

Learning from Sequences with Determining Correlation between Transmembrane and Protein Disorder

Jack Y. Yang

Harvard University, Harvard Medical School and
Massachusetts General Hospital, Department of
Radiation Oncology, Boston, Massachusetts 02114
USA (jyang@hadron.mgh.harvard.edu)

Mary Qu Yang

National Human Genome Research Institute
National Institutes of Health, U.S. Department of
Health and Human Services, 5625 Fishers Ln, 5N01
Rockville, MD 20852 (yangma@mail.NIH.GOV)

Abstract – *Determining membrane protein structures is consistently challenging today's experiments, although such proteins account for significant portions of proteins in genomes and play crucial roles ranging from signal transduction to energy metabolism. In our attempts to construct methods for automated structural and functional annotation of genes and proteins, the identification of transmembrane proteins is an important but difficult task.*

It is suggested that amino acid sequence codes for both protein structure and function. Features that are useful for the identifying transmembrane segments and intrinsic unstructured (disorder) regions are extracted by a comprehensive detailed analysis of the amino acid compositions and various biophysical and biochemical properties of different types of proteins. We discriminate transmembrane proteins from globular proteins and identify the correlation with intrinsic unstructured proteins. We developed the boosting with bagging algorithm to enhance the power of the predictor based on our newly developed variants of the self-organizing feature map algorithm. Results have been compared favorably to other traditional classifiers such as support vector machines and decision trees.

Keywords: Transmembrane, Intrinsic Unstructured Proteins, Physicochemical Features, Boosting with Bagging, Variants of Self-Organizing Feature Map.

1 Introduction

Bioinformatics is a burgeoning field that holds great promise for deepening our understanding of biochemical pathways, for understanding the genetic differences between species and how they arose, and for studying the genetic basis of various disease processes. Many of the central questions in bioinformatics relate to protein structure and function[1]. Proteins participate in virtually every biological process. In many cases, understanding the composition, 3-D structure, and chemical activity of the proteins may be the key to meeting some of the most pressing clinical and scientific challenges. However, many protein regions and some entire proteins lack specific 3-D structures, existing instead as dynamic, disordered ensembles under physiological conditions. They are variously called protein disorder [2], natively unfolded, or

Intrinsically Unstructured Proteins (IUP). Many physiologically and pharmaceutically important proteins are membrane proteins, and the majority of therapeutic drugs target membrane proteins [3]. Despite their biological and medical significance, very little is known about the function and structure of membrane proteins due to difficulties in determining membrane structure using classical experimental methods. We therefore analyze the sequences from different types of proteins such as IUP, membrane and globular proteins. If they are different from each other based on sequence information alone, then amino acid sequence must code for both protein structure and function. In order to test this hypothesis, we extract information from sequences and develop new computational intelligence algorithms such as synergic variants of self-organizing feature map and boosting with bagging by confidence information algorithm to classify proteins utilizing machine learning methodologies (TM proteins as an example). The advantage of machine learning approaches over traditional laboratory methods is not only that the former are generally faster and less expensive, but also the later are limited on large scale. The ultimate goal is to develop reliable computational solutions for structural and functional genomics – such as predicting gene function and protein structure from sequences, *in silico* screening of leading compounds and designing new drugs, and annotation of genes and proteins for Human Genome and comparative genomes. In order to deal effectively with learning from sequences, we start with studying physicochemical properties of the amino acids that make up proteins, along with the amino acid compositions of the various types of proteins. These properties have been analyzed for our feature extractions and have applied to our variants of the self-organizing feature map algorithm enhanced by boosting with bagging to classify TM segments and IUP (also called protein disorder) regions.

2 The Data

Despite the fact that there are over 31,600 protein structures in the Protein Data Bank (PDB), less than 1% of these are membrane proteins; the solved structures in PDB are mainly globular proteins because membrane proteins are difficult to crystallize and tend to denature upon removal from the membrane. We have examined all the

current 110 membrane proteins with known structure, we found that 51 of them have multiple TM segments, and these segments are always connected by loops, which we have identified them mostly IUP regions [4]. Our dataset (Table 1) consisting 51 TM proteins with multiple TM segments, are obtained from SWISSPROT and are cross checked with PDB. There are 7638 TM, 10368 non-TM, 800 IUP (disorder) and 8191 structured (non-IUP) amino acid residues.

