithm 1 Search

Input: Dictionary: $\mathcal{D}$, Sequence: S, Position: p
Output: Number of segments:$segNum_i, 1 \leq i \leq N$
Length of the word segmented: $wordLen_i, 1 \leq i \leq N$ # N is the length of S

if $pos = N$ # The end of S **then**
  $segNum_p \leftarrow 1, wordLen_p \leftarrow 1$
  Return $segNum_p$
end if
if $segNum_p \neq 0$ **then**
  Return $segNum_p$ # The position has been visited
end if
Initialize **len** and **num** to zero arrays with a maximum size of $maxLen$; $count \leftarrow 0$
for $k = 1$ to $maxLen$ **do**
  if k-mer (beginning from p) $\in \mathcal{D}$ **then**
    $count \leftarrow count + 1, len_{count} \leftarrow k$
    $num_{count} \leftarrow 1$+Search$(\mathcal{D}, s, p + count)$
  end if
end for
if Multiple segmentation ways have the same least number of segments **then**
  Calculate weight product for each segmentation which has the least segments.
end if
$wordLen_p \leftarrow len_k$, $segNum_p \leftarrow num_k$, where the kth segmentation way has the maximum weight product.
Return $segNum_p$

3. Results

To demonstrate the effectiveness of our method in extracting features for protein sequence classification, we tested it on two problems. One is SCOP families classification, and the other is G-Protein Coupled Receptor (GPCR) subfamily recognition (including level I and level II). The two data sets have different sequence diversity levels.

We selected N top ranked words according to certain measure, and converted each protein to a N-dimentional feature space. In the following experiments, N equals to 320, which is the summary of 20 amino acids and 100 words of 2-mer, 3-mer and 4-mer, respectively. In the current settings, the experimental results show little improvement by increasing the number of words per length because low-ranked informative words could be selected and deteriorate the accuracy. We also investigated the impact of maximum word length on the prediction performance. And we found that four is the most suitable length for the classification task. k-mers with lengths bigger than four have too low frequencies to contribute useful information.

Here we chose LibSVM version 2.6[12] as our classifier. RBF kernel performed the best in our experiments. The experimental results reported in the following sections were all

obtained with the best kernel parameter γ and penalty parameter C from a grid search procedure. All experiments were performed on a Pentium 4 double CPU(2.8GHz) PC with 2GB RAM.

3.1. *SCOP family Classification*

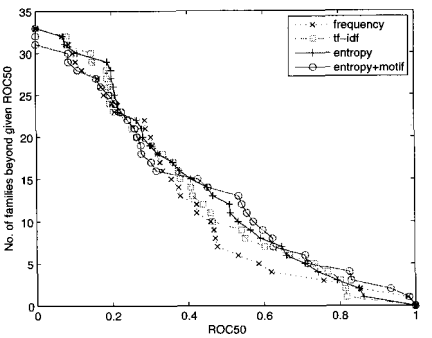

Figure 2. Results of four word selection criteria

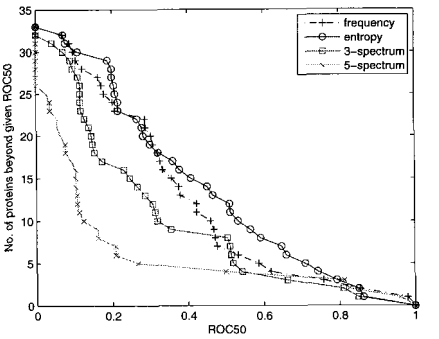

Figure 3. Comparison with spectrum-kernel

This data set consists of 33 families collected by Jaakkola *et al.* [7]. Four kinds of criteria for selecting words were compared, including a) Frequency, b) tf-idf, c) Entropy and d) Entropy plus motif. Especially, d) includes both k-mers selected by entropy value and also motifs collected from PROSITE database. Fig. 2 depicts the ROC_{50} values of 33 families in descending order of these four criteria used with segmentation.

To compare with other methods using all combinations of k-mers, we present in Fig. 3 the results of 3- and 5-spectrum kernel methods as well as our methods using frequency and entropy as the word selection criterion. A k-spectrum kernel method calculates all k-mer frequencies as features inexplicitly in the kernel function. We can see that 5-spectrum kernel does not have a satisfying result for this classification task.

