Transmembrane Segment Prediction from Protein Sequence Data*

Sholom M. Weiss † Dawn M. Cohen † Nitin Indurkhya ‡

† Department of Computer Science, Rutgers University
New Brunswick, New Jersey 08903, USA
‡ Telecom Australia Research Laboratories
770, Blackburn Rd, Clayton, Vic 3168, AUSTRALIA

Abstract

We consider the automated identification of *transmembrane domains* in membrane protein sequences. 324 proteins (containing 1585 segments) were examined, representing every protein in the PIR database having the transmembrane domain feature annotation. Machine learning techniques were used to evaluate the efficacy of alternative hydrophobicity measures and windowing techniques. We describe a simpler measure of hydrophobicity and a new variable window size concept. We demonstrate that these techniques are superior to some previous techniques in minimizing the segment error rate. Using these new techniques, we describe an algorithm that has a 7.9% segment error rate on the sampled proteins, while classifying 16.7% of the amino acid residues as transmembrane.

Introduction

The ability to predict aspects of a protein's structure from its sequence of amino acids is one of the main problems studied in modern molecular biology. A specific example of this is the problem of identifying substrings representing *transmembrane segments.* These are portions of certain proteins observed to be located within cellular membranes.

The insides of cellular membranes are by nature *hydrophobic*, or water repellent. Some amino acids found in proteins are also hydrophobic, while others are more naturally drawn to the watery enviroments surrounding membranes. As a result, portions of a protein spanning a membrane might be expected to contain many hydrophobic amino acids, with portions on either side of such segments being mainly non-hydrophobic.

The problem of identifying transmembrane domains is to find those subsequences of a protein which might be found inside a cell membrane. The true segments can be determined only by using expensive and time-consuming laboratory procedures. Thus, there has been considerable work to develop simple, automated methods for addressing this problem. Most of these methods rely on the observation that the problem can essentially be reduced to that of finding regions containing large proportions of hydrophobic amino acids.

Previously, several measures of amino acid hydrophobicity have been developed [Kyte and Doolittle, 1982],[Engelman *et al.*, 1986]. These scales can be employed by automated algorithms to transform the amino acid sequence into a sequence of hydrophobicity values and identify transmembrane segments within the transformed sequence. Results for various algorithms developed along these lines show that they can be quite strong when applied to specialized subclasses of known membrane proteins. For example, almost perfect results for an automated algorithm have been reported by [von Heijne, 1992] for 24 bacterial inner membrane proteins.

In this paper, we consider the larger class of all membrane proteins. We examine all proteins with the transmembrane domain feature in the *Protein Information Resource* (PIR) database. Figure 1 is a typical example of a sequence found in the PIR. For this example, there is one transmembrane segment in positions 118 to 137 within the sequence of 142 amino acid residues.

Clearly, the symbols of which the sequence is composed have biological meaning. There is ample and well-documented discussion of this topic in the molecular biology literature [Branden and Tooze, 1991],[Argos, 1989]. Nevertheless, the problem can also be characterized as a computer-based search problem. Given a string of characters in this alphabet, find the substrings that are "interesting", i.e. transmembrane. If we can describe some characteristics of interesting substrings, then the problem can be solved.

In the molecular biology literature, numerical scales have been developed to describe and measure the hydrophobicity both of amino acids and strings of these. Each type of amino acid is assigned a numerical hydrophobicity value, and strings of amino acids are as-

*DMC is supported by the National Institutes of Health under grant GM21589 and the National Science Foundation under grant NSF BIR 90 12772.

