Journal of Computational Intelligence in Bioinformatics.
ISSN: 0973-385X Volume 1 Number 1 (2008) pp. 13–33

http://www.ripublication.com/jcib.htm

Backward Smoothing Approach to Transmembrane Protein Structure Prediction with Stochastic Dynamical Systems

[1]Takashi Kaburagi and [2]Takashi Matsumoto

[1]Department of Electrical Engineering and Bioscience,
Waseda University, 3-4-1 Ohkubo, Shinjukuku, Tokyo 169-8555, Japan
E-mail: kaburagi@matsumoto.elec.waseda.ac.jp
[2]Department of Electrical Engineering and Bioscience,
Waseda University, 3-4-1 Ohkubo, Shinjukuku, Tokyo 169-8555, Japan
E-mail: takashi@mse.waseda.ac.jp

Abstract

A backward smoothing approach was used to predict transmembrane protein structure. The proposed scheme utilizes a stochastic dynamical system with two-dimensional vector trajectories consisting of a hydropathy index and formal charge. Given a sequence of amino acids of unknown structure, each residue was predicted to be or not to be in a transmembrane region by a backward smoothing process. The proposed scheme, relative to the standard smoothing schemes applicable to the proposed model, was evaluated using publicly available benchmarking sequences. Our algorithm was found to yield reasonably good results.

Introduction

Many machine-learning algorithms have been successfully applied to the analysis of biological data. However, in many of the applications of machine learning to protein structure, prediction problem accuracy must be further improved. We used our algorithm on prediction problems for structures of one class of protein: transmembrane proteins.

Transmembrane proteins have long been considered to be critical in understanding biological functions such as cell signaling, ion transport, and intercellular communication [1][2][3]. It has been reported that approximately 45% of the drugs in use today target G protein-coupled receptors (GPCRs) [4][5]. Some 20% to 30% of genes in an average genome are estimated to encode membrane proteins [6]. Because

of their biological and pharmaceutical importance, identification of transmembrane helices in membrane proteins is a priority. Although promising methods in Xray crystallography and nuclear magnetic resonance (NMR) have begun to open avenues to the determination of these structures [7][8][9], the number of known three-dimensional structures remains small [10][11][12]. Therefore, reliable algorithms to predict transmembrane protein structures would be very useful.

There are two basic methods of looking at protein structure predictions. One basic method is to use algorithms based solely on the construction principles of proteins associated with the physicochemical properties of amino acids. No training is involved. In this method, windowed averages of physicochemical quantities are taken. There are several successful examples of algorithms of this type [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23]. The other basic method is to collect data sets of known structures, extract their features, and use machine-learning algorithms to make predictions. Some improvements have been made in using this type of algorithm, but further advances are necessary to improve the reliability of predictions [10] [24] [25] [26] [27] [28]. We used a novel machine-learning algorithm to predict protein structures and evaluated the reliability of the predictions.

A machine-learning algorithm assumes that there are models and associated parameters behind the available data sets. Generally, the degree of success of a machine-learning algorithm depends on two factors: how well the model structure characterizes the target molecule from which the data was taken, and how well the learning algorithm incorporates the available data sets. Among protein structure prediction problems, there are, in general, three important aspects in transmembrane structure prediction:

i. The data is sequential with respect to a one-dimensional space variable, and a particular amino acid is correlated with other amino acids.
ii. The data sets have uncertainties in that a particular structure may be observed to have different amino acid sequences.
iii. The number of training data sets for learning is severely limited because of the difficulties associated with using X-ray crystallography or NMR for transmembrane proteins.

This paper considers the restricted class of transmembrane protein structure prediction problems instead of general classes of problems. Specifically, we assumed that a particular amino acid sequence is from a transmembrane protein, even though predictions as to whether the sequence is water-soluble or a transmembrane protein could have been attempted instead. A primary reason for this is because there are several very good tools available for such prediction problems [29]. The goal of this paper is, given an amino acid sequence, to predict transmembrane regions, i.e., predict whether each amino acid belongs to a transmembrane region.

These problems are nontrivial because, as previously mentioned, so few transmembrane protein structures have been fully characterized. A finer model that captures the nature of a set of data would improve prediction accuracy, provided that a sufficient number of training data sets were available. Because there are so few available data sets, serious consideration is essential to make both the model structure and the associated learning algorithm as simple as possible without losing sight of the

nature of the problem. Therefore, transmembrane protein structure prediction is a significant challenge for machine-learning approaches.

This paper proposes a novel algorithm for predicting the transmembrane regions in a given test amino acid sequence. Contributions of this paper are listed below.

i. A novel prediction scheme is proposed for predicting transmembrane regions utilizing a finite-state, stochastic dynamical system (Hidden Markov Model (HMM)). The scheme is applied to our previously proposed model reported in [30] and is evaluated in comparison with the Viterbi algorithm and other standard smoothing schemes as well.
ii. HMM topology consists of open-loop connections of submodels made up of the transmembrane region and loop region submodels. A stochastic dynamical system runs concurrently with inner state dynamics, so that once the dynamical system leaves a particular state, it never returns to that state. Some of the previous HMMbased algorithms, in contrast, are designed to have five or seven states that could be revisited.
iii. The results reported in section III-.3 suggest that our proposed prediction scheme yields reasonably good results.

Stochastic Dynamical System Approaches

The application of finite-state stochastic dynamical systems (also known as HMM) is very broad and its techniques are suitable for characterizing the nature of sequential data. These techniques have been used to solve a variety of problems in, for example, speech recognition and handwriting recognition. In the HMM framework, an unobserved state sequence $\{|Q_t\}^T_{t=1}$ is assumbed to exist behind an observed sequence $\{|Q_t\}^T_{t=1}$.

Approaches for predicting transmembrane regions based on HMMs have been successfully realized in such tools as TMHMM [10][27] and HMMTOP [28]. Krogh et al. defined seven types of states in TMHMM: loop cytoplasmic, cap cytoplasmic, helix core, cap non-cytoplasmic, short loop non-cytoplasmic, long loop non-cytoplasmic, and globular domains. A probability distribution of the 20 amino acids, which was learned from the training data set, was defined in each state taking into account the grammar. Tusnady and Simon also proposed an HMM-based method in HMMTOP [28]. This model employs five states (inside loop, inside helix tail, helix, outside helix tail, and outside loop). This algorithm focuses on the differences in the amino acid distributions in the structural parts, rather than on the amino acid distribution itself.

In these approaches, as well as in our approach, the “state” of the HMM has a value indicating whether the state is in a transmembrane region or a loop region. To predict the transmembrane regions in such approaches an algorithm to determine an “optimal” state sequence $\{Q^*_t\}$ plays a critical role in the performance of the algorithm. We used a novel method to determine an optimal state sequence $\{Q^*_t\}$ for predicting transmembrane regions.

