RESEARCH ARTICLES

Predicting Peptides That Bind to MHC Molecules Using Supervised Learning of Hidden Markov Models

Hiroshi Mamitsuka*
C&C Media Research Laboratories, NEC Corporation, Kawasaki, Kanagawa, Japan

ABSTRACT **The binding of a major histocompatibility complex (MHC) molecule to a peptide originating in an antigen is essential to recognizing antigens in immune systems, and it has proved to be important to use computers to predict the peptides that will bind to an MHC molecule. The purpose of this paper is twofold: First, we propose to apply supervised learning of hidden Markov models (HMMs) to this problem, which can surpass existing methods for the problem of predicting MHC-binding peptides. Second, we generate peptides that have high probabilities to bind to a certain MHC molecule, based on our proposed method using peptides binding to MHC molecules as a set of training data. From our experiments, in a type of cross-validation test, the discrimination accuracy of our supervised learning method is usually approximately 2–15% better than those of other methods, including backpropagation neural networks, which have been regarded as the most effective approach to this problem. Furthermore, using an HMM trained for HLA-A2, we present new peptide sequences that are provided with high binding probabilities by the HMM and that are thus expected to bind to HLA-A2 proteins. Peptide sequences not shown in this paper but with rather high binding probabilities can be obtained from the author (E-mail: mami@ccm.cl.nec.co.jp). Proteins 33:460–474, 1998.**

Key words: major histocompatibility complex; antigen; stochastic models; machine learning; protein docking; computational biology; immunology

INTRODUCTION

An important role of immune systems is to detect antigens originating in foreign (or self) organizations. This process is known to be regulated by a major histocompatibility complex (MHC) molecule, which can bind to a peptide corresponding to a part of foreign (or self) proteins. An MHC molecule binding to a peptide presents the peptide to a T-cell receptor on the surface of a cell, so that the receptor can recognize the peptide (antigen) (Rammensee et al., 1993). In short, the binding of an MHC molecule to a peptide is an essential step in the functioning of an immune system. However, not all peptides are capable of binding to an MHC molecule, and hence it is important to determine which peptides can bind to a given MHC molecule, for various purposes. (About 1 in 100–200 peptides are said to bind to an MHC.) For example, for designing effective peptide vaccines, it is necessary to find out what peptides can bind to each of the distinct MHC molecules of various patients.

This problem of predicting peptides that bind to MHC molecules has been extensively investigated since around the late 1980s or early 1990s. However, experimental approaches for determining the binding ability of peptides requires time-consuming and costly steps, e.g., synthesis of peptides and measuring their binding ability. This feature of biochemical assays has made it extremely difficult for experimental researchers to determine thoroughly the sequences of peptides binding to MHC proteins. In striving to attain greater efficiency in performing such investigation, efforts are being made to use computers in predicting peptides that will bind to an MHC molecule (or activate T-cell proliferation), and what is now desired is a new precise computational approach to this problem (Gulukota et al., 1997).

Existing computational approaches can be roughly classified into two types: 1) methods that use a number of peptides whose binding ability is experimentally determined; and 2) methods that do not.

An example of the second category is one using a protein threading approach usually used for predicting protein 3D structures (Altuvia et al., 1995). This

*Correspondence to: Hiroshi Mamitsuka, C&C Media Research Laboratories, NEC Corporation, 4-1-1, Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 216 8555, Japan. E-mail: mami@ccm.cl.nec.co.jp

Received 12 December 1997; Accepted 10 August 1998

approach predicts the peptides that fit into an MHC groove, using a table representing the preference of sterically neighboring amino acid pairs, which is already prepared in the context of protein 3D structure prediction (Bowie et al., 1991; Sippl, 1990). On the other hand, our method belongs to the first category, and hence, we regard existing methods in the category as previous work of our method. Previous work can be classified into the following three categories, according to the models used: 1) simple motifs; 2) detailed motifs (matrix models); and 3) artificial neural networks.

The simplest method uses a sequence motif as a predictor, which is determined from a large number of existing known binding peptides (Falk et al., 1991; Rammensee et al., 1995). The specific amino acids appearing in a motif are called *anchor residues,* e.g., an HLA-A2-restricted peptide motif of nine residues has Leu at the second and Val at the ninth position as its anchor residues. Such a motif can be characterized as one that focuses on partial frequent sequence features (patterns) of the peptides having the ability to bind to an MHC molecule (Margalit et al., 1987; Rothland and Taylor, 1988). However, such simple partial information of binding peptides has proved to be insufficient to explain the comprehensive binding ability of a given peptide (e.g., Ruppert et al., 1993; Bouvier and Wiley, 1994).