	5	10	15	20	25	30
1	PNIQN	PDPAV	YQLRD	SKSSD	KSVCL	FTDFD
31	SQTNV	SQSKD	SDVYI	TDKTV	LDMRS	MDFKS
61	NSAVA	WSNKS	DFACA	NAFNN	SIIPE	DTFFP
91	SPESS	CDVKL	VEKSF	ETDTN	LNFQN	LSVIG
121	**FRILL**	**LKVAG**	**FNLLM**	**TLRLW**	SS	

Figure 1: Sequence for PIR entry Pirl:Rwhuac. Transmembrane segment is shown in bold.

signed a hydrophobicity value that is a function of the individual hydrophobicities of their components. Transmembrane segments have a tendency towards large average hydrophobicities, despite the presence of non-hydrophobic residues within them. (Table 1 shows the hydrophobicity scales that have been used in this study. Discr: a discretization of Kyte-Doolittle, suggested in [Arikawa *et al.*, 1992]; K-D: Kyte-Doolittle [Kyte and Doolittle, 1982]; Eng: Engelman [Engelman *et al.*, 1986]; Heijne: [von Heijne, 1992])

The most common approach used in the molecular biology literature to the problem of identifying transmembrane regions, has been to select a fixed window size, e.g. 20 characters, and to sequentially segment the string looking for regions where the average hydrophobicity exceeds some threshold. Researchers have developed varying indexes [Kyte and Doolittle, 1982],[Engelman *et al.*, 1986] and classification algorithms [von Heijne, 1992], implemented in computer programs, to predict transmembrane segments for newly sequenced, but unsegmented data. Performance has been evaluated by comparing the predicted segments with the true segments of known protein sequences.

While the traditional molecular biology approach has been to program an algorithm directly based on experience and reports in the research literature, there has been at least one attempt to apply computer-intensive machine learning techniques to a large volume of segmented sequence data [Arikawa *et al.*, 1992]. In that study, all transmembrane segments in the PIR database were collected. However, samples of non-transmembrane segments were obtained by randomly selecting segments of length 30 from all proteins (membrane and non-membrane) in the PIR database. The size of the non-transmembrane set was very large – about 28 times the size of the transmembrane set. Using these samples, they attempted to learn decision trees which would separate the transmembrane from non-transmembrane segments. In that study, decision tree attributes were not based directly on hydrophobicity indexes, but rather on a restricted class of regular expressions [1] appearing in one class, but not the other. Further experiments were considered by mapping each of the 20 amino acids into one of three groups based on the hydrophobicity index, Discr, of Table 1. However, overall, their results are substantially weaker than others reported in the molecular biology literature for hydrophobicity-based classification. Moreover, while their experiments may provide some insight about characteristics of transmembrane regions, they cannot be used directly to locate such regions in new proteins.

In this paper we consider a different machine learning approach for sequence analysis. Instead of regular expressions, we return to the standard hydrophobicity analysis. We apply classifier learning techniques. This requires that we find methods for transforming variable length segment training examples into fixed length feature vectors. In this way, we can identify rules for classifying whether a given sequence is or is not transmembrane. We examine the efficacy of alternative hydrophobicity measures and windowing techniques for learning useful rules. We develop a new, simpler measure of the hydrophobicity of a sequence and a new variable window size concept. We demonstrate that these techniques are superior to previous techniques, in minimizing the segment error rate. Finally, we show how the results of simulation experiments can be used to formulate a new segmentation algorithm that performs well on the membrane proteins in the large PIR dataset. (The final step is necessary, because a good solution to the problem of classifying segments does not immediately provide a method for segmenting an unlabeled new sequence. It is useful, however, for providing a relatively unbiased framework for comparing alternative hydrophobicity measures, using computer-based train and test evaluations.)

Methods

The central goal of this work is to develop an algorithm that can segment unlabeled new sequences of amino acids of membrane proteins. The input is a string of characters and the output is a list of the positions of the transmembrane segments. We will describe several experimental procedures that are compatible with this goal. Machine learning techniques are used to evaluate and compare performance of alternative measurements and techniques for determining if segments are transmembrane.

Computer-based techniques that learn from data have several key advantages over strictly human-derived solutions:

- The computer can search over a large space of pos-

[1] Patterns that contained *don't care* subpatterns were considered.