Here, two of the standard approaches for predicting an optimal state sequence $\{Q^*_t\}$ will be discussed. One of the commonly used algorithms for finding the most

likely sequence of hidden states $\{Q^*_t\}$ for an HMM is the Viterbi algorithm. The Viterbi algorithm is a dynamic programming approach to optimize the state sequence. $\{Q^*_t\}$, given the observation sequence $\{O_t\}$. The Viterbi algorithm is implemented in HMMTOP [28]:

$$\{Q_t^*\}_{t=1}^T := \underset{Q_t}{\operatorname{argmax}} P(\{O_t\}_{t=1}^T, \{Q_t\}_{t=1}^T)$$

Another common algorithm for determining an optimal state sequence $\{Q^*_t\}$ can be defined by:

$$Q_t^* := \underset{i}{\operatorname{argmax}} P(Q_t = q_i \mid O_1, \cdots, O_T, w, \mathcal{H}). \tag{1}$$

Generally, in dynamical systems, estimation of

$$P(Q_t \mid O_1, \cdots, O_{t-1}, O_t, \cdots, O_T), \quad 1 < t < T$$

is called smoothing, estimation of

$$P(Q_T \mid O_1, \cdots, O_{t-1}, O_t, \cdots, O_T)$$

is called filtering, and estimation of

$$P(Q_t \mid O_1, \cdots, O_{t-1}, O_t, \cdots, O_T), \quad 1 < t < T$$

is called prediction. The algorithm defined in (1) can be described as a smoothing method. This method is implemented in TMHMM [10].

Now, let us recall our prediction algorithm proposed in [30]. In [30], {Q*t} is defined as:

• for $t = 1$

$Q^*_1 := q_1$

• for $t > 1$

$$Q_t^* := \underset{q_i \in \{q_j, q_{j+1}\}}{\operatorname{argmax}} P(Q_t = q_i \mid O_1, \cdots, O_T, w, \mathcal{H}) \tag{2}$$

where $Q^*_{t-1} = q_j$

This method is a revised method of (1), so that the predicted $\{Q^*_t\}T_{t=1}$ incorporates topological consistency.

Here we propose a novel smoothing method to predict $\{Q^*_t\}T_{t=1}$. the proposed prediction scheme emphasizes the dependency on the previous state Q^*_{t-1} more than the previous observation data (O_1, …, O_{t-1}) once the previous state has been estimated.

The Proposed Smoothing Approach

- for $t = 1$

  $Q_1^* := q_1$

- for $t > 1$

$$Q_t^* := \underset{q_i}{\operatorname{argmax}} P(Q_t = q_i \mid \boldsymbol{O}_t, \cdots, \boldsymbol{O}_T, Q_{t-1}^*, w, \mathcal{H}) \quad (3)$$

Details of this method are presented in Sec. II-F.

The performances of the methods described above are evaluated in Section III.

Algorithm

Observation sequence $\{O_t\}$

Consider an amino acid sequence of length T. In this paper, we consider the two-dimensional vector trajectory O_t associated with amino acids, instead of the 20 letter symbol sequence:

$$\begin{aligned} \{\boldsymbol{O}_t &:= (O_t^1, O_t^2)\}_{t=1}^T \\ O_t^1 \in v_{k_1}^1 &\quad , \quad O_t^2 \in v_{k_2}^2 \\ k_1 = 1, \cdots, K_1 &\quad , \quad k_2 = 1, \cdots, K_2 \end{aligned}$$

The first component of output $O^1{}_t$ is the hydropathy index; the KD index is used in this paper1[1]. Even though the hydropathy index is real valued, there are only a finit number of index values k_1, e.g., $K_1 = 17$ for the KD index. The second component $O^2{}_t$ is the formal charge associated with an amino acid[2]. Similarly, there is only a finit number of formal charge values k_2, i.e., +1, 0, and –1 ($K_2 = 3$). A major consequence in considering these physicochemical indices instead of the 20 letter symbols is the fact that account. That is, two amino acids with a similar hydropathy index can be considered close to each other based on thisparticular metric. This allows operations to avoit overfitting problems. Although these operations are sometimes called "smoothing", to avoid confusion, the term "anti-overfitting" will be used in this paper.

Unobserved sequence $\{Q_t\}$

One way of taking into account the sequential nature of the problem, i.e., the fact that each amino acid is correlated with other amino acids, is to consider an unobserved

[1] There may be better hydropathy indices than the KD index, and as many as 80 different hydropathy indices have been proposed.

[2] Histidine can be assumed to have two possible formal charge values, depending on pH. The typical pKa value of histidine in proteins in 6.0 [31]. This indicates that most histidine residues are in a state in which the formal charge is 0. Therefore, the histidine formal charge will be assumed to be 0 in the experiment reported in section III. Since the number of histidines appears to be small in the data sets used in our experiment, our tentative assumption did not appear to have a significant effect on prediction performance.

auxiliary sequence $\{Q_t\}$ of length T and to look at O_t as an *output* with uncertainty. This sequence $\{Q_t\}T_{t=1}$ is a trajectory of a finite-state inner stochastic dynamical system indexed by a one-dimensional parameter t. Here $Q_t \in \{q_1, \ldots, q_N\}$, where q_i is the state within an inner stochastic dynamical system, and N denotes the number of states.

Joint probability distribution

The joint probability distribution of $\{O_t, Q_t\}^T_{t=1}$ is described by:

$$
\begin{aligned}
& P(\{O_t\}_{t=1}^T, \{Q_t\}_{t=1}^T \mid w, \mathcal{H}) \\
&= P(\{O_t\}_{t=1}^T \mid \{Q_t\}_{t=1}^T, w, \mathcal{H}) P(\{Q_t\}_{t=1}^T \mid w, \mathcal{H}) \\
&= \prod_{t=1}^T P(O_t^1, O_t^2 \mid Q_t, w, \mathcal{H}) \\
&\quad \times P(Q_1 \mid w, \mathcal{H}) \prod_{t=2}^T P(Q_t \mid Q_{t-1}, w, \mathcal{H}) \\
&= \prod_{t=1}^T P(O_t^1 \mid Q_t, w, \mathcal{H}) P(O_t^2 \mid Q_t, w, \mathcal{H}) \\
&\quad \times P(Q_1 \mid w, \mathcal{H}) \prod_{t=2}^T P(Q_t \mid Q_{t-1}, w, \mathcal{H})
\end{aligned}
\tag{4}
$$

where w is a parameter set and H stands for the underlying model structure. The first two equations in Eq. (4) are in the general form of a stochastic dynamical system, whereas the last equation is an HMM, which will be used in this paper.