Further modification of the motifs reaches a matrix model as a predictor, in which each column corresponds to a position of a certain length of peptides, and each row corresponds to an amino acid (Parker et al., 1994; Kondo et al., 1995; Davenport et al., 1995; Brusic et al., 1997b; Gulukota et al., 1997). Each entry of the matrix indicates a kind of binding strength of an amino acid at a position specified by a row and a column, respectively. These entries are calculated from actual values obtained from biochemical assays. The matrix can be recognized as a detailed version of a sequence motif, but this representation assumes that each residue of a peptide independently relates to peptide binding to MHC molecules.

In response to this shortcoming of matrix models, layered neural networks have been used to discriminate binding peptides from non-binding ones (Bisset and Fierz, 1993; Brusic et al., 1994; Adams and Koziol, 1995; Gulukota et al., 1997). Parameters within a neural network are trained by the backpropagation algorithm with a number of peptides whose binding ability is already known, and the trained network (hereafter termed *backpropagation neural network*) predicts whether a given unknown peptide can bind or not. Gulukota et al. (1997) reported that the performance of the backpropagation neural networks exceeds those of matrix models and motifs in discriminating peptides that bind to an MHC molecule from other peptides.

There are three major problems in the previous work that has been published. The first of these is that all these methods assume that the size of peptides that bind to MHC molecules is fixed, though actually the length of peptides that bind to MHC molecules is *variable* and can range from 8 to more than 20 residues. Thus, existing methods cannot predict the binding ability of a peptide whose length is longer or shorter than that of the peptides used in training, and thus, available training and test data are extremely limited. The second problem is that both matrix models and motifs present only one sequence pattern in the given set of data that will bind to an MHC molecule. They cannot extract *multiple* sequence patterns hidden in a given set of data separately, even if each of them has sufficient binding ability. The third problem occurs with neural networks. Even though these networks are able to learn such multiple sequence patterns in a given set of data automatically, the network parameters are given only as real-valued weights attached to edges connecting nodes in the network. Consequently, the weights cannot present any *understandable* training results.

To overcome the shortcomings of the previous methods, we propose to apply supervised learning of a hidden Markov model (HMM), which has been widely used in the fields of speech recognition (Rabiner, 1989; Lee, 1989) and computational molecular biology (Churchill, 1989; Baldi et al., 1994; Krogh et al., 1994; Eddy, 1996).

HMMs are suitable for representing time-series sequences (strings) having flexible lengths, and since the early 1990s they have been vigorously applied to the problem of automatically aligning multiple biological sequences. As HMMs can deal with data having a variety of lengths, they can solve the first of the above problems. In our experiments, we use a fully connected HMM, which can automatically divide multiple sequence patterns hidden in a given set of data into separate patterns. Thus, our HMM will be able to solve the second of the problems. Furthermore, a trained HMM can be presented as a comprehensible form, just like a sequence profile derived from multiple sequence alignment. This feature of HMMs enables them to solve the third of the problems.

The most popular learning algorithm of an HMM is the Baum-Welch algorithm (or the *forward-backward algorithm*). This algorithm, which belongs to a class of unsupervised learning, is a local optimization algorithm for maximum likelihood settings of probability parameters of a given model. Most approaches using HMMs in the computational biology field have trained their models based on this unsupervised learning algorithm. However, we here use a *supervised learning* algorithm that allows us to train an HMM with a set of data in which each sequence has its own target value (Mamitsuka, 1996, 1997).

This is because each sequence dealt with here is obtained through biochemical experiments, and hence has its own value indicating to what degree it can bind to an MHC molecule or activate T-cell proliferation.

Furthermore, the structure of an HMM used in this paper is different from the one used in the previous HMM-based approaches (e.g., Krogh et al., 1994). The earlier HMM (hereafter termed *alignment HMM*) was proposed for the purpose of aligning given multiple sequences, and it can be recognized as a so-called left-to-right type HMM in which an edge starting from a state must go to a state on the right side of the state or to the state itself. However, as mentioned earlier, the HMM we use is a *fully connected* HMM in which any transition between two states is allowed, except for a transition from the starting state to the finishing state and transitions from the finishing state to any other states. Roughly speaking, the alignment HMM represents only a single sequence pattern of given training data. On the other hand, a fully connected HMM should be able to represent more than one sequence pattern hidden in a set of given training data, because there is no constraint in the structure of a fully connected HMM.