Code	Discr Scale	K-D Scale	Eng. Scale	Heijne Scale
I	high	4.5	3.1	0.971
V	high	4.2	2.6	0.721
L	high	3.8	2.8	0.623
F	high	2.8	3.7	0.427
C	high	2.5	2.0	1.806
M	high	1.9	3.4	0.136
A	high	1.8	1.6	0.267
G	neutral	-0.4	1.0	0.160
T	neutral	-0.7	1.2	-0.083
S	neutral	-0.8	0.6	-0.119
W	neutral	-0.9	1.9	-0.875
Y	neutral	-1.3	-0.7	-0.386
P	neutral	-1.6	-0.2	-0.451
H	low	-3.2	-3.0	-2.189
Q	low	-3.5	-4.1	-1.814
N	low	-3.5	-4.8	-1.988
E	low	-3.5	-8.2	-2.442
D	low	-3.5	-9.2	-2.303
K	low	-3.9	-8.8	-2.996
R	low	-4.5	-12.3	-2.749

Table 1: Hydrophobicity Scales Used In This Study

sibilities and find a solution that is near or even optimal.

- The computer can hide some data, train on the remainder and test on the hidden data. Thus it can simulate predictions for new sequences.

For our experiments, it is most appropriate to make use of two related classification techniques: decision trees or decision rules. A standard classification problem is formulated in terms of relating class labels to particular features or measurements. In our experiments, the features are hydrophobicity values and positional information. Rules or trees identify ranges of these variables appropriate for distinguishing the classes "transmembrane" and "non-transmembrane". Numerical variables are searched over their entire ranges in the data set to find cutoffs of minimum classification error.

Several alternative methods are available for obtaining good thresholds. Decision tree learners such as CART [Breiman *et al.*, 1984] or C4 [Quinlan, 1987] perform this search over a single variable at a time. In this study, a system capable of performing more extensive search was used, namely, the rule induction program Swap-1 [Weiss and Indurkhya, 1991].

Rule induction methods, such as Swap-1, attempt to find a compact covering rule set that completely separates the classes. The covering set is found by heuristically searching for a single best rule that covers cases for only one class. Having found a best conjunctive rule for a class C, the rule is added to the rule set, and the cases satisfying it are removed from further consideration. The process is repeated until no cases remain to be covered. Unlike decision tree induction programs and other rule induction methods, Swap-1 has an advantage in that it uses optimization techniques to revise and improve its decisions. Once a covering set is found that separates the classes, the induced set of rules is further refined by either pruning or statistical techniques. Using train and test evaluation methods, the initial covering rule set is then scaled to back to the most statistically accurate subset of rules. For a more detailed description of the Swap-1 learning method, the reader is referred to [Weiss and Indurkhya, 1991].

Some experiments involved only a single variable, and the results for decision tree and rule induction are identical. When more than one numerical variable is necessary, Swap-1 should give a better answer because it is capable of optimizing multiple variables and thresholds. It is also more likely to find a compact solution which is amenable to algorithmic description. A comparison of Swap-1 with CART and other algorithms on several real-world applications can be found in [Weiss and Indurkhya, 1991].

Just as important as the method of learning from data is the technique for evaluating performance. We measured performance in terms of error rates estimated by 10-fold crossvalidation [Stone, 1974]. For 10-fold crossvalidation, the data are randomly partitioned in 10 groups of 10% of the cases. The computer trains in turn on 90% of the cases and tests on the remaining 10%, for each partition. The reported error rate is the average of the 10 trials. This gives a relatively unbiased estimate of future performance [Stone, 1974].

Hydrophobicity Indexes

There are two well-known hydrophobicity scales: [Kyte and Doolittle, 1982] and [Engelman *et al.*, 1986]. In addition, a hydrophobicity scale has recently been described, derived specifically for bacterial inner membranes [von Heijne, 1992]. We use these scales, which assign hydrophobicities to individual amino acids, in order to define ***hydrophobicity indexes*** for strings of amino acids. We compare the results for indexes derived from each of the scales.