Schemes described by Eq. (4) are sometimes successful for nonlinear time series prediction problems, in which the inner dynamical system has in infinite number of states [32], handwriting recognition problems [33][34], and online signature verification problems [35], in which the inner dynamical system has a finite number of states. In these three problem classes, index parameter t is *time*, whereas in a protein primary sequences, t stands for spatial *position* from the N-terminus. Specification Eq. (4) assumes that the inner stochastic dynamical system is the first order, and the observations mechanism is independent (with respect to probabilities involved) of the inner dynamical system, although generalizations are possible.

1). Emmission probabilities: The emission probabilities of the hydropathy index and formal charge are defined as:

$$
\begin{aligned}
& P(O_t^1 = v_{k_1}^1 \mid Q_t = q_i, w, \mathcal{H}) := b_{i,k_1}^1 \\
& P(O_t^2 = v_{k_2}^2 \mid Q_t = q_i, w, \mathcal{H}) := b_{i,k_2}^2 \\
& \quad i = 1, \cdots, N \qquad k_1 = 1, \cdots, K_1 \qquad k_2 = 1, \cdots, K_2
\end{aligned}
$$

where $b^1_{i,k1}$ and $b^2_{i,k2}$ satisfy the constraints $b^1_{i,k1} \in [0, 1]$, $b^2_{i,k2} \in [0, 1]$ and $\sum_{k_1=1}^{K_1} b_{i,k_1}^1 = 1, \sum_{k_2=1}^{K_2} b_{i,k_2}^2 = 1$.

In the formulation Eq. (4), emission probalities $\{b^1_{i,k1}\}$ and $\{b^2_{i,k2}\}$ are assumed to be independent for the sake of simplicity, whereas in reality, they are not.

2). *State transition probabilities*: State transition probability from state $Q_{t-1} = q_i$ to $Q_t = q_j$ is defined as:

$$P(Q_t = q_j \mid Q_{t-1} = q_i, w, \mathcal{H}) := a_{ij}$$
$$i, j = 1, \cdots, N$$

where aij ∈ [0, 1] and $\sum_{j=1}^{N} a_{ij} = 1$.

3). *Initial state probability*: The probability of initial state Q_1 is defined as:

$$P(Q_1 = q_i \mid w, \mathcal{H}) := \pi_i$$
$$i = 1, \cdots, N$$

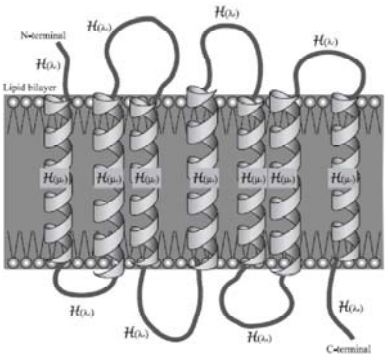
where $\pi_i \in [0, 1]$ and $\sum_{i=1}^{N} \pi_i = 1$.

Model structure

Successful applications of machine learning algorithms crucially depend on the particular model structure chosen. Model structure must be carefully designed taking into account the specific purpose(s) of the prediction problem as well as the available data sets. HMM is no exception. A researcher may wish to design a model structure that is as detailed as possible and takes into account many aspects of transmembrane proteins. Because so few transmembrane protein structures are known, it would not be feasible to tune the detailed models with many delicate parameters. This is one aspect of the data fitting versus simplicity dilemma (Occam's razor).

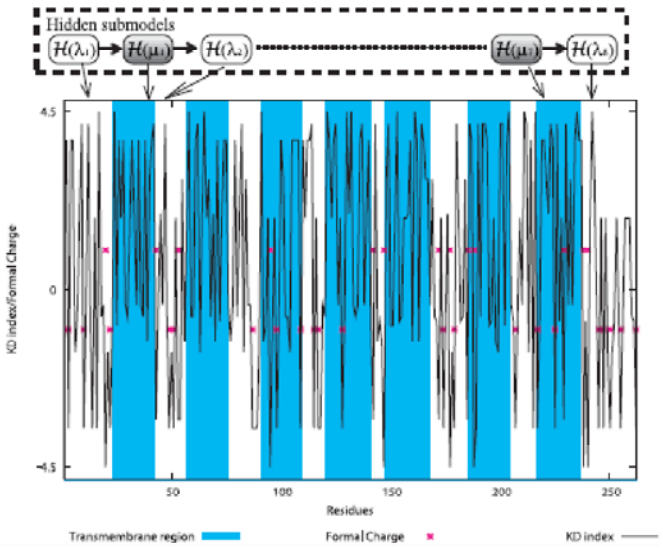
Model *H* used in this paper consists of the following.

i. The model *H* carries a fixed number of transmembrane regions *M*.
ii. The model *H* has an entirely "open-loop" structure consisting of alternating connectins of transmembrane region submodels $H(\mu_v)$, $v = 1, \ldots, M$ and loop region submodels $H(\lambda_u)$, $u = 1, \ldots, M + 1$.[3] Fig. 1 illustrates a schematic picture of the model when $M = 7$.



(a) Schematic diagram of a transmembrane protein in a lipid bilayer. The first loop region corresponds to $H(\lambda_1)$, whereas the first transmembrane region corresponds to $H(\mu_1)$, and so on.

[3] In contranst, transitions among submodels are allowed in TMHMM so the M is not fixed [10].



(b) Schematic diagram of the proposed model structure and an example of the observation trajectories when $M = 7$. The KD index plots are connected with lines in order to show changes with respect to residues. Zero charges are not shown for clarity.

Figure 1: The overall topology of the proposed model. Each submodel is connected by the left-to-right topology.

(iii) Within a submodel $H(\lambda u)$ for a loop region, there are $\tau(\lambda_u)$ states in which a simple left-to-right topology with a self-loop is built in, as shown in Fig. 2(a). $\tau(\lambda_u)$ is a parameter to be learned from the training data set.

(iv) Within a submodel $H(\mu_u)$ for a transmembrane region, there are $\tau(\mu_v)$ states in which a simple left-to-right topology with a self-loop is built in, as shown in Fig. 2(b).

(v) Each state q_i ($i = 1, \ldots, N$) is associated with a unique submodel $H(\mu_v)$ or $H(\lambda_u)$.

(a) Submodel for Loop Region (b) Submodel for Transmembrane Region

Figure 2: Details of each submodel.

Learning

Here, the method to set HMM parameter vector w using a traiing data set and hyperparameters is described.

Consider the following available training data sets:

$$\begin{aligned} D_{train} &:= \{D_h\}_{h=1}^{H} \\ &:= \{\{O_t(h), F_t(h)\}_{t=1}^{T(h)}, m(h)\}_{h=1}^{H} \end{aligned}$$

where $F_t(h) \in \{\lambda_1, \ldots\lambda_{m(h)+1}, \mu_1, \ldots, \mu_{m(h)}\}$ is an annotation sequence which denotes the region associated with observation O_t, $m(h)$ is the number of transmembrane regions, and H is the total number of avaible sequences for the training data set.