In addition, we examined whether the segmentation process took effect in producing more informative features. For example, if we have a dictionary $\mathcal{D}$={A, E, P, S, SP, TPT, AAAA} and a sequence $\mathcal{S}$=TPTSPPPAAAAPAE, we can segment $\mathcal{S}$ as TPT|SP|P|P|AAAA|P|A|E. Thus the feature vector is {1,1,3,0,1,1,1}. When using the current dictionary without segmentation, i.e., the selected words are same but the feature values are calculated by counting their occurrence time overlappingly, the feature vector becomes {5,1,5,1,1,1,1}. To eliminate the influence of classifiers, we adopted linear-kernel SVMs for our methods. Fig. 4 shows results of three methods: a)3-spectrum string kernel, b) Entropy criterion with segmentation, c) Entropy criterion with overlapping counting (without segmentation).

From the experimental results, several observations can be made. Firstly, among the three measures for selecting words, frequency, tf-idf and entropy, entropy has the best ROC scores followed by tf-idf. Frequency performs slightly worse than the other two

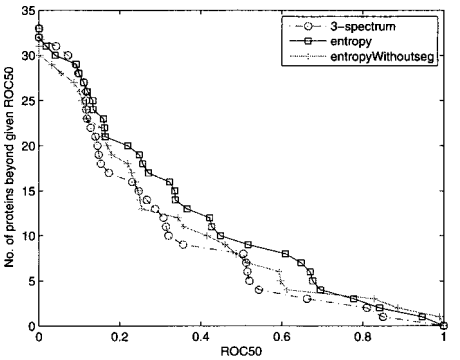

Figure 4. Comparison of methods with segmentation and without segmentation

methods, but it still achieves accuracy improvement and has minimum cost on computation. It should be noticed that adding motifs as words does not obtain an obvious better performance. Instead, it seems to widen the gap of ROC scores between the protein families which are easy to be recognized and which are not. On one hand, not all proteins have annotated motifs; on the other hand, many subfamilies share similar motifs instead of distinct motifs to discriminate them. Therefore, the motif features may even hurt the classification accuracy sometimes.

Secondly, compared with k-spectrum kernel methods which use all the k-mers, our feature extraction method gains an obvious improvement in classification accuracy, and reduces feature space dimension significantly. This is demonstrated in Fig. 3. All the four measures achieve improvement on the accuracy to a certain extent.

Thirdly, feature vectors counted through a segmentation process generally obtain better results than those counted without segmentation. Fig. 4 shows the effectiveness of segmentation. Segmentation method regards the protein sequence as a linear description of proteins, and a concatenation of words. Actually, the outcome of segmentation is to strengthen the influence of longer words in the classification by avoiding counting short words multiple times. Generally, longer words are more representative in denoting sequence features.

Finally, we could also observe from Fig. 4 that without segmentation, the feature vectors still perform better than using all k-mers, which again proves the effectiveness of our statistical measures for selecting informative words.

3.2. *GPCR protein subfamily Classification*

In Section 3.1, we present some results of protein familiy classification. Compared with family classification, the subfamilies within a certain protein family may share more similar characteristics and be more difficult to be discriminated. Therefore, to further examine the performance of the new method, we conducted another experiment to classify subfamilies of GPCRs.

GPCRs are a rich protein family of transmembrane receptors, which play a key role in a

wide variety of physiological processes. They involve in many diseases, and have particular importance in drug designs. The family is usually divided into subclasses according to transmitter types, such as muscarinic receptors, catecholamine receptors, odorant receptors, etc. Classification of GPCR proteins is a very challenging task because of the large number of family members and high diversity of sequences. The GPCR family has a hierarchical organization. There are five major classes (Classes A-E), each of which can be divided into level I subfamilies, and the subfamilies can be further divided into level II subfamilies.

The data set used in our experiment consists of Class A (receptors related to rhodopsin and the adrenergic receptor) and Class C (receptors related to the metabotropic receptors). The task is to discriminate subfamilies at level I and II within the two major classes. There were totally 1418 sequences labeled as 19 and 72 classes for level I and level II subfamily classification, respectively. A same two-fold cross-validation was conducted using the training and test data split in Ref. 13.