TABLE I. INFORMATION ON ON DATA

Proteins	TM	Non-TM	IUP(disorder)	Structured
Residues	7638	10364	800	8191

3 Biophysical & Biochemical Analysis

3.1 Comparison of Amino Acid Compositions

The first step is feature extraction from sequences. We study the characteristic of amino acids. The R group which refers to as a side chain makes the differences among 20 amino acid residues. They can be classified as polar and non-polar residues. Non-polar residues are usually hydrophobic and usually form the core of most proteins, stabilized by numerous van der Vaar interactions (though, it appears that the surface of some proteins is composed of polar residues). Some residues also carry electrical charges which are refereed as charged residues. They are mainly polar residues, and are likely on the surface of proteins and thus often interact with water and other molecules.

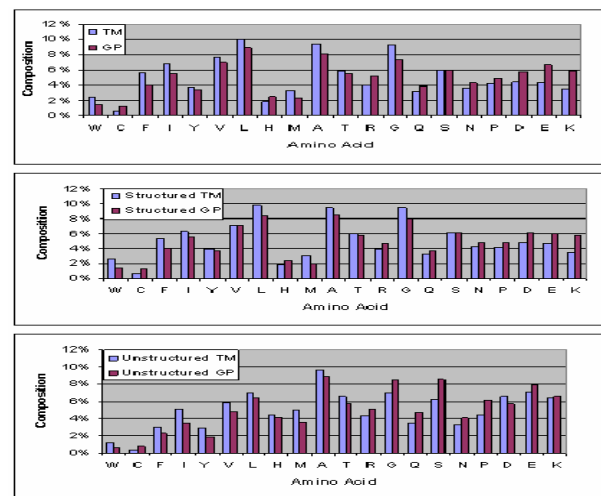


Figure 1. Amino acid composition analysis of different types of proteins in transmembrane (TM) and globular (GP) and structured TM/GP and unstructured TM/GP

It is quite evident (Figures 1-2) that the amino acid compositions and physicochemical properties of globular proteins and TM proteins are different. The resulting

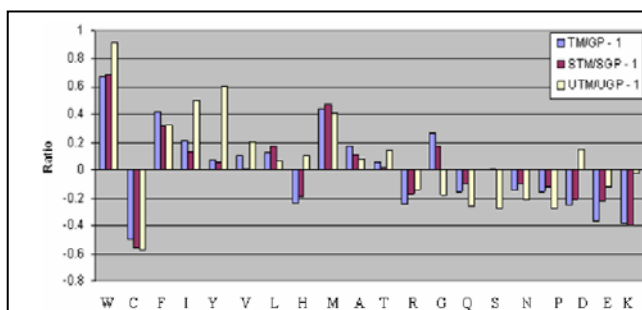
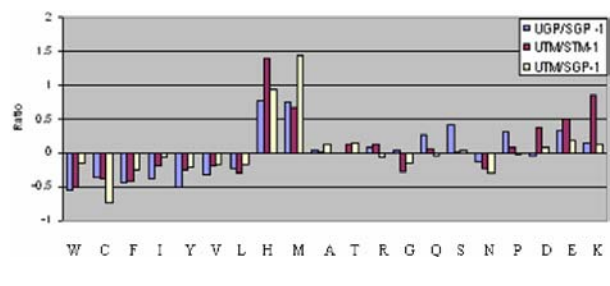


Figure 2. Comparisons of Amino acid compositions of transmembrane (TM) globular (GP), structured TM (STM), unstructured TM (UTM), structured GP (SGP) and unstructured GP (UGP) regions in protein sequences. Positive Peak represents composition enrichment and negative peak represents composition depletion.



information can be used to distinguish TM proteins from the globular proteins. TM proteins contain domains where the polypeptide chain passes through the plasma membrane of the cell. They are typically 12-35 residues long and consist of hydrophobic residues with the average polarity value greater than 1.6. We found that amino acid composition is not random, but shows a pattern, different properties of amino acids in sequence encode structural and functional information.

The sequence complexity [5] is low for IUP regions. It appears different protein folding classes may be identified by the differences in their amino acid compositions. We infer that different composition regions may correspond to new fold classes. Therefore, on the other hand, we are able to distinguish proteins by amino acid compositional differences in proteins. Figure 1 shows the amino acid compositions in TM and globular proteins along with IUP and structured proteins. Specifically, the top graph shows amino acid compositions of transmembrane proteins (TM) and globular proteins (GP). The middle graph shows compositions of structured transmembrane protein (STM) and structured globular protein (SGP), and the bottom graph shows compositions of intrinsic unstructured transmembrane proteins (UTM) and intrinsic unstructured globular proteins (UGP). Horizontal axis in Figure 1 is arranged according to solvent accessibility scale of amino acids. Moreover, Figure 2 systematically shows differences between TM proteins and globular proteins in amino acid compositions. For example, if the difference is calculated

by (transmembrane protein - globular protein) / globular protein, then the positive peaks represent composition enrichments and negative peaks represent composition depletions (of amino acids in TM proteins as compared to globular proteins). The relative differences between those types of proteins have been observed. Note that we use TM/GP-1 to denote (TM-GP)/GP. Similar denotations apply to other analysis in the graphs. Given the above analysis, we conclude: TM are very different from globular proteins. IUP are different from structured proteins. We can use amino acid compositions as features for predicting protein structure and function from sequence although using physiochemical features give better results [4].