The proposed algorithm attempts to construct one model from one training data set. Thus, if a model H_h is given, only one training sequence D_h will correspond. The learned parameter vector can be denoted as $w_h := \{\{a_{ij}\}, \{b^1_{jk1}, b^2_{ik2}\}, \{\pi_i\}\}$.

Step 1: Number of transmembrane regions:

Learning M_h: For model H_h, the number of transmembrane regions M_h is set to be the same as the number of transmembrane regions of the training data $m(h)$.

$$M_h := m(h)$$

Step 2: States:

The number of states in a submodel $\tau_h(\lambda_u)$ and $\in_h(\mu_v)$ depends on the number of residues in each region $\eta_h(\mu_v)$. Therefore, $\eta_h(\lambda_u)$ and $\eta_h(\mu v)$ are defined prior to the number of states.

The numbers of residues in the u-th loop region $\eta_h(\lambda_u)$ and in the v-th transmembrane region $\eta_h(\mu_v)$ are defined as:

$$\begin{aligned} \eta_h(\lambda_u) &:= \sum_{t=1}^{T_h} \delta_{F_t(h),\lambda_u}, \qquad u = 1, \cdots, m+1 \\ \eta_h(\mu_v) &:= \sum_{t=1}^{T_h} \delta_{F_t(h),\mu_v}, \qquad v = 1, \cdots, m \end{aligned}$$

where δ_{ij} is the Kronecker delta.[4] Note that the sum of η is equal to the total length of the observation sequence T:

$$\sum_{u=1}^{m+1} \eta_h(\lambda_u) + \sum_{v=1}^{m} \eta_h(\mu_v) = T(h)$$

Learning $\tau_h(\lambda_u)$: $\tau_h(\lambda_u)$, the number of states in a submodel for a loop region $H_h(\lambda_u)$, is defined as:

$$\tau_h(\lambda_u) := \begin{cases} 1 & \eta_h(\lambda_u) < 130 \\ 4 \times \lfloor \rho \times \eta_h(\lambda_u) \rfloor & 130 < \eta_h(\lambda_u) < 300 \\ 6 & 300 < \eta_h(\lambda_u) \end{cases} \tag{5}$$

[4] If $i = j$, $\delta_{i,j} = 1$, otherwise $\delta_{i,j} = 0$.

where p is a hyperparameter. Here, $[x]$ denotes a floor function, which yields the largest integer less than or equal to x.

Learning $\tau_h(\mu_v)$: The number of states in a submodel for a transmembrane region $H_h(\mu_v)$, $\tau_h(\mu_v)$, is a parameter to be tuned.

Note:

The total number of states N can be described as:

$$N := \sum_{v=1}^{\mathcal{M}_h} \tau_h(\mu_v) + \sum_{u=1}^{\mathcal{M}_{h+1}} \tau_h(\lambda_u).$$

Step 3: State transition probabilitie:

Learning a_{ij}: For each state q_i of a submodel for a loop region $H_h(\lambda_u)$, set

$$\hat{a}_{ij}(\lambda_u) := \begin{cases} 1 - \frac{\alpha_\lambda}{\eta_h(\lambda_u)} & j = i \\ \frac{\alpha_\lambda}{\eta_h(\lambda_u)} & j = i+1 \\ 0 & otherwise \end{cases}$$

where α_λ is a parameter to be tuned.

For each state q_i of a submodel for a transmembrane region $H_h(\mu_v)$, set

$$\hat{a}_{ij}(\mu_v) := \begin{cases} \alpha_\mu & j = i \\ 1 - \alpha_\mu & j = i+1 \\ 0 & otherwise \end{cases}$$

where α_μ is a parameter to be tuned.

Step 4: KD index emission probabilities:

Learning b^1_{ik1}*: Step 4.1 (Flooring):* For each state qi of a submodel for loop region $H_h(\lambda_u)$, let

$$\tilde{b}^1_{ik_1}(\lambda_u) := \frac{n(\{KD\}, k_1; \lambda_u) + \beta_\lambda}{\sum_{\substack{KD\ Index\ k_1 \\ within\ \mathcal{H}_h(\lambda_u)}} n(\{KD\}, k_1; \lambda_u) + \beta_\lambda}$$

For eash state q_i of a submodel for transmembrane region $H_h(\mu_v)$, set

$$\tilde{b}^1_{ik_1}(\mu_v) := \frac{n(\{KD\}, k_1; \mu_v) + \beta_\mu}{\sum_{\substack{KD\ Index\ k_1 \\ within\ \mathcal{H}_h(\mu_v)}} n(\{KD\}, k_1; \mu_v) + \beta_\mu}$$

where

$n(\{KD\}, k_1; \lambda_u)$:= number of residues with KD index k_1 within loop region $H_h(\lambda_u)$.

$n(\{KD\}, k_1; \mu_v)$:= number of residues with KD index k_1 within transmembrane region $H_h(\mu_v)$, and

β_λ and β_μ are hyperparameters.

Step 4.2 (Anti-overfitting):
The following operation is performed to avoid overfitting problems. As mentioned earlier, this operation is also known as "smoothing". However, to avoid confusion, the term "anti-overfitting" is used in this paper.

$$\hat{b}^1_{ik_1}(\mu_v) := \frac{1}{\zeta_i} \sum_{j:|k_1-k_j|\leq 1} v_j \tilde{b}^1_{ik_1}(\mu_v)$$
$$\hat{b}^1_{ik_1}(\lambda_u) := \frac{1}{\zeta_i} \sum_{j:|k_1-k_j|\leq 1} v_j \tilde{b}^1_{ik_1}(\lambda_u)$$
$$v_j := \int_{x=|k_1-k_j|-\frac{1}{2}}^{x=|k_1-k_j|+\frac{1}{2}} \exp\left(\frac{-x^2}{2\pi\sigma^2}\right) dx$$

where σ is a hyperparameter and ξ_i is a normalization constant.

In this algorithm, the emission probabilities are the same within individual submodels. Note that Step 4.2 would have been impossible if the nearness between two amino acids were note defined, which is the case when considering the sequence of the 20 letter symbols. Also note that there are four amino acids out of 20 that have the same KD index (-3.5): ASP, ASN, GLU, and GLN.