In the simplest form of hydrophobicity index, i.e. equation 1, the average hydrophobicity is computed over a string, where n is the length of the string, and hp_i is the hydrophobicity of the ith residue in the string (from Table 1). Current algorithms almost exclusively search over a fixed window size n, for example a window of size 21 as in [von Heijne, 1992]. This approach can be modified slightly by assigning different weights to each residue in a window, as a function of position within the window (see equation 2). This method has been used, giving the central region full weight, but lower weights for the outer regions.

$$hp_{ave} = \frac{\sum_{i=1}^{n} hp_i}{n} \qquad (1)$$

$$hp_{wave} = \frac{\sum_{i=1}^{n} w_i * hp_i}{n} \qquad (2)$$

With several different indexes of amino acid hydrophobicity that vary substantially, one might speculate that interpretation of the indexes over strings is rather subjective. Thus, we proposed a new, simpler measure that is described in equation 3. As in [Arikawa *et al.*, 1992], we map the 20 character alphabet into 3 groups: high, neutral, or low. These are listed in Table 1. Given a string coded in this 3-character alphabet, we define a new measure $\mathrm{hp}_{\mathrm{dif}}$, which is the difference between the fraction of high values and low values in the string. In equation 3, N_{h} is the number of high characters and N_{l} is the number of low characters.

$$hp_{dif} = \frac{N_h - N_l}{n} \qquad (3)$$

The goal of this calculation is the same as the other hydrophobicity indexes: to search for regions where high values are found in greater abundance than low values. $\mathrm{hp}_{\mathrm{dif}}$ can also be interpreted as a weighted average hydrophobicity of the segment by assuming a 1/0/-1 weighting for the Discr Scale of Table 1. Unlike the weighting of Equation 2 though, $\mathrm{hp}_{\mathrm{dif}}$ does not assign weights according to position within the segment but simply depends on the number of high and low characters in the segment. Beyond simplicity, the computation of $\mathrm{hp}_{\mathrm{dif}}$ has another interesting property. When searching over a space of variable length n, its maximum value occurs when the first and last characters of the substring are high.

hp Index	Error rate %
Kyte	2.5
Engelman	2.7
Heijne	2.8
Dif	2.5

Table 2: Results of Experiment I

Experiments and Results

We performed several computationally intensive experiments to determine which measures are best suited for transmembrane domain identification, using hydrophobicity information. For these experiments, we included all proteins in the PIR database having at least one *transmembrane domain* feature annotation. We found 324 such proteins, containing 635 (40%) transmembrane segments and 950 (60%) non-transmembrane (some of the proteins had more than one transmembrane segment).

Experiment I: Hydrophobicity of Segmented Sequences

In this experiment, we compare samples of transmembrane segments and non-transmembrane segments to test whether hydrophobicity indexes can discriminate these two classes. We represent each segment by its hydrophobicity index value, and input them to Swap-1. We compare the accuracy of the rules learned using each of the three indexes derived from the literature, as well as the *Dif* index defined in the previous section. Predictive performance is measured by 10-fold crossvalidation.

The results of experiment I are listed in Table 2. All error rates are measured by crossvalidation. In this experiment we found that Dif did as well as the index based on the Kyte-Doolittle scale of hydrophobicity. While the index based on Heijne's scale was reported to give almost perfect results for bacterial inner membrane proteins, it is slightly weaker on the larger class of all membrane proteins.

By considering the classification of segmented sequences, this experiment does not exactly correspond to the real-world problem where the segment boundaries are not known for a new sequence, but nevertheless the results of this experiment are useful because they set an upper bound on potential performance.

Experiment II: Maximum Hydrophobicity of Variable Length Segmented Sequences

The strong results of experiment I indicate that it may be possible to distinguish known segments by their hydrophobicities. However, there may be subsegments of the known segments of each class which have hydrophobicities more characteristic of the other class. (The limiting case, of course, is a single hydrophobic amino acid in a non-transmembrane segment, which considered by itself would appear to be transmembrane.) The real

Max HP Index Method	Error rate %
fixed window-Heijne	15.6
fixed window-Engelman	13.0
variable window-Heijne	15.2
variable window-Kyte	11.7
variable window-Engelman	10.9
variable window-Dif	10.8

Table 3: Results of Experiment II

hp greater than	Predictive value %	Cases covered
2.5	93.5	31
2.0	89.4	302
1.5	79.0	587
1.0	69.5	809
.5	62.3	965
.0	56.7	1079

Table 4: Results of Experiment III: Beginning at Position 1

problem is to find the correct segmentation. Thus, we must consider some measure that summarizes a search over the complete space of possible segmentations. The maximum hydrophobicity found for *any* substring of a labeled segment can be used to characterize that segment. If the hydrophobicity for substrings of transmembrane segments were always greater than those for non-transmembrane segments, we would have the answer: look for segments that exceed a certain threshold and label these as transmembrane.