After a thorough study of known membrane tertiary structures, we have observed that more than half of transmembrane proteins contain IUP (protein disorder). This portion is significantly more compared to other proteins, which only 35% of them contain IUP. We concluded that TM proteins are richer in IUP than the other type of proteins and TM segments and IUP regions are both statistically and biologically correlated in proteins [4]. Such discovery can be useful in future drug design, because many drugs target both membrane and IUP.

- Polarity (2 different scales)
- Polarizability
- Van Der Vaar Volume
- Bulkiness
- Flexibility
- Electronic Properties

Certain properties can be measured in different ways; this results in different scales. We determine which scale is most effective for distinguishing IUP and identifying TM. One way to estimate the effectiveness of a property in distinguishing two classes of protein is to estimate the Bayes error for each class; the Bayes error gives the smallest probability of error attainable by any classifier. The smaller the Bayes error, the more useful the property is for distinguishing the two classes.

Consider the general classification problem in a Bayesian setting for the 2-class case, in which the class label (0 or 1) is viewed as a random variable, and the goal is to assign a feature vector to one of the two classes. For the optimal threshold, the Bayes Error, is the smallest probability of error attainable by any classifier.

$$\begin{aligned} \text{Bayes Error} &= P\{\text{class } 0\} \int_{x>t} p(x|\text{class } 0) dx + P\{\text{class } 1\} \int_{x\leq t} p(x|\text{class } 1) dx \\ &= \int_{x>t} p(x, \text{class } 0) dx + \int_{x\leq t} p(x, \text{class } 1) dx \end{aligned}$$

While we provide graphs of the joint probability distributions of the property value x and the protein class that can be used to visually assess the Bayes Error, the criterion that we actually used is the Area Ratio:

$$\text{Area Ratio} = \frac{\text{overlap area of two conditional probability curves}}{\text{whole area covered by two conditional probability curves}}$$

We describe how the Area Ratio is calculated for a particular case of the property called hydrophathy[6,10]. The hydrophathy H of a given residue is calculated by centering a window (of amino acids in a sequence) of fixed length over that residue, and averaging the hydrophobicities of each amino acid contained in that window. We construct a graph as follows: We plot hydrophathy along the x -axis. We divide the hydrophathy axis (i.e. the x -axis) into bins. The y -value associated with a given hydrophathy bin is the fraction of all residues in the training set that belong to protein class 1 and that have a hydrophathy value mapped to that bin. The graph represents an approximation to the function: $P\{\text{class } 1|H\}$; we therefore call this a Bayes function. We also define a corresponding Bayes function for protein class 0 using the relation:

$$P\{\text{class } 0|H\} = 1 - P\{\text{class } 1|H\}.$$

The Area Ratio is calculated from the Bayes functions; the numerator is the area below both Bayes functions, while

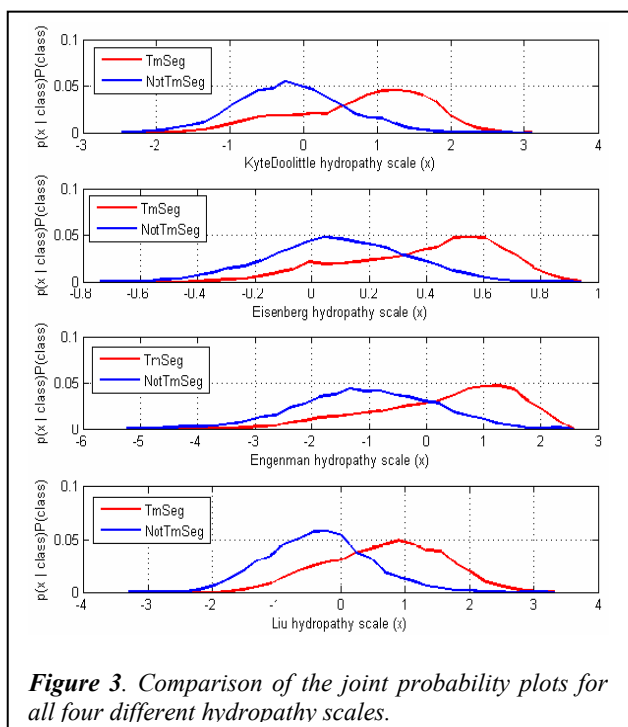


Figure 3. Comparison of the joint probability plots for all four different hydrophathy scales.