In our experiments, we focused on HLA-A2, which is a human MHC class I molecule, because it is an important MHC molecule that has been widely studied in the context of immunology (e.g., Matsumura et al., 1992; Bjorkman and Burmesister, 1995); as a result, a larger amount of peptide data exists related to this than for other types of MHC molecules. The main purpose of this paper is to present peptides that have the potential to bind to this MHC molecule, using a supervised learning algorithm of an HMM and currently available data. We also present two HMMs trained by peptides that bind to HLA-DR1 and HLA-DR4, which are human MHC class II proteins, since we can obtain a large number of these peptides from a currently available database.

Two experiments were performed. First, we verified the discrimination ability of our supervised learning method compared with other two methods, i.e., a backpropagation neural network and the Baum-Welch learning of an HMM. In this experiment, we used actual peptide data in association with their real-valued ability to activate T-cell proliferation; these data were obtained from the MHCPEP database developed by Brusic et al. (1997a). The experiment was performed by conducting a cross-validation test while varying the proportion of training data to all obtained data. The result of this experiment shows that at any proportion of training data to all data, the average discrimination accuracy of our method is approximately 2–15% better than other methods, i.e., the Baum-Welch reestimation of fully connected or alignment HMMs and the backpropagation neural network, which so far has been regarded as the most accurate method in predicting MHC binding peptides.

Second, for each of three MHC proteins, including HLA-A2, we used all data obtained from the MHCPEP database and trained 100 HMMs using the data with our supervised learning algorithm. Out of the 100 models trained for an MHC protein, we chose the one that could explain the data best and showed the HMM. Using the model trained by the data of HLA-A2, we randomly generate peptides that are expected to have a high ability to bind to HLA-A2, but that are not yet known. From this experiment, we ascertained that an HMM trained by our algorithm captures frequent sequence patterns in training data separately, and found that the extracted patterns include not only existing motifs of MHC binding peptides but also new sequence patterns, each of which characterizes a part of the training peptides.

MATERIALS AND METHODS

Hidden Markov Models

We here briefly review an HMM. For more detailed information, interested readers should consult Rabiner (1989).

The structure of an HMM consists of states and one-directional edges, each of which connects two states. An HMM has two types of parameters, i.e., transition probabilities and symbol generation probabilities, and contains three types of states, i.e., normal states, starting states, and finishing states. In an HMM, a transition between two states is repeated starting from a starting state and finishing to a finishing state. The transition probability, which is the probability of moving from a state to a state when the two states are connected by an edge, is attached to the edge, and the symbol generation probability, which is the probability of emitting a symbol at a state except starting and finishing states, is attached to the state. Note that the extent to which state i is dependent on one of the states connected to state i, say state j, is given by the transition probability attached to the edge connecting from state j to state i.

Here, let a given HMM be H, the transition probability from state i to state j be a_{ij}, and the symbol generation probability of symbol c at state j be $b_j(c)$. They must satisfy the following equations:

$$0 \le a_{ij},\, b_j(c) \le 1, \qquad \sum_j a_{ij} = 1,\, \sum_c b_j(c) = 1. \quad (1)$$

When a new sequence is given to HMM H, we can calculate "forward" and "backward" probabilities from the transition probabilities a_{ij} and symbol generation probabilities $b_j(c)$. Let the given s-th symbol sequence be O^s, the length of O^s be l_s, and the t-th symbol of O^s be O^s_t, i.e., $O^s = O^s_1 \ldots O^s_{l_s}$.

We can define forward probability $\alpha^s_t(j)$, which is the probability that the model generates the first t

symbols of the input sequence O^s and arrives at state i. When the HMM H has only one starting state and one finishing state, the forward probability can be iteratively calculated from transition probabilities and symbol generation probabilities, as follows:

$$\alpha_t^s(j) = \sum_i a_{ij} b_j(O_t^s) \alpha_{t-1}^s(i) \qquad (t = 1, \ldots, l_s),$$

$$\alpha_0^s(j) = 1, \text{ if } j \text{ can be the starting state}, \qquad \textit{and}$$