The standard technique for generating and evaluating substrings is to use a fixed window size with tapered edges. In our fixed window-size experiments, we used the window specifications of [von Heijne, 1992] which have also been used by other researchers. Independently of the fixed window methods, we introduce a variable window-size concept. Within a known segment, we search every substring of length between 10 and 70 and record the maximum hydrophobicity index as a feature of the string.

Positional information is also needed. "Signal segments", usually found at the beginning of the string, mimic the hydrophobicity of transmembrane segments, but should be classified as non-transmembrane. As the rule induction program soon discovers, the position of the maximum hydrophobicity subsegment is quite useful in differentiating the signal and true transmembrane segments.

We use Swap-1 with the known segments, represented by these two features, namely the maximum hydrophobicity within a segment and the starting position of the maximum substring. In this way, we attempt to distinguish transmembrane from non-transmembrane segments, using data that would be more readily available for new proteins. In addition to evaluating the performance of different hydrophobicity indexes, we compare variable and fixed window techniques.

The results of experiment II are listed in Table 3. They clearly demonstrate the superiority of variable windows over fixed windows. The results of these experiments move us much closer to the real-world problem.

Experiment III: Hydrophobicity Predictive Value

For the set of 24 bacterial proteins considered in [von Heijne, 1992], it was possible to a find a sequence hydrophobicity threshold above which all segments were unambiguously transmembrane. If such a method held for all membrane proteins, then one could use such thresholds to separate out the segments that could be unequivocally labeled as transmembrane, leaving the labeling of those with lower hydrophobicity values for a more complicated analysis. Experiment III is designed to test whether such thresholds can be obtained for our larger set of proteins. The effectiveness of possible thresholds as candidates for such filtering can be assessed by computing the ***positive predictive value*** which measures the percentage of transmembrane segments in the set of segments that exceed the threshold. A value of 100% predictive value for a threshold would indicate that all segments above the threshold are transmembrane. If it also covered a relatively large number of all segments, it would allow us to generalize the hypothesis in [von Heijne, 1992] to all membrane proteins.

We used the maximum hydrophobicity within a segment to characterize it and considered the predictive value of using the hydrophobicity index at different thresholds. Because of the significance of signal segments, the experiments were also performed with a starting position that would tend to exclude such segments.

The results of experiment III are listed in Tables 4 and 5. Here we take the full dataset and consider the predictive value, i.e. the percentage of correct predictions, and the number of cases that are covered when the fixed window index exceeds the specified thresholds. The results would appear to indicate that for our larger group of segments, thresholding on the hydrophobicity value of a segment alone cannot be used to identify transmembrane segments. This is in contrast with the results of [von Heijne, 1992] for the smaller set of 24 bacterial proteins which suggested that segments with hydrophobicity above some threshold could be definitely labeled as transmembrane, leaving those at lower values to a more complicated analysis.

Segmentation of Membrane Proteins

After reviewing these results, we specified an algorithm to segment unlabeled sequences. This algorithm is presented in Figure 2. The algorithm uses a variable length window and our hp$_{\text{dif}}$ index.

hp greater than	Predictive value %	Cases covered
2.5	93.5	31
2.0	93.9	279
1.5	88.6	491
1.0	81.1	651
.5	71.5	793
.0	64.3	899

Table 5: Results of Experiment III: Beginning at Position 25

Starting with position 1, The algorithm searches for a substring of length 10 to 70 that exceeds a ratio of .71[2]. It examines the longer strings first, and if necessary examines every possible substring. If the ratio is exceeded, the substring is labeled as transmembrane, and the algorithm continues at the position following the end of the substring.