Step 5: Charge emission probabilities:
Learning b^2_{ik2}: For state q_i of a submodel for loop region $H_h(\lambda_u)$, let

$$\hat{b}^2_{ik_2}(\lambda_u) := \frac{n(\{Charge\}, k_2; \lambda_u) + \gamma_\lambda}{\sum_{\substack{Charge\ k_2 \\ within\ \mathcal{H}_h(\lambda_u)}} n(\{Charge\}, k_2; \lambda_u) + \gamma_\lambda}$$

For each state q_i of a submodel for transmembrane region $H_h(\mu_v)$, set

$$\hat{b}^2_{ik_2}(\mu_v) := \frac{n(\{Charge\}, k_2; \mu_v) + \gamma_\mu}{\sum_{\substack{Charge\ k_2 \\ within\ \mathcal{H}_h(\mu_v)}} n(\{Charge\}, k_2; \mu_v) + \gamma_\mu}$$

where

$n(\{Charge\}, k_2; \lambda_u)$:= number of residues with formal charge k_2 within loop region $H_h(\lambda_u)$,

$n(\{Charge\}, k_2; \mu_v)$:= number of residues with formal charge k_2 within transmembrane region $H_h(\mu_v)$, and γ_λ and γ_μ are hyperparameters.

Step 6: Initial state probability π_i:
Since the proposed model utilizes left-to-right topology, the initial state probability π_i is set as follows:

$$\pi_i := \begin{cases} 1 & if\ i = 1 \\ 0 & otherwise \end{cases} \quad (6)$$

Here, all steps of the learning phase are summarized below.

Learning

For $h = 1, \cdots, H$, the parameters $\hat{w_h}$ associated with model $\mathcal{H}_h$ are trained by the training data D_h as:

1) Define the number of transmembrane regions $\mathcal{M}_h$
2) Learn the number of states $\tau_h(\lambda_u)$ and $\tau_h(\mu_v)$
3) Learn state transition probabilities a_{ij}
4) Learn KD index emission probabilities $b^1_{ik_1}$
5) Learn charge emission probabilities $b^2_{ik_2}$
6) Set initial state probability π_i

Note:

We chose not use the Baum-Welch-method learning algorithm for two reasons. First, it often suffers from local minima. Second, we wanted to test our first trial parameter values so that our proposed structure would make sense. Of couse, the learning scheme must be improved in various ways including using Monte Carlo methods, which is a subject of ongoing research.

Predictions

The proposed algorithm is designed to have two steps in the prediction phase: 1) selecting the best model, and 2) predicting transmembrane regions.

Let $D_{test} := \{O_t^{test}\}_{t=1}^{T_{test}}$ be a test sequence. Note that in the prediction phase, m and the associated annotation sequence $\{F_t\}$ are unknown. As previously mentioned, the state sequence $(Q_t\}$ is also unobserved.

Step 1: Selection of the Best Model:

In this step, given a test residue sequence D_{test}, the best model H_h is selected. H_h is the model that gives the largest joint probability distribution of $\{O_t, Q_t\}_{t=1}^{T_{test}}$.

$$\begin{aligned}\hat{h} &:= \underset{h}{\operatorname{argmax}} \left[P\left(D_{test}, Q_{T_{test}} = q_N \mid \hat{w}, \mathcal{H}_h\right)\right] \\ &= \underset{h}{\operatorname{argmax}} \left[\sum_{\substack{all\ possible \\ paths \{Q_t\}_{t=1}^{T_{test}-1}}} P\left(\{O_t^{test}\}, \{Q_t\}, Q_{T_{test}} = q_N \mid \hat{w}, \mathcal{H}_h\right)\right]\end{aligned}$$

Step 2: Prediction of Transmembrane Regions:

In this step, given a test residue sequence $D_{test} := \{O_t^{test}\}_{t=1}^{T_{test}}$, each residue O_t^{test} was predicted to belong or not to belong to a transmembrane region by using the unobserved state sequence $\{O_t^*\}_{t=1}^{T_{test}}$.

An amino acid residue associated with O_t^{test} was predicted to be in the *v*-th transmembrane region μ_v if $Q^*_t \in H_h(\mu_v)$, or in the *u*-th Loop Region λ_u if $Q^*_t \in H_h(\lambda_u)$.

First, let us recall our prediction algorithm proposed in [30]. In [30], $\{Q^*_t\}_{t=1}^{T_{test}}$ is predicted as:

• for $t = 1$

$Q^*_1 := q_1$

• for $t > 1$

$$Q_t^* := \underset{q_i \in \{q_j, q_{j+1}\}}{\operatorname{argmax}} P(Q_t = q_i \mid \boldsymbol{O}_1^{test}, \cdots, \boldsymbol{O}_T^{test}, \hat{w}, \mathcal{H}_{\hat{h}})$$
$$where\ Q_{t-1}^* = q_j \tag{7}$$

We attempted to predict $\{O^*_t\}_{t=1}^{T_{test}}$ by performing the following operations:

Prediction of Transmembrane Regions

- for $t = 1$

  $Q_1^* := q_1$

- for $t > 1$

  $$Q_t^* := \underset{q_i}{\operatorname{argmax}} P(Q_t = q_i \mid \boldsymbol{O}_t^{test}, \cdots, \boldsymbol{O}_{T_{test}}^{test}, Q_{t-1}^*, \hat{w}, \mathcal{H}_{\hat{h}}) \tag{8}$$

Note that both schemes begin with the same quantity q_1 at $t = 1$. In order to see differences in the two prediction schemes, note that equation (7) searches for the state which maximizes

$$P(Q_t = q_i \mid \boldsymbol{O}_1^{test}, \cdots, \boldsymbol{O}_{t-1}^{test}, \boldsymbol{O}_t^{test}, \cdots, \boldsymbol{O}_T^{test}) \tag{9}$$

whereas equation (8) maximizes

$$P(Q_t = q_i \mid \boldsymbol{O}_t^{test}, \cdots, \boldsymbol{O}_T^{test}, Q_{t-1}^*). \tag{10}$$

In equation (9), the target probability is conditional on all the observation data ($\boldsymbol{O}_t^{test}, \cdots, \boldsymbol{O}_T^{test}$), whereas equaiton (10) does not take into account the earlier observation data ($\boldsymbol{O}_1^{test}, \cdots, \boldsymbol{O}_{t-1}^{test}$). The latter equation (10), however, incorporates the previous state, whereas the former equation does not consider the previous state. Therefore, the proposed prediction scheme (8) emphasizes the dependency on the

previous state Q^*_{t-1} over the previous state has been estimated. Although the observation data contains information about Q_t, they contain uncertainties as well because they are regarded as realizations of random variables. Since the presumed topology is highly structured (left-to-right topology), information could be of better quality than (7).

Since our proposed scheme attempts to estimate Qt given $(O_t^{test}, \cdots, O_T^{test})$, this could be regarded as a smoothing scheme, that is, a "*backward*" *smoothing*.