3.2 Determining Useful Features by Bayes Function

Due to the different side chains, each amino acid has different physiochemical properties. We have analyzed most, if not all of these properties in order to determine which physiochemical property are the most relevant for distinguishing IUP regions and identifying TM segments. Specifically, we have analyzed 11 features as following:

- Hydrophathy (4 different scales)

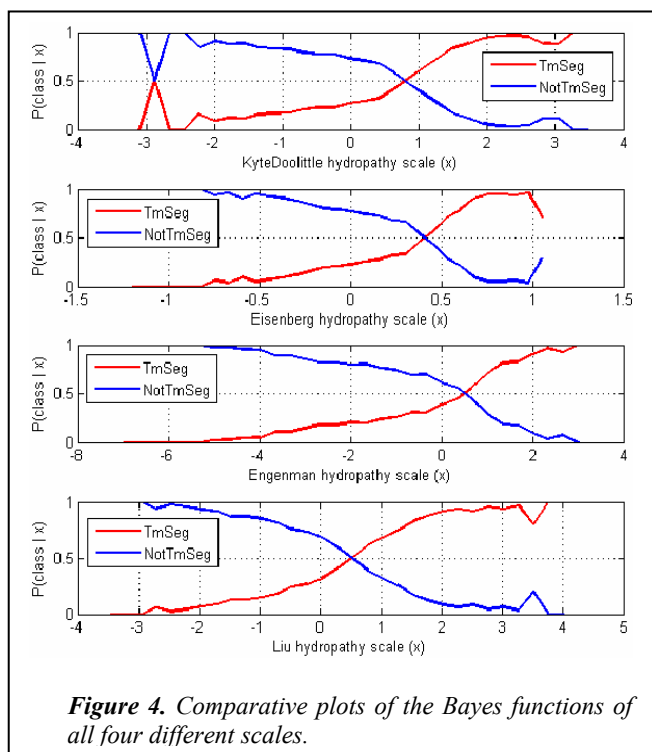


Figure 4. Comparative plots of the Bayes functions of all four different scales.

the denominator is the sum of the areas under each Bayes function. It is worth to point out that the Area Ratio is not exactly the Bayes Error, but it is closely related to it. Hence the smaller the Area Ratio is, the more useful the property being examined is (for distinguishing the two classes). Plots of the joint probability of hydrophathy (4 different scales) and the protein class label are shown in the Figure 3, while Figure 4 shows the Bayes functions. These plots reveal that TM segments (TmSeg) tend to more hydrophobic, whereas non-TM segments (NotTmSeg) tend to be more hydrophilic, which is indeed consistent with the composition analysis illustrated in Figures 1-2 (which show that larger values of hydrophathy correlate with the formation of TM segments in membrane proteins).

Specifically, Figure 3 compares the joint probability plots for all four hydrophathy scales; it appear that all four scales can be used to distinguish TM segments from non-TM segments; to decide which is the best, we plot the Bayes functions of all four scales (Figure 4), and calculate the Area Ratio for each scale. The Liu scale [6] best discriminates between TM segments and non-TM segments. The Liu scale measures the propensity of an amino acid to be in an alpha-helical state based on circular dichroism (CD). Similarly, we found that the Kyte-Doolittle [11] and Eisenberg [7] scales better discriminate between IUP and ordered regions than others. In particular, the Eisenberg scale is based on the hydrophobic moment, a measure of the amphiphilicity of a polypeptide chain. Between Grantham and Zimmerman polarity scales, the Grantham scale is better because it has a smaller average error for detecting TM. All the rest useful features are determined accordingly. Flexibility, bulkiness and

electronic properties can be used as features to discriminate between TM segments and non-TM segments in proteins as shown in Figure 5. However, the performance of classifiers constructed using bulkiness and electronic properties as features is not as good as that of classifiers constructed using hydrophathy and polarity as features. Furthermore, the performance of classifiers that use flexibility as a feature is slightly worse than that of classifiers constructed using hydrophathy and polarity as features. Features based on polarizability and van der Vaar volume cannot separate TM segments and non-TM domains well (as shown in Figure 5, red and blue curves almost completely overlap). So we exclude those two features as input for our classifier for TM. The extension to identify other useful properties is straightforward.