Strings that exceed the .71 threshold are typically short, on the order of length 10 to 15 characters. Experiment I showed that the correctly labeled segments typically have an hp_{dif} greater than .31. This naturally leads to an expansion routine that attempts to expand the string with a ratio greater than .71 to a longer string with a lower ratio. In the algorithm, an expansion threshold of .35 was used[3]. The expansion routine is described in Figure 4. It can be efficiently coded using dynamic programming concepts.

The expansion process is equivalent to measuring another feature of the variable length segment under consideration. One might wonder how this feature would perform in some of the earlier experiments. We repeated Experiment II for a variable window size and hp_{dif}. We added a new feature: the length of the maximum hydrophobicity string when expanded to a threshold of .4. The measured error rate was 9.97%, surpassing the results for the alternative features.

The segmentation algorithm has two parameters that can be adjusted: the substring threshold and the expansion thresholds. Good values for these were obtained by an examination of the rules learned by Swap-1 in the earlier experiments. We tried several different values around the Swap-1 suggested values and obtained best results by setting these two thresholds to .71 and .35 respectively.

Segmentation Algorithm: Evaluation

Scoring the performance of an algorithm that segments an unlabeled sequence is not identical to the evaluation of the highly controlled laboratory experiments described earlier that deal with classification of pre-segmented sequences. In our segmentation algorithm, the search does not take place solely within the boundaries of a known labeled segment. Rather, the objective is to identify and label the segments themselves. In order to assess the quality of such a procedure several metrics are necessary to reflect the different kinds of errors that might arise:

- **False Positive Segments:** In the sample protein sequence in Figure 1, a negative segment (that is, non-transmembrane segment) is known to extend from position 1 to 117. If the algorithm predicts a transmembrane segment from position 15 to 45, then we record a *False Positive Error* for the negative segment. It is important to note that in order to record a false positive error, the positive segment identified *must lie completely within the boundaries* of a known negative segment (as in the example above) or extend beyond the boundaries of the known negative segment (for example, a positive segment 1-121, would result in a false positive error). Scoring of partial matches is discussed later.
- **Multiple False Positives:** In the same example above, suppose the algorithm predicts another transmembrane segment from position 65 to 95. Thus, within the negative segment 1-117, our algorithm has predicted two separate transmembrane segments. While the false positive errors reflect the number of negative segments incorrectly classified as positive segments, we also record as *Multiple False Positives* the number of incorrect positive segments generated by the algorithm. Thus, for this example, we would record one false positive segment and two multiple false positives.
- **False Negative Segments:** In the Figure 1 example, there is a transmembrane segment that extends from position 118 to 137. If the algorithm does not identify *any portion* of this segment as a transmembrane segment, then we record this as a *False Negative Error.*
- **Scoring Partial Matches:** In reality, we will often have to contend with partial matches. For example, for the Figure 1 sequence, suppose the algorithm identifies a transmembrane segment from position 110 to 130. This partially matches the known transmembrane segment 118-137. While several methods for scoring partial matches can be used, we experimented with two possible schemes for scoring partial matches:

1. **PM Scheme:** In the *PM Scheme* we did not record any error as long as some portion of the identified positive segment matched some portion of a known positive segment. By this scheme, all partial matches would be scored as correct. One criticism of this is that it might be too lenient as a scoring scheme. As an extreme example, consider a protein sequence of length 300 where a positive segment is known to be from position 100 to 150,

[2] This value was determined from previous experiments with Swap-1.