From a biological point of view, this scheme, could be explained as follows:

The proposed prediction scheme is designed to predict its annotation from $t = 1$, which is the N-terminus of the protein. Since the translation process in protein biosynthesis starts from the N-terminus, this scheme is anticipated yo yield good results. Also, as an amino acid chain grows in the translation process, amino acids are added at the carboxyl end of the chain. The growing chain, as the amino acids are added to the end, will immediately tend to fold into a particular conformation. Because of this tendency, when predicting the state Q^*_t at position t, it seems natural to use the sequence information from t to T_{test}, but not from 1 to $t - 1$.

For comparison, another method may be constructed by taking into account the observation $(O_t^{test}, \cdots, O_T^{test})$:

- for $t = T$

$$Q_T^* := q_N$$

- for $t < T$

$$Q_t^* := \underset{q_i \in \{q_j, q_{j-1}\}}{\operatorname{argmax}} P(Q_t = q_i, O_1^{test}, \cdots, O_t^{test}, \mid \hat{w}, \mathcal{H}_{\hat{h}}) \tag{11}$$

where $Q^*_{t+1} = qj$.

Note that this scheme begins with the quantity q_N at $t = T$, which is the C-terminus of a protein. This method, that is, the "*forward*" *smoothing* scheme, uses the observation data $(O_t^{test}, \cdots, O_T^{test})$ to estimate Q^*_t. In contrant, the proposed "*backward*" *smoothing* method described in (8) begins with the quantity q_1 at $t = 1$, and uses the observation data $(O_t^{test}, \cdots, O_T^{test})$.

All steps of the prediction phase can be summarized as follows:

Prediction

For test data $D_{test} := \{O_t^{test}\}$:

1) Select the best model by:

$$\hat{h} := \underset{h}{\operatorname{argmax}} [P(D_{test}, Q_{T_{test}} = q_N \mid \hat{w}, \mathcal{H}_h)]$$

2) Predict transmembrane region by predicting $\{Q_t^*\}_{t=1}^{T_{test}}$.

Evaluation

In this section, we report the evaluation results of the novel algorithm we used. We then discuss the performances of the five prediction methods applied to our proposed model. The results summarized in Table II.

(i) The proposed method ("backward" smoothing) described in (8) (referred to as Method (i)).
(ii) Our previous method reported in [30] (referred to as Method (ii)).
(iii) The standard Viterbi method (referred to as Method (iii)).
(iv) A standard smoothing method described in (1) (referred to as Method (iv)).
(v) A "forward" smoothing method described in (11) (referred to as Method (v)).

In order to perform experiments, appropriate data sets must be obtained. Currently, one of the most difficult problems in protein structure prediction in general, and in transmembrane protein structure prediction in particular, is the difficulty in obtaining appropriate data sets for experiments. We used two publicly available data sets, one collected by Möller et al. [36], and another collected by Kernytsky et al. [37].

The accuracy of the predictions of our algorithm as to whether particular amino acids were from a transmembrane region is discussed below. Evaluation methods follow those used in möller et al. [38]. The details of the evaluation methods are described in section III-2.

For comparison, using the same test data sets, we also tested the performance of TMHMM [10][5], HMMTOP [28][6], and SOSUI [29][7], which are three well-known transmembrane structure prediction tools.

1) Data sets: Here, we describe the details of the data sets used in our experiments. One is a data set collected by Möller et al., which is a well-characterized transmembrane protein data set [36]. This data set will be called Dataset1 in this paper. The other is a data set collected by Kernytsky et al., which is a benchmarking data set [37]. This data set will be called Dataset2 in this paper. The sequences were downloaded from their websites.[8] For training HMM parameters, sequences from Dataset1 were used. For evaluation, sequences from Dataset2 were used.

Since the sequences in Dataset1 were collected from the SwissProt database released in the year 2000, the annotations have been updated. To copy with this change, the authors updated the annotations and sequences by searching the UniPort database by ID or accession number.

In order to validate the performance of the proposed algorithm, sequences from Dataset2 were used [37]. Dataset2 contains 2247 sequences, but without descriptions of origins or annotations. Since the proposed algorithm targets only transmembrane

[5] http://www.cbs.dtu.dk.services/TMHMM-2.0/
[6] http://www.enzim.hu/mhhtop/
[7] http://bp.nuap.nagoya-u.ac.jp/sosiu/
[8] Data set collected by Möller et al.: ftp://ftp.ebi.ac.uk//testsets/transmembrane
Data set collected by Kernytsky et al.: http://cubic.bioc.columbia.edu/services/tmh_benchmark/

proteins, we were required to select only transmembrane proteins.

In order to sect transmembrane proteins out of the 2247 sequences in Dataset2, we ran a FASTA search against the entire UniPort database. We found 128 complete matches that were annotated as transmembrane proteins in UniProt. All 128 sequences were used for testing.

Of the amino acid sequences in Dataset1 and Dataset2, those with the following clear annotations are used for out experiment:

DOMAIN CYTOPLASMIC, DOMAIN MATRIX, DOMAIN EXTRACELLULAR, DOMAIN INTERMEMBRANE, DOMAIN PERIPLASMIC, and TRANSMEM for which we have interpreted CYTOPLASMIC, MATRIX, EXTRACELLULAR, INTERMEMBRANE, and PERIPLASMIC as loop segments with TRANSMEM as a transmembrane segment.

The number of sequences used for training and testing is shown in Table I. Thus, using the two data sets described above, training data sets and test data sets were selected as follows:

Training dataset:

244 sequences of Dataset1 were used for training.

Test dataset:

128 sequences of Dataset2 were used for testing.

Table 1: Number of Sequences of Datasets used for Evaluation:

m	**Training**	**Testing**
1	69	34
2	17	9
3	19	8
4	38	18
5	13	8
6	18	16
7	28	11
8	6	6
9	3	1
10	8	5
11	1	0
12	21	11
13	1	0
14	1	1
15	1	0
Total	**244**	**128**

2) Evaluation criteria: The evaluation criteria of transmembrane region prediction follows the method described in [38]. In order to define this performance criterion, consider:

True Positive (TP) Segments:
A TP segment must share *at least nine residues* with a transmembrane region of the reference annotation. The following gives a schematic of this concent, where "T" stands for an amino acid within a transmembrane region, while "-" stands for an amino acid in a loop region.

```
Annotated  -------TTTTTTTTTTTTTTT--------
Predicted  -----TTTTTTTTTTTTTTT---------
```

False Negative (FN) Segments:
An FN segment is a transmembrane region that is not predicted; this is schematically described by:

```
Annotated  -------TTTTTTTTTTTTTTT--------
Predicted  ------------------------------
```

False Positive (FP) Segments:
An FP segment is a predicted transmembrane region that is not a transmembrane region in the reference protein test set. This is schematically described by:

```
Annotated  ------------------------------
Predicted  -------TTTTTTTTTTTTTTT--------
```

Note: Each predicted transmembrane region should correspond to only one reference transmembrane region. This excludes the posibility of double counting TP segments. For instance, the following prediction has one TP and one FN instead of two TP segments:

```
Annotated  -----TTTTTTTTT-TTTTTTTTTT-----
Predicted  -----TTTTTTTTTTTTTTTTTTT------
```

Accuracy of transmembrane region prediction is defined by:
Transmemebrane region prediction accuracy

$$:= \left(1 - \frac{n(FN) + n(FP)}{n(TP) + n(FN)}\right)$$

where $n(TP)$, $n(FN)$, and $n(FP)$ denote the number of True Positive segments, False Negative segments, and False Positive Segments, which we presume is the criterion in Möfller et al. [38]. However, the equation is not explicitly written.