$$\alpha_0^s(j) = 0, \text{ if not.}$$

$$\alpha_{l_s+1}^s(j) = \sum_i a_{ij} \alpha_{l_s}^s(i).$$

Similarly, we can define backward probability $\beta_t^s(i)$, which is the probability that the model will generate the rest (all but the first t symbols) of the input sequence O^s given that it is now at state i. For the HMM H with one starting state and one finishing state, the backward probability can be calculated from transition probabilities and symbol generation probabilities, as follows:

$$\beta_t^s(i) = \sum_j a_{ij} b_j(O_{t+1}^s) \beta_{t+1}^s(j) (t = l_s - 1, \ldots, 0),$$

$$\beta_{l_s}^s(i) = \sum_j a_{ij} \beta_{l_s+1}^s(j),$$

$$\beta_{l_s+1}^s(i) = 1, \text{ if } i \text{ can be the finishing state}, \qquad \textit{and}$$

$$\beta_{l_s+1}^s(i) = 0, \qquad \text{if not.}$$

With these probabilities, one can calculate the probability that sequence O^s is generated by the HMM H, i.e., $P(O^s | H, l_s)$, as follows:

$$P(O^s | H, l_s) = \sum_i \alpha_{l_s+1}^s(i) \beta_{l_s+1}^s(i) = \sum_i \alpha_0^s(i) \beta_0^s(i). \quad (2)$$

However, when given a sequence, the probability that the sequence is generated by an HMM is typically calculated by the Viterbi algorithm, which is obtained by replacing Σ by max in calculating forward probability $\alpha_t^s(i)$.

An Algorithm for Supervised Learning

As mentioned in the Introduction, the most popular learning algorithm of an HMM is the Baum-Welch algorithm (or the forward-backward algorithm). This algorithm is a local optimization algorithm for the maximum likelihood settings of the probabilities of a given HMM, when a set of data that should be represented by the model is given.

Our supervised learning algorithm is also a local optimization algorithm, and it gradually minimizes the difference between the *real probability* of given training data and its *target probability*. Thus, this algorithm allows us to deal with a set of data in which each peptide sequence has its own real-valued true score of binding to MHC molecules (or activating T-cell proliferation). We here briefly review our supervised learning algorithm used in this paper. Interested readers should consult Mamitsuka (1996) and Mamitsuka (1997) for further information.

As a preliminary step, we use the forward and backward probabilities to define the following two types of probabilities γ and ξ, which are used to describe our learning algorithm.

$$\gamma_t^s(i) = \frac{\alpha_t^s(i)\beta_t^s(i)}{P(O^s | H)}, \quad (3)$$

$$\xi_t^s(i,j) = \frac{\alpha_t^s(i) a_{ij} b_j(O_{t+1}^s) \beta_{t+1}^s(j)}{P\,(O^s | H)} (t = 0, \ldots, l_s - 1),$$

$$\xi_{l_s}^s(i,j) = \frac{\alpha_{l_s}^s(i) a_{ij} \beta_{l_s+1}^s(j)}{P(O^s | H)}. \quad (4)$$

Here $\gamma_t^s(i)$ corresponds to the probability of being in state i at time t given the sequence O^s and the model H, and similarly $\xi_t^s(i, j)$ indicates the probability of being in transition from state i to state j at time t given the O^s and the H.

First, we introduce the real-valued parameters ω_{ij} and $\nu_j(c)$, which can be replaced with probability parameters a_{ij} and $b_j(c)$, respectively, as follows:

$$a_{ij} = \frac{e^{\lambda \omega_{ij}}}{\sum_k e^{\lambda \omega_{ik}}}, \qquad b_j(c) = \frac{e^{\lambda \nu_j(c)}}{\sum_k e^{\lambda \nu_{j(k)}}},$$

where λ is a constant.

When given HMM H and a set of training sequences, let the number of the training sequences be n, the real probability of the s-th sequence be p_s, i.e., $p_s = P(O^s | H\ , l_s)$, and the target probability of the s–th sequence be p_s^*. Furthermore, let the difference between the real probability of the s-th sequence and its target probability be D_s, and the difference D_s is defined as $D_s = d_s^2$ where $d_s = \log\,(p_2^*/p_s)$.