Using Bayes' rule, we estimate the approximate Bayes error for each feature. By calculating the Area Ratio for a number of physiochemical properties of amino acids, we rank properties according to how well they discriminate between the two protein classes under the consideration. Furthermore, using the Naive Bayes framework, we estimate how well the two classes can be separated when multiple features are considered. We investigate the likelihood function and posterior distribution of all of the 11 features extracted from the primary structure of proteins. The advantage of considering multiple features is that two classes that are not separable using any individual feature may be separable when multiple features are considered simultaneously.

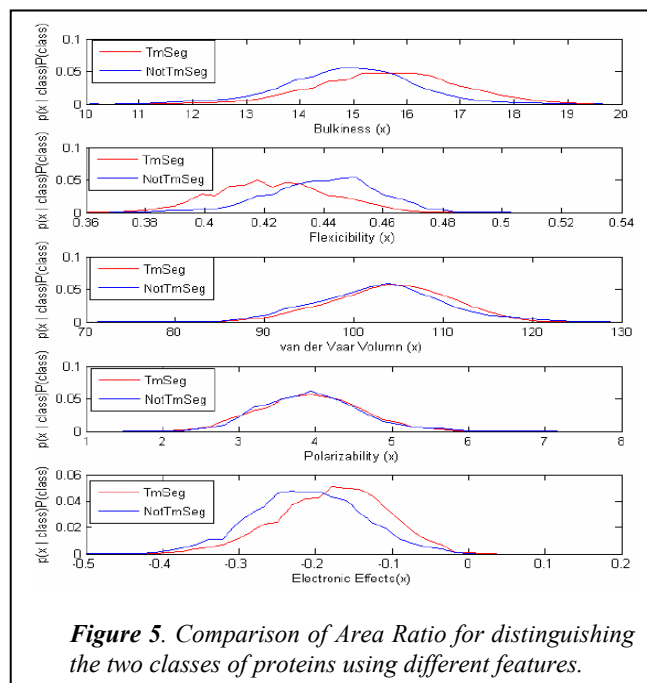


Figure 5. Comparison of Area Ratio for distinguishing the two classes of proteins using different features.

From the analysis of composition of TM and exploration of the relationship between various properties of amino acids and the formation of TM segments and IUP regions, we now can suggest that the primary structure of a protein encodes its tertiary structure and function - that is,

the sequence of amino acids determines how protein folds and what is the function of the protein. We can predict certain characteristics of 3-D structure or even function [1] of a protein by looking only at the sequence of amino acids.

4 The Algorithms

4.1 Variants of self-organized feature maps

We developed new variants of self-organizing feature map algorithm that have significantly improved the predictor's power. The algorithm was originated from self-organizing map (SOM) algorithms [8] but differs from the other algorithms by dropping the topological neighborhood and replacing it with the concept of a universal neighborhood generated by ranking, with novel variants as following:

The first variant modifies the way neurons are initialized in the feature space.

We create a new initialization procedure, each neuron has associated with it a topological neighborhood, and the neighboring neurons in the topological space tend to arrange themselves over time into a grid in feature space that mimics the neighborhood structure in the topological space.

Neural Networks including SOM update the weights after each new instance is presented to the network. Because of this, the results may be affected by the order in which instances are presented to the network. To solve this issue, let's assume that the feature space is d dimensional, so that the feature vectors X_i belong to R . For each feature k , our variants of the algorithm find the largest and smallest value of that feature over the entire training set, which are respectively L_k and U_k : where X_{ik} is the k^{th} element of the feature vector X_i . Then the initial positions of the m neurons are chosen as:

$$W_{jk}(0) = L_k + (j-1)(L_k - U_k)/(m-1)$$

Where $j = 1, 2, \dots, m$ and $k = 1, 2, \dots, d$.

Thus the m neurons are evenly distributed along the line connecting (L_1, L_2, \dots, L_d) to (U_1, U_2, \dots, U_d) . This approach has several advantages over other initialization methods:

- It guarantees that the neurons are in some sense evenly distributed throughout the feature space. Random initialization, on the other hand, does not guarantee this. If one has a large feature space, such as 80 dimensional, and comparatively few neurons, such as 30, then with random initialization those neurons are with high probability not evenly distributed throughout the feature space.
- Even a small number of neurons can be used to populate the feature space. If we consider an alternate initialization procedure in which one populates the feature space with a d -dimensional grid of neurons, and there are q grid points along each feature space axis, then the total number of neurons required to populate this grid is q^d .