3) Results: The results are shown in Table II. The proposed "backward" smoothing method (Method (i)) gave:

$n(TP) = 585$, $n(FN) = 29$, $n(FP) = 18$, and

Transmemebrane region prediction accuracy = 92.3%.

Figure 3 illustraces some of the prediction results using the "backward" smoothing method. Note that Methods (i) through (v) use the same model and same parameters. As one can see from Table II, however, the proposed "backward" smoothing method (Method (i)) performs reasonably well.

Precise comparisons with other prediction algorithms are difficult because the sequences used for their training could have been different, as mentioned earlier. For

comaprison purposed, however, we tested the same test sequences against three well-known tools for predicting transmembrane helices.

Table II: Accuracies of Transmembrane Region Predictions

Methods/tools	**n(TP)**	**n(FN)**	**n(FP)**	**Accuracy**
Method (i)	585	29	18	92.3%
Method (ii)	582	32	21	91.4%
Method (iii)	582	32	21	91.4%
Method (iv)	567	47	23	88.6%
Method (v)	583	31	20	91.7%
TMHMM	563	51	28	87.1%
HMMTOP	580	34	47	86.8%
SOSUI	544	70	29	83.9%

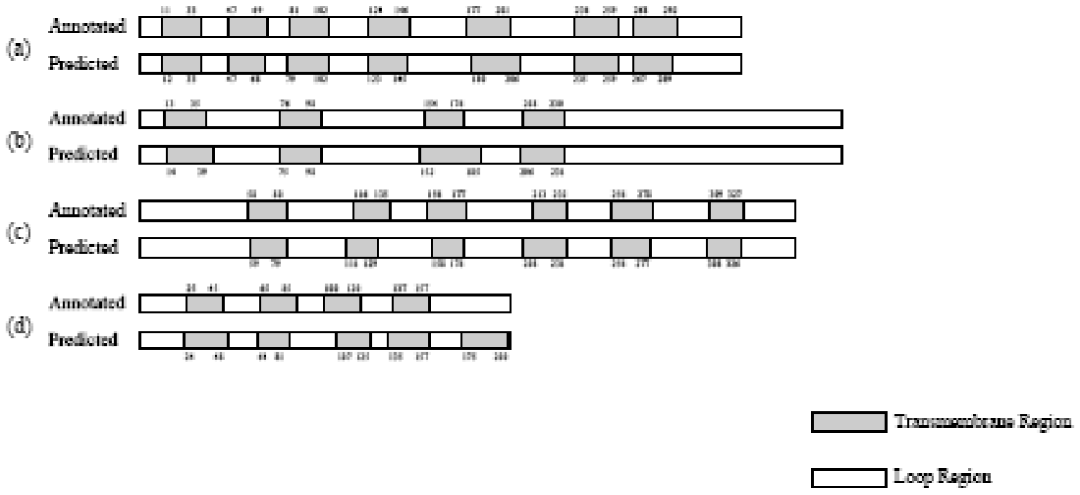


Figure 3: Typical Transmembrane Region Prediction Results. Numbers denote number of residues from the N-terminus.

Conclusions

We proposed a novel scheme (the "backward" smoothing scheme) to predict transmembrane regions utilizing a finitestate, stochastic dynamical system. The proposed prediction scheme described in (8) emphasizes the dependency on the previous state Q^*_{t-1} over the previous observation data (O_1, …, O_{t-1}) once the previous state has been estimated. Since the model structure employs a left-to-right topology, the proposed scheme was anticipated to yield better results than the prediction scheme reported in [30]. The experimental results reported in section III-.3 suggest that the "backward" smoothing scheme did perform reasonably well.

Although it produced reasonable results, the algorithm has several drawbacks and a number of aspects that need to be improved, as described below.

i. The "backward" smoothing scheme is a general form that could be applied to applications of HMM. Applying the proposed scheme to other models, such as Bayesian HMM [34], is an interesting idea, and further improvement is anticipated.
ii. When the probability distribution landscape is not simple, a one-time parameter/state estimation, including the Baum-Welch method, as well as the algorithm reported in this paper, has limited success. The proposed learning algorithm is too simplistic, and a more advanced procedure is called for. Parameters, hyperparameters, and states as well can be inferred via a Bayesian framework where the Monte Carlo method can be utilized [32]. This is the subject of our ongoing research.
iii. Sidedness (interior/exterior) can be predicted in situations where formal charge trajectories could be more important than the present problems.

Acknowledgments

We would like to express our appreciation to Professor Keiji Wada of the National Center of Neurology and Psychiatry, Dr. Kenji Mizuguchi of the National Institute of Biomedical Innovation, Mr. Takayuki Ohnishi of the Tokyo Medical Dental University, Mr. Yohei Nakada and Mr. Takahiro Hamada of Waseda University, and Professor Steffen M¨oller of University of L¨ubeck for their advice.

References

[1] Hettema, E.H., Distel, B., and Tabak, H.F., 1999, "Import of proteins into peroxisomes," Biochim. Biophys. Acta., 1451(1), pp. 17–34.
[2] Patil, C., and Walter, P., 2001, "Intracellular signaling from the endoplasmic reticulum to the nucleus: the unfolded protein response in yeast and mammals," Curr Opin Cell Biol, 13(3), pp. 349–355.
[3] Le Borgne, R., and Hoflack, B., 1998, "Protein transport from the secretory to the endocytic pathway in mammalian cells," Biochim Biophys Acta, 1404(1-2), pp. 195–209.
[4] Marchese, A., George, S. R., Kolakowski, L. F. Jr., Lynch, K. R., and O'Dowd, B. F., 1999, "Novel GPCRs and their endogenous ligands: expanding the boundaries of physiology and pharmacology," Trends Pharmacol Sci., 20(9), pp. 370–375.
[5] Drews, J., 2000, "Drug Discovery: A Historical Perspective," Science, 287(5460), pp. 1960–1964.
[6] Wallin, E., and von Heijne, G., 1998, "Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms," Protein Sci., 7(4), pp. 1029–1038.
[7] Blundell, T. L., Mizuguchi, K.,2000, "Structural genomics: an overview," Prog Biophys Mol Biol., 73(5), pp. 289-295.
[8] Caffrey, M., 2003, "Membrane protein crystallization," J. Struct. Biol., 142(1),