We here define function g_s as follows:

$$g_s = \frac{D_{\max} - D_s}{D_{max}},$$

where D_{max} is a constant and satisfies $D_{max} > D_s (s = 1, \ldots, n)$. The function g_s is maximized to be 1 as the difference D_s reduces to zero. Hence, we define the following energy function E and try to

TABLE I. Number of Peptides Relevant to HLA-A2, HLA-DR1, and HLA-DR4, All of Which Are Obtained From the MHCPEP Database†

	NO	YL	YM	YH	Total
a. HLA-A2					
Binding	0	138	162	172	472
Activating	79 (53)	17 (6)	46 (13)	57 (30)	199 (102)
b. HLA-DR1					
Binding	0	93	152	166	411
Activating	0	17	11	1	29
c. HLA-DR4					
Binding	0	130	165	225	520
Activating	16	20	4	0	40

†Number of peptides of nine residues are shown in parentheses.

minimize it in our algorithm. Note that general unsupervised learning based on maximum likelihood uses $E = \Sigma_s - \log p_s$ instead of Eq. (5). We thus can obtain the updating rules of unsupervised learning by removing the term $d_s/(D_{max} - D_s)$ from the updating rules of our supervised learning.

$$E = \sum_s - \log g_s. \qquad (5)$$

To minimize the function E, we use a gradient-descent learning algorithm and optimize the real-valued parameters, ω_{ij} and $\nu_j(c)$.

Below, we show the updating rules for the parameters ω_{ij} and $\nu_j(c)$, which are mathematically derived according to the gradient-descent.

$$\omega_{ij}^{new} = \omega_{ij}^{old} + C_a \sum_s \frac{d_s}{(D_{max} - D_s)} \cdot \sum_{t=1}^{l_s} [\xi_t^s(i,j) - a_{ij}\gamma_t^s(i)],$$

$$v_j(c)^{new} = v_j(c)^{old} + C_b \sum_s \frac{d_s}{(D_{max} - D_s)} \cdot \sum_{t=1}^{l_s} [\gamma_t^s(j)_{O_t^s = c} - b_j(c)\, \gamma_t^s(j)],$$

where C_a and C_b are constants.

RESULTS

Data and Parameters

We obtained peptide sequences and their ability to bind to MHC molecules (and activate T-cell proliferation) from the MHCPEP database developed by Brusic et al. (1997a). Out of the 9,827 peptides in the current version of the database, there are 1,008 that are relevant to HLA-A2 molecules, which is the largest number among the peptides noted in the database. HLA-DR1 and HLA-DR4 are the only other MHC molecules to which more than 400 peptides noted in the database are individually related. Thus, we consider not only HLA-A2 but also HLA-DR1 and HLA-DR4 in our experiments.

In the MHCPEP database, there are two types of measures used in evaluating the ability of peptides, i.e., ability to bind to MHC molecules (binding peptides) and activating T-cell proliferation (activating peptides). Each peptide can be assigned one of six labels to indicate its ability: "none (NO)," "yes and little (YL)," "yes and moderate (YM)," "yes and high (YH)," "yes and unknown," and "unknown." Out of the six labels, we use only four (NO, YL, YM, and YH) because the peptides whose labels are unknown cannot be dealt with by our supervised learning method, in which the binding and activating ability of each peptide needs to be *real-valued*. Table I shows the number of peptides relevant to the MHC molecules dealt with in our experiments. The table shows *all* data obtained from the MHCPEP, and thus any bias in the data is a result of the choice of sequences by experimenters, and not by the author of this paper. From the table, it can be seen that there were no peptides with NO binding ability, and that the total number of binding peptides exceeds that of activating peptides in all the MHC molecules in the table.

To represent the peptide data obtained, we use a fully connected HMM, in which there is one starting state and one finishing state and any transition between two states (except that between the starting and finishing states and between the finishing state and any others) is allowed. In training the HMM, our supervised learning algorithm needs to attach the target probability to each peptide in our data. When the parameters of the fully connected HMM of N states are assumed to be uniform distributions [i.e., $a_{ij} = (1/N)$ $(i, j = 1, \ldots, N)$ and $b_j(c) = 1/M$ $(j = 1, \ldots, M)$], the probability that a sequence of length l is generated by the HMM is given as $1/N$ $(1/M)^l$ from Eq. (2), where M is the number of symbol types. In consideration of this calculation and based on a preliminary experiment, we fixed the target probability of a given peptide of length l to be $L^{0.05}$ if the peptide is in the YH class, $L^{0.1}$ if it is YM, $L^{0.2}$ if it is YL, and $L^{2.0}$ if it is NO, where $L = 1/N\,(1/M)^l$.