[3] Experiment I indicated that a value of .31 would be helpful. We experimented with several values around this and obtained best results with .35

```
Input: PS, a protein sequence in which each amino acid is
        coded by its hydrophobic category (high, neutral, or low)
Output: TM, a set of transmembrane segments
TM := {}
m := index to first position in PS
n := min(m+69, index to last position in PS)
segl := segment(m,n)
while (length(segl) >= 10) do
        if xdif(segl) >= 0.71 then
                seg2 := expand(segl)
                if (length(seg2) >= 25) then
                        TM := TM ∪ {seg2}
                        m := index to first position in PS after seg2
                        n := min(m+69, index to last position in PS)
                        segl := segment(m,n)
                else
                        segl := segment(m,n-1)
                endif
        else
                segl := segment(m,n-1)
        endif
endwhile
output TM
```

Figure 2: Segmentation Algorithm

```
Xdif(seg)
        xp := number of high's in seg
        xn := number of low's in seg
        len := length of seg
        xdif := (xp - xn)/len
        return xdif
```

Figure 3: XDIF Hydrophobicity Function

```
Input: seg, a segment of sequence PS
Output: expseg, an expanded segment
let seg have endpoints m and n so that seg = segment(m,n)
m1 := n - 69
n1 := m + 69
maxseg := segment(m1,n1)
expseg := seg
for (each subsegment, subseg, of maxseg) do
        let subseg = segment(m2,n2)
        if length(subseg) > 70 then move to next subsegment
        if seg is not a subsegment of subseg then move to next subsegment
        if m2 < 23 then move to next subsegment
        if xdif(subseg) < 0.35 then move to next subsegment
        if length(subseg) > length(expseg) then expseg := subseg
endfor
return expseg
```

Figure 4: EXPAND Segment Function

	Matching Scheme	
Metrics	**PM**	**MPM**
Proteins	324	324
Positive Segments	635	635
Negative Segments	950	950
Total Segment Errors	125	137
False Negatives	66	78
False Positives	59	59
Multiple False Positives	115	115
Segment Error-Rate	7.9%	8.6%
True Positive Rate	7.5%	7.5%
Predicted Positive Rate	16.7%	16.7%

Table 6: Evaluation of Segmentation Algorithm

and the algorithm identifies a positive segment from position 149 to 298. By the PM Scheme, no errors would be recorded for this protein sequence. But clearly, in this case, the algorithm's prediction is not of good quality.

2. **MPM Scheme:** To correct the bias in the PM Scheme, in the *MPM Scheme* we require that at least half the predicted positive segment match a known positive segment. Scoring by the MPM Scheme could result in some partial matches being recorded as false negative errors.

A comparison of the number of false negatives under the PM Scheme and the MPM Scheme would also give an indication of the quality of partial matches. If the false negative errors did not increase substantially when the PM Scheme is replaced by the MPM Scheme, this would indicate that the partial matches are of good quality.

- **Predicted Positive Rate:** An important metric to judge the specificity of the segmentation procedure, is to measure the *Predicted Positive Rate* – the percentage of the entire protein sequence that is predicted as transmembrane. For example, for the sequence in Figure 1, suppose a positive segment is predicted from position 110 to 130, then the predicted positive rate is 14.78%. For the same sequence, the *True Positive Rate* is 14.08%. The predicted positive rate measures the specificity of the segmentation procedure. For the same number of errors, a predicted positive rate that is closer to the true positive rate is more desirable.

These metrics were used in evaluating the segmentation algorithm which was applied to the full dataset of 324 proteins representing every transmembrane protein in the PIR Dataset. The results are listed in Table 6. We report the total segment error-rate which is the percentage of all segments misclassified (false positive and false negative errors). We also report results with both the PM and MPM schemes of scoring partial matches. Under the MPM Scheme, the error-rate increases by less than 1% over the PM Scheme. This is an indication that the predicted positive segments are mostly of good quality, even under the more lenient PM Scheme. Our algorithm misclassified 7.9% of the segments, while classifying as transmembrane 16.7% of the amino acids. In actuality, 7.5% of the amino acids belong to transmembrane segments. A higher expansion threshold is effective in bringing the predicted positive rate closer to the true positive rate, but this also makes the segment error-rate much higher (at higher expansion thresholds, the number of false negatives rises sharply). In our experiments with different threshold values, we attempted to minimize the segment error-rate.