Table 1. Comparison of methods on GPCR classification

Classifier	No. of Feature	Feature type	Accuracy(%)	
			I	II
SVM	9n[1]	Fisher score vector (FSV)[2] space	88.4	86.3
	320	Segmentation	**92.6**	**88.8**
Decision Tree	9723	n-Gram counts	77.2	66.0
	900-2800	Binary	77.3	70.2
Naive Bayes	9702	n-Gram counts	90.0	81.9
	5500-7700	Binary	**93.0**	**92.4**
BLAST			83.3	74.5
SAM-T2K HMM			69.9	70.0

[1] n is the number of match states of the given HMM; [2] The component of FSV is the gradient of the log likelihood that a protein sequence of interest is generated by the given HMM model

We performed a multi-class classification using SVMs with one-versus-rest strategy. RBF kernel and frequency criterion were adopted. The prediction accuracies of various classification methods at level I and level II are listed in table 1, denoted by I and II, respectively. The results on the first row using SVMs and last two rows using BLAST and profile HMM were reported by Karchin *et al.*[13]. Results of Decistion Tree and Naive Bayes were reported by Cheng *et al.*[8]. With the same classifiers, SVMs, our feature extraction method obtained better results both in classification of level I and II subfamilies. It can be noticed that our method used much less features than Cheng's method, and obtained nearly equal accuracy on level I classification compared with the best result using binary features with Naive Bayes. In Ref. 8, the Naive Bayes classifier with thousands of binary features achieved the highest accuracies of 92.4% on level II subfamily classification, while our method got a relatively lower accuracy. For the extremely condensed feature extraction method, it becomes more difficult to classify the data set with such a large number of class labels when the samples that may differ only slightly. However, it is still among the best classifiers for this problem.

4. Conclusion

This study focuses on seeking efficient feature extraction of protein sequences. We aim to develop a general method for mining the information encoded in enormous protein sequences. Noticing the similarity between text and protein sequences, a method combining text categorization and segmentation techniques is proposed to separate sequences of consecutive characters to words with various lengths, and represent as feature vectors by counting frequencies of the words segmented.

To demonstrate our method, we use the feature vectors to discriminate proteins of different families and subfamilies. The extremely condensed feature set not only results in a high classification efficiency, but also achieves better result than methods based on whole k-spectrum feature space. It is shown to be very competent compared with the most successful systems for detecting protein remote homology and protein sequence classification.

As a general method for feature extraction from protein sequences, this method is not limited to solve protein family classification. It could be also applied to other classification problems based on protein sequences.

References

1. C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: A string kernel for SVM protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, 7:566–575, 2002.
2. C. Leslie, E. Eskin, J. Weston, and W.S. Noble. Mismatch string kernels for SVM protein classification. *Advances in Neural Information Processing Systems*, 15:1441–1448, 2003.
3. R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004.*, pages 152–160, 2004.
4. L. Liao and W.S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proceedings of the Sixth Annual International Conference on Research in Computational Molecular Biology*, pages 225–232, 2002.
5. K. Blekas, D.I. Fotiadis, and A. Likas. Motif-Based Protein Sequence Classification Using Neural Networks. *Journal of Computational Biology*, 12(1):64–82, 2005.
6. Y. Yang and B.L. Lu. Extracting Features from Protein Sequences Using Chinese Segmentation Techniques for Subcellular Localization. *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8, 2005.
7. T. Jaakkola, M. Diekhans, and D. Haussler. A Discriminative Framework for Detecting Remote Protein Homologies. *Journal of Computational Biology*, 7(1-2):95–114, 2000.
8. B.Y.M. Cheng, J.G. Carbonell, and J. Klein-Seetharaman. Protein Classification Based on Text Document Classification Techniques. *Proteins: Structure, Function and Bioinformatics*, 58:955–970, 2004.
9. K. Aas and L. Eikvil. Text Categorization: A Survey. *Technical Report*, 941, 1999.
10. N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P.S. Langendijk-Genevaux, M. Pagni, and C.J.A. Sigrist. The PROSITE database. *Nucleic Acids Res*, 34:D227–D230, 2006.
11. M. S. Sun and B. K. Tsou. Overview of Chinese Word Segmentation. *Modern language (in Chinese)*, 3(1):22–32, 2001.
12. C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines. *Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm*, 80:604–611, 2001.
13. R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18(1):147–159, 2002.