Although the length of the peptides obtained shown in Table I ranges from 7 to 25, most of the peptides are 9–13 residues long. Actually, in the binding peptides relevant to HLA-A2, 434 peptides (91.9% of the total) are 9–13 residues long; in particular, the number of nonamer peptides is 192, and they occupy 61.9% of the total. However, the peptides relevant to HLA-DR1 and HLA-DR4 are relatively longer than those relevant to HLA-A2 molecules. The most frequent length of the former peptides is 13, and such peptides account for 37.2% (153 out of 411) and 51.2% (266 out of 520) of all peptides related to HLA-DR1 and HLA-DR4, respectively. Thus, we fix the number of states of HMMs at 22 for HLA-A2, and 32 for HLA-DR1 and HLA-DR4. (Note that the number includes a starting state and a finishing state, neither of which generates any symbols.) As the number of states is set at roughly twice the length of most peptides in training examples, the HMMs are expected to extract separately multiple sequence patterns hidden in the data.

Comparing Our Supervised Learning of HMMs With Other Methods in a Type of Cross-Validation of Discriminating Sequences

We compare the performance of our supervised learning algorithm of an HMM with those of two other methods. The first of the two is a backpropagation neural network, which has been used in predicting MHC binding peptides and is regarded as the most effective approach. The second is the Baum-Welch algorithm, i.e., the most popular learning algorithm of an HMM. Using the Baum-Welch, we tested two types of HMMs, i.e., fully connected and alignment HMMs. A fully connected HMM is one in which any pair of states is connected, except for the pair of the starting and finishing states. On the other hand, an alignment HMM is a type of left-to-right HMM and was proposed for aligning multiple sequences and presenting their sequence profiles (Krogh et al., 1994).

Data

In this experiment, testing is done by binary prediction, i.e., YES or NO. Thus, we use activating peptides relevant to HLA-A2, because there are no non-binding data in any of the three MHC molecules used, and the amount of activity data of other MHC molecules is too small to be used as training data here. (Basically, there is no non-binding peptide in the MHC database, because it gathers peptides that bind to MHC proteins. Thus, among the peptides in the database, even those that cannot activate T-cell proliferation will bind to MHC proteins. In other words, such peptides can be regarded as false-positive data if peptides that can activate T-cell proliferation are called positive data. In this sense, the discrimination experiment performed here is a severe test.)

As backpropagation neural networks are used in the comparisons, all peptides used here are of nine residues, since all previous work based on backpropagation neural networks uses only nonamer peptides. The predictive performance must be measured by discriminating whether a given unknown peptide has a certain ability or not. Thus, we use the activating peptides of HLA-A2, for which Table Ia shows the number of peptides used in this experiment.

Note that the three methods differ in data usage. Our method uses four types of targets, i.e., YH, YM, YL, and NO. On the other hand, backpropagation neural networks are trained by two types of target values, i.e., YES or NO, as done in Gulukota et al. (1997), and the Baum-Welch algorithm uses only the target value of YES (YH, YM, YL) since it is an unsupervised learning algorithm.

Backpropagation neural network

We here briefly explain the network used in our experiment, which is the same as the one used by Gulukota et al. (1997).

The network has three layers, i.e., an input, a hidden, and an output layer, each of which consists of a fixed number of nodes. The numbers of input, hidden, and output nodes are $180(= 20 \times 9)$, 50, and 1, respectively. A set of 20 nodes in the input layer, each of which corresponds to one of 20 types of amino acids, corresponds to one of nine residues in a given peptide. When a peptide is given, only one node in the set of 20 nodes outputs 1 and the other 19 nodes in the set output 0. Any two nodes between input and hidden layers and between hidden and output layers are connected by a one-directional edge, to which a weight is attached.

Let the output value of the j-th node be x_j and the weight attached to the edge connecting from the i-th node to the j-th node be w_{ij}. We calculate the x_j in the hidden and output layers as follows:

$$x_j = f\left(\sum_i w_{ij} x_i\right), \tag{6}$$

where the function f is a sigmoid function satisfying $f(x) = \dfrac{1}{1 + e^{-x}}$. Weights w_{ij} are trained by the general backpropagation learning algorithm (Rumelhart, et al., 1986). In this learning, 1 is given as a teaching signal for the output value of this network if a given training peptide is a positive example; otherwise 0 is given, and the backpropagation minimizes the squared error-loss at the output node by a gradient descent algorithm. In prediction, when a new peptide is given, the output value of the output node, which can be calculated from Eq. (6), is given to the peptide.