Our second variant removes the dependence on the order in which instances are presented by only updating the

weights after each cycle, where a cycle involves presenting the entire training set to the network, one instance at a time. This is batch update:

- We use a "batch update" strategy for updating the positions of the neurons in feature space. This eliminates the dependence of the results on the order in which instances are presented to the network, and also stabilizes the trajectories of the neurons, which is expected to reduce the variance of the results.

- We use a fixed, but small, stepsize $b(t)$, which eliminates the problem of the weights getting stuck because the stepsize $b(t)$ went to zero too quickly. We have given a convergence proof [1] demonstrating that the fixed stepsize strategy converges to a neighborhood of the optimal solution.

1. Initialization: Choose initial positions $W_j(0)$ in feature space for the m neurons. Set $t = 0$.

2. Repeat the following until the positions of the neurons do not change (i.e. $W_j(t) - W_j(t+1) < e$, for all neurons j).

(a). Let Z_j be the "accumulator" corresponding to neuron j . Initialize Z_j to 0 for all neurons j .

(b). Present the instances (X_i, y_i) in the training set to the network, one at a time. After each instance is presented, the "accumulators" are updated as follows:

- Identifying Winning Neurons: Find the R closest neurons to the feature vector X_i , that is, find the R neurons with the smallest value of $\|X_i - W_j(t)\|$. These R neurons constitute the "neighborhood" of the input vector. Let T be the set of indices of the R winning neurons.

- Updating Accumulators: Adjust the accumulators corresponding to each of the R closest neurons using the update rule: $Z_j = Z_j + b(t)(X_i - W_j(t))$, for $j \in T$ where $b(t)$ is the learning rate.

(c). Updating Neurons: After all instances in the training set have been presented to the network, update the neurons using the update rule: $W_j(t+1) = W_j(t) + Z_j/n$ for all neurons j . where n is the number of instances in the training set.

(d). Check for Convergence: If the change in the position of each neuron is very small, then we consider the training process to have converged. The criterion for convergence is that $\|Z_j\| < e$ for each neuron j , for some small number $e > 0$. If the training process has converged, go to Step 4 below; otherwise go to Step 2(a) above.

4. Assigning Classes to Neurons: Associated with each neuron j is a count of the number of instances belonging to each class that are closer to neuron j than any other neuron. This count is calculated as follows:

- For each neuron, initialize the counts to zero.
- For each instance (X_i, y_i) in the training set, find the closest neuron to the feature vector X_i , that is, find the neuron with the index j , where $j = \arg \min \|X_i - W_j(t)\|$ and increment the count in neuron j corresponding to class y_i by 1.

• After all instances in the training set have been considered, each neuron is assigned to the class corresponding to the largest count for that neuron. Therefore the algorithm can be viewed as a stepwise procedure to minimize an objective function. The batch

update rule has advantages over the stochastic gradient update rule as following:

- Because the neuron weights are updated only after all instances in the training set have been presented to the network, the results are independent of the order in which instances are presented to the network.
- Because the batch update performs gradient descent on an “averaged” error surface, the trajectories resulting from the batch update rule will be much smoother than those resulting from the stochastic update rule; this borne out by the trajectories. Because the trajectories are smoother, it is expected that there will be less variability in the results obtained using the batch update rule than with the stochastic update rule.
- Many algorithms specify that the learning rate should be decreased during the course of training (i.e. an exponential decay rate). The problem is that if the learning rate is decreased too rapidly, then the neurons may get stuck before they have reached their optimal positions, while if it is reduced too slowly, the algorithm may become unstable and may not converge. The solution to this problem is to use a small, but fixed, learning rate. The problem with this approach is that when a fixed learning rate is used with the stochastic update rule, then the trajectories are very erratic. However, when our batch update rule is used in conjunction with a fixed learning rate, the trajectories are stable and smooth.

4.2 Boosting with Bagging

Boosting and Bagging are ensemble methods that somehow seek to combine the decisions of several classifiers (including somehow same type of classifiers using different training data) in order to improve the performance. We have experimented boosting, bagging and in particular a boosting algorithm that uses confidence information returned by the classifier [9]. To further improve the performance, we study the synergic effects of ensemble methods; we developed an algorithm that combines the bagging with boosting with confidence information. Boosting emphasizes on weaker learner for each boosting run. Assume we have N training instances, then we construct classify $f(X_i)$. Class label y_i is either 0 or 1. The square error of classify $f(X_i)$ is given by

$$error(i) = \{f(X_i) - y_i\}^2$$

The procedure of the Boosting with Bagging is described as following:

- Initialization: $a_0 = 1$; $t = 1$; $W_i = P_i = 1/N$, where $i=1, 2, \dots, N$
- for $t = 1$ to T