Discussion

We have performed a number of computer-intensive experiments to determine which measures and procedures are best for identifying transmembrane segments within membrane proteins using sequence data. We examined a large collection of data to obtain these results and rigorously tested the predictive capability of these approaches. Experiment I showed that correctly segmented sequences could be distinguished with about a 2-3% error rate. This could be viewed as an upper bound on the performance of classifiers based on hydrophobicity indexes. Alternatively, the error might be reduced with improved indexes or perhaps the labels assigned in the PIR by the original experimenters are incorrect.

Experiment II supports several conclusions, among them:

- The Engelman hydrophobicity index performs the best among the indexes studied here. In [von Heijne, 1992] the Engelman index was used but a modified index was also developed based on the sequence data. In our experiments, this modified index was weaker than the others, most likely because it is derived for a very specific class of data: bacterial inner membrane proteins.
- The variable window is superior to the fixed window. While one might naively guess that its computation is intractable for real-world problems, our algorithm disproves this hypothesis, taking on average only .1 seconds on a Sparcstation-2 to segment a protein sequence.

Experiment III disproves the hypothesis that there exists a hydrophobicity threshold for guaranteeing that a string is transmembrane. While it may be true for specialized groups of proteins, such as in [von Heijne, 1992] where a hydrophobicity greater than 1 was sufficient to label a segment transmembrane, this hypothesis does not seem to hold for the more general case.

Examining these results, we combined the best techniques into a unified algorithm for segmenting unlabeled membrane sequences. The results were surprisingly good. The thresholds used in the algorithm can readily be varied resulting in a tradeoff of false positive or negative errors. Having considered numerous hydrophobicity indexes, these experimental results appear to suggest that any further improvements are less likely to come from new hydrophobicity indexes than from supplementary sequence measures and descriptors.

Several underlying assumptions in this work may have introduced some bias into the transmembrane segment identification algorithms developed here. First of all, our training sample contained only proteins that had transmembrane segments. Clearly, it might be difficult to distinguish between (non-transmembrane) *hydrophobic core* regions of proteins and transmembrane regions, which both contain large proportions of hydrophobic residues. As a result, methods implemented in this study can most appropriately be applied to proteins known to contain transmembrane, where we only wish to know which subsequences are actually found in the membrane. It might be interesting in the future to study whether, with minor changes, our methods could be extended to distinguish the transmembrane regions from the more general class of core regions. Second, we considered *all* proteins from the PIR containing transmembrane segments. We did not attempt to account for homologous or related sequences, so that proteins from families of closely related sequences may have been given undue importance. Third, in this study, we relied exclusively on hydrophobicity measures for learning transmembrane segment classifiers. Clearly there are many other factors which could influence the exact placement of amino acids in or out of a membrane, and it may be useful to include some of these as additional features to improve on our results.

References

Argos, P. 1989. Predictions of protein structure from gene and amino acid sequences. In Creighton, T., editor 1989, *Protein Structure: A Practical Approach.* IRL Press at Oxford University Press.

Arikawa, S.; Kuhara, S.; Miyano, S.; Shinohara, A.; and Shinohara, T. 1992. A learning algorithm for elementary formal systems and its experiments on identification of transmembrane domains. In *Hawaii International Conference on System Sciences.* 675–684. also Japanese Knowledge Acquisition Workshop.

Branden, C. and Tooze, J. 1991. *Introduction to Protein Structure.* Garland Publishing, New York.

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Tress.* Wadsworth, Monterrey, Ca.

Engelman, D.; Steitz, T.; and Goldman, A. 1986. Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu. Rev. Biophys., Biophys. Chem* 15:321–353.

Kyte, J. and Doolittle, R. 1982. A simple method for displaying the hydropathic character of protein. *J. Mol. Biol.* 157:105–132.

Quinlan, J. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

Stone, M. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* 36:111–147.

Heijne, G.von 1992. Membrane protein structure prediction: Hydrophobicity analysis and the positive-inside rule. *J. Mol. Biol.* 225:487–494.

Weiss, S. and Indurkhya, N. 1991. Reduced complexity rule induction. In *Proceedings of IJCAI-91*, Sydney. 678–684.