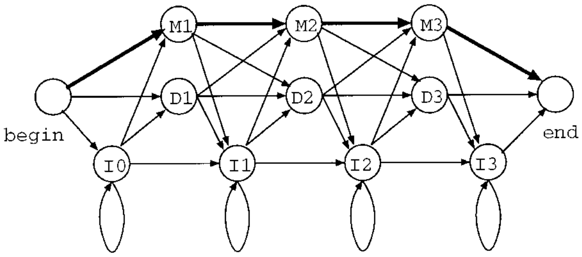

Fig. 1. Alignment hidden Markov model.

Alignment HMM

Figure 1 shows the structure of an alignment HMM, which has a particular structure consisting of three types of states, i.e., matching (M1, M2, M3 in Fig. 1), insertion (I0, I1, I2, I3 in the figure), and deletion (D1, D2, D3 in the figure) states. In the HMM, a matching state is a normal state that emits a symbol according to a probability distribution attached to the state, whereas an insertion state emits a symbol according to a fixed uniform distribution and a deletion state does not emit any symbol [see Krogh et al. (1994), Baldi et al. (1994), or Eddy (1996) for the details of the HMM].

In our experiment, the number of matching states is fixed at 20, which is the same as the number of states in the fully connected HMM used in the experiment, except for its starting and finishing states. The number of deletion and insertion states (20 and 21, respectively) is automatically determined from the number of matching states.

Baum-Welch algorithm

The Baum-Welch reestimation rules for a_{ij} and $b_j(c)$ are easily derived from the two probabilities γ and ξ in Eqs. (3) and (4) as follows:

$$\hat{a}_{ij} = \frac{\sum_{s} \sum_{t=1}^{l_s} \xi_t^s(i, j)}{\sum_{s} \sum_{t=1}^{l_s} \gamma_t^s(i)}$$

$$\hat{b}_i(c) = \frac{\sum_{s} \sum_{t=1, O_t^s = c}^{l_s} \gamma_t^s(i)}{\sum_{s} \sum_{t=1}^{l_s} \gamma_t^s(i)}$$

Experimental procedure

We randomly divide each set of peptides of four classes into two, i.e., training and test, with a certain proportion of training data to all obtained data, and repeat this random division five times, that is, we generate five random sets of training and test data for a given proportion.

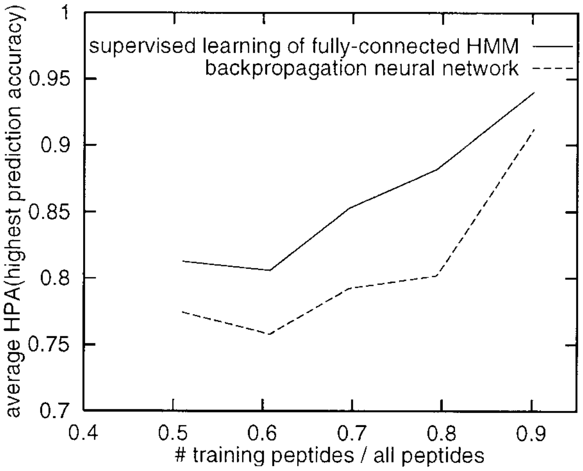

Fig. 2. Comparing our supervised learning of hidden Markov models with backpropagation neural networks.

In training, we randomly generate five HMMs (or backpropagation neural networks) having different initial parameter values. For each of the five, we repeatedly train it and use it to predict unknown test data five times, with the five respective random sets of training and test data already generated. Thus, a total of 25 trials are done at a given proportion of training data to all data. We vary the proportion of training data to the whole data from 50% to 90% at 10% intervals.

In testing, we measure the performance of each method by binary prediction, i.e., predicting whether a given peptide belongs to any of YH, YM, and YL (i.e., YES) or NO. In this prediction, we consider the highest prediction accuracy (hereafter, termed *HPA*) for test data that can be obtained by changing a cut-off value (which classifies test examples into two classes, i.e., YES and NO) for the output values of the test peptides. We calculated 25 HPAs for all 25 trials, and the performance of our method is evaluated by their average.

Learning curves

Figure 2 shows the learning curves of our supervised learning algorithm of HMMs and of backpropagation neural networks. As shown in the figure, the average HPA of the former is approximately 5–10% better than that of the latter at any proportion of training data