Take n subsamples, choose one of subsamples that gives smallest error.

$$\epsilon_t = \sum_{i=1}^N P_i (1 - h_{y_i}(\vec{x}_i))$$

Therefore,

- The Update coefficient a_t , weight W_i of training instance and probability P_i of instance at t boosting round is given by:

$$\alpha_t = -\ln\left(\frac{\epsilon_t}{1 - \epsilon_t}\right)$$

- Normalizing the weights gives a probability distribution over the training data:

$$P_i^{t+1} = \frac{W_i^{t+1}}{\sum_{i=1}^N W_i^{t+1}}$$

- The weights are updated according to

$$W_i^{t+1} = W_i^t e^{-\alpha_t h_{y_i}^t(\vec{x}_i)}$$

- The confidence of instance x belong to class k is determined by the following equation:

$$H_k(\vec{x}) = \sum_{t=1}^T \alpha_t h_k^t(\vec{x})$$

Our bagging with boosting algorithm reduces variance error but does not affect the bias error; it can be verified as following: Assume observations $X_1, X_2, X_3, \dots, X_n$

Estimator $\hat{\theta}(X_1, X_2, X_3, \dots, X_n)$ and corresponding true value $\theta(X_1, X_2, X_3, \dots, X_n)$. Thus

$$\begin{aligned} \text{Error} &= E[(\hat{\theta} - \theta)^2] \\ &= E[\{(\hat{\theta} - E[\hat{\theta}]) + (E[\hat{\theta}] - \theta)\}^2] \\ &= E[(E[\hat{\theta}] - \hat{\theta})^2 + 2(\hat{\theta} - E[\hat{\theta}])(E[\hat{\theta}] - \theta) + (E[\hat{\theta}] - \theta)^2] \\ &= E[(E[\hat{\theta}] - \hat{\theta})^2] + \underbrace{E[2(\hat{\theta} - E[\hat{\theta}])(E[\hat{\theta}] - \theta)]}_0 + \underbrace{E[(E[\hat{\theta}] - \theta)^2]}_{\text{constant}} \\ &= \text{Var}(\hat{\theta}) + \{\text{Bias}(\hat{\theta}, \theta)\}^2 \end{aligned}$$

And there are m observed estimators: $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_m$

$$\begin{aligned} \text{Var}(\bar{\theta}) &= \text{Var}\left(\frac{1}{m} \sum_{i=1}^m \hat{\theta}_i\right) \\ &= \frac{1}{m^2} \text{Var}\left(\sum_{i=1}^m \hat{\theta}_i\right) \\ &= \frac{1}{m^2} \sum_{i=1}^m \text{Var}(\hat{\theta}_i) \\ &\cong \frac{1}{m^2} m \text{Var}(\hat{\theta}_1) \\ &= \frac{1}{m} \text{Var}(\hat{\theta}_1) \end{aligned}$$

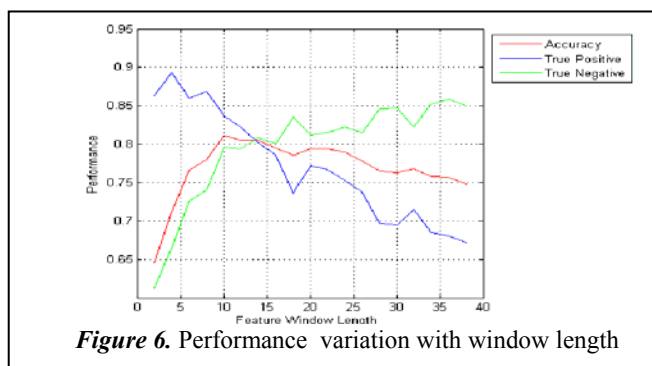
From equations above, we can see that variance error had been reduced, while bias error almost kept the same.

TABLE II. COMPARISON OF PHYSIOCHEMICAL PROPERTIES

Protein	Hydropathy	Polarity	Bulkiness	Flexibility	Electric. Eff.
TM	High	Low	High	Low	High
IUP	Low	High	Low	High	Low

TABLE III. COMPARISON WITH SVM AND DECISION TREE

Method	Our algorithm	<i>S.V.M.</i>	<i>Decision Tree</i>
Accuracy	80.08%	72.75%	74.12%
STD	1.4%	3.6%	2.7%



4.3 Joint Secondary Structure and IUP Predictor

We generate homology information augmented with IUP/order and hydrophobicity information. The homology information is generated by using the PSI-Blast program iteratively to perform multiple sequence alignment over the protein database, guided by a score matrix. The initial score matrix used in these multiple alignment calculations is BLOSUM62. The final score matrix is of dimension M by 20, where M is the length of the protein sequence, and 20 is the number of possible amino acid residues. Since we want to augment the homology information with IUP/ordered information and hydrophobicity information, we use an M by 22 matrix, where the two additional columns contain IUP/ordered and hydrophobicity information. In this approach, two stages are used in predicting secondary structure. In the first stage, the structure is predicted from the sequence information, while in the second stage, the structure information is modified to yield a physically plausible structure. As there appears to be a correlation between secondary structure and the presence of IUP regions, it is advantageous to predict these jointly.