pp. 108-132.
[9] W¨uthrich, K., 2001, “The way to NMR structures of proteins,” Nature Struct. Biol., 8(11), pp. 923-925.
[10] Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. L., 2001, “Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes,” J. Mol. Biol., 305(3), pp. 567–580.
[11] Chen, C. P., and Rost, B., 2002, “State-of-the-art in membrane protein prediction,” Appl. Bioinformatics, 1(1), pp. 21–35.
[12] Zhou, C., Zheng, Y., and Zhou, Y., 2004, “Structure prediction of membrane proteins,” Genomics Proteomics Bioinformatics, 2(1), pp. 1–5.
[13] Kyte, J., and Doolittle, R. F., 1982, “A simple method for displaying the hydropathic character of a protein,” J. Mol. Biol., 157(1), pp. 105–132.
[14] Eisenberg, D., Weiss, R. M., and Terwilliger, T. C., 1982 “The helical hydrophobic moment: a measure of the amphiphilicity of a helix,” Nature 299(5881), pp. 371–374.
[15] Klein, P., Kanehisa, M., and DeLisi, C., 1985, “The detection and classification of membrane-spanning proteins,” Biochim. Biophys. Acta., 815(3), pp. 468–476.
[16] von Heijne, G., 1992, “Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule,” J. Mol. Biol., 225(2), pp. 487–494.
[17] Nakai, K., and Kanehisa, M., 1992, “A knowledge base for predicting protein localization sites in eukaryotic cells,” Genomics, 14(4), pp. 897–911.
[18] Persson, B., and Argos, P., 1996, “Topology prediction of membrane proteins,” Protein Sci., 5(2), pp. 363–371.
[19] Cserzö, M., Wallin, E., Simon, I., von Heijne, G., and Elofsson, A., 1997, “Prediction of transmembrane alpha-helices in prokaryotic membrane proteins: the dense alignment surface method,” Protein Eng., 10(6), pp. 673–676.
[20] Juretic, D., Zucic, D., Lucic, B., and Trinajstic, N., 1998, “Preference functions for prediction of membrane-buried helices in integral membrane proteins,” Comput. Chem., 22(4), pp. 279–294.
[21] Pasquier, C., Promponas, V. J., Palaios, G. A., Hamodrakas, J. S., and Hamodrakas, S. J., 1999, “A novel method for predicting transmembrane segments in proteins based on a statistical analysis of the SwissProt database: the PRED-TMR algorithm,” Protein Eng., 12(5), pp. 381-385.
[22] Jayasinghe, S., Hristova, K., and White, S. H., 2001, ”Energetics, stability, and prediction of transmembrane helices,” J. Mol. Biol., 312(5), pp. 927–934.
[23] Deber, C. M., Wang, C., Liu, L. P., Prior, A. S., Agrawal, S., Muskat, B. L., and Cuticchia, A. J., 2001, “TM Finder: a prediction program for transmembrane protein segments using a combination of hydrophobicity and nonpolar phase helicity scales,” Protein Sci., 10(1), pp. 212–219.
[24] Jones, D. T., Taylor, W. R., and Thornton, J. M., 1994, “A model recognition approach to the prediction of all-helical membrane protein structure and topology,” Biochemistry, 33(10), pp. 3038–3049.

[25] Rost, B., Fariselli, P., and Casadio, R., "Topology prediction for helical transmembrane proteins at 86% accuracy," Protein Sci., 5(8), pp. 1704–1718.

[26] Fariselli, P., and Casadio, R., "HTP: a neural network-based method for predicting the topology of helical transmembrane domains in proteins," Comput. Appl. Biosci., 12(1), pp. 41-48.

[27] Sonnhammer, E. L. L., von Heijne, G., and Krogh, A., 1998, "A Hidden Markov Model for Predicting Transmembrane Helices in Protein Sequences," Proc. 6th Int. Conf. Intell. Syst. Mol. Biol., Canada, Montreal, pp.175-182.

[28] Tusnady, G. E., and Simon, I., 1998, "Principles governing amino acid composition of integral membrane proteins: application to topology prediction," J. Mol. Biol., 283(2), pp. 489–506.

[29] Hirokawa, T., Boon-Chieng, S., and Mitaku, S., 1998, "SOSUI: classification and secondary structure prediction system for membrane proteins," Bioinformatics, 14(4), pp. 378–379.

[30] Kaburagi, T., Muramatsu, D., and Matsumoto, T., 2007, "Transmembrane Structure Predictions with Hydropathy Index/Charge Two-Dimensional Trajectories of Stochastic Dynamical Systems," J. Bioinformatics and Computational Biology, 5, (in press).

[31] Voet, D., and Voet, J. G., 1995, Biochemistry (Second Edition), J. Wiley & Sons, New York, p. 49.

[32] Matsumoto, T., Nakajima, Y., Saito, M., Sugi, J., and Hamagishi, H., 2001, "Reconstructions and predictions of nonlinear dynamical systems: A Hierarchical Bayesian Approach," IEEE Trans. Signal Processing, 49(9), pp. 2138–2155.

[33] Yasuda, H., Takahashi, K., and Matsumoto, T., 2000, "A discrete HMM for online handwriting recognition," Int. J. Pattern Recognition and Artificial Intelligence, 14(5), pp. 675–688.

[34] Sasaki, H., Nakada, Y., Kaburagi, T., and Matsumoto, T., 2007, "Bayesian Angle Information HMM with a von Mises Distribution and its Implementation using a Bayesian Monte Carlo Method," Proc. European Symposium on Time Series Prediction, Finland, Otaniemi, pp. 29–38.

[35] Muramatsu, D., and Matsumoto, T., 2003, "An HMM On-line Signature Verifier Incorporating Signature Trajectories," Proc. Int. Conf. On Document Analysis and Recognition, Scotland, Edinburgh, pp. 438–442.

[36] Möller, S., Kriventseva, E. V., and Apweiler, R., 2000, "A collection of well characterized integral membrane proteins," Bioinformatics, 16(12), pp. 1159–1160.

[37] Kernytsky, A., and Rost, B., 2003, "Static benchmarking of membrane helix predictions," Nucleic Acids Research, 31(13), pp. 3642-3644.

[38] Möller, S., Croning, M. D. R., and Apweiler, R., 2001, "Evaluation of methods for the prediction of membrane spanning regions," Bioinformatics, 17(7), pp. 646–653.