5 Result and Discussion

We developed boosting with bagging and new variants of self-organizing feature map classifier that have been used to predict residues as either TM or non-TM, and/or IUP or structure region in proteins. In this approach, a series of classifiers are constructed based on the training data. To aid in learning the training data, a distribution over the training data is supplied to the classifier construction procedure; this distribution becomes more concentrated on the instances that are the most difficult to learn. Combining bagging with boosting with confidence information returned by the classifier has shown great promise in boosting the power of our new variants of self-organizing feature map algorithm. Under certain conditions such as boosting with bagging reduces the variance component of the error, while not affecting the bias. The length of the window over which features are extracted is a significant factor in determining the performance of the resulting classifier. Figure 6 shows the performance of the classifier as a function of the window length. Because our classifier provides the additional flexibility of trading of the true negative rate for a higher true positive rate, our results

suggested that the optimum window length for distinguishing between TM and non-TM segments is around 10. This is close to the lower limit of the length of TM segments, which tend to be between 12 and 35 residues long. Our classifier requires that two parameters be specified - the number of neurons and the neighborhood size. We obtained the best results on the dataset using 16 neurons, with a neighborhood size of 2. The algorithm identifies TM segments in proteins. Based on the analysis, the ranked best features for classifying TM are hydrophathy (Liu scale), polarity (Grantham scale) and flexibility, as this combination outperformed other combinations of features.

We benchmarked our algorithm against two other algorithms: support vector machines and decision trees, the results are given in Table 3. The resulting average accuracy of our classifier in combination with boosting with bagging reached 80%. We made several fundamental discoveries:

- Transmembrane segments and intrinsically unstructured regions tend to have opposite properties (Table 2). For example, unstructured segments tend to have a low hydrophathy value, whereas transmembrane segments tend to have a high hydrophathy value.
- Transmembrane proteins appear to be much richer in intrinsically unstructured segments than other proteins.
- Transmembrane proteins tend to be richer in buried residues and more deficient in exposed residues in comparison to non-transmembrane proteins.
- Intrinsic unstructured (also called disorder) proteins tend to be richer in buried residues and more deficient in exposed residues in comparison to ordered proteins.
- Furthermore, in the context of evolutionary pathways, we found IUP regions in TM proteins are not conserved [4].

These observations may provide insight into the structural and functional roles that IUP play in membrane proteins, and may also aid in the prediction of IUP and TM segments from primary protein structure information.

Acknowledgment

We thank Dr. Andrzej Niemierko, Dr. Laura L. Elnitski, Dr. Craig W. Codrington, Dr. A. K. Dunker and Dr. Okan K. Ersoy for many useful scientific and clinical discussions.

References:

- [1] Mary Qu Yang, O. K. Ersoy, and Jack Y. Yang "Sequential bifurcation approach to learning protein functional classes". In *Advances in Bioinformatics and its Applications, Vol. 8 of Series in Mathematical Biology & Medicine* p264–75. World Scientific, 2005.
- [2] A. K. Dunker et.al. "Protein trinity" *Nature Biotech.* 19(9): 805, 2001
- [3] C.Heusser et.al. "Therap. Potent. anti-IgE" *Curr. Opin. Immun.* 9, 1997
- [4] Mary Qu Yang "Ph.D. thesis with Interdisciplinary Bilsland Dissertaion Fellowship Award for biological physics and computer engineering dual-degree program". *Purdue University*, 2005
- [5] A.K.Dunker et.al. "Folding minimal sequences" *FEBS L.* 462(3), 1999
- [6] L.Liu&C.Deber "Guidelines membrane protein" *Biopoly.* 5(47), 1998.
- [7] D.Eisenberg et.al. "Analysis membrane hydrophobic." *JMB* 179, 1984
- [8] T. Kohonen "Self-organizing maps" *Biol. Cyberneticss*, 43:59, 1982.
- [9] C. Codrington "Boosting with Confidence Information" *ICML*, 2001
- [10] J. Kyte & R.Doolittle "Display hydrophatic character" *JMB*, 157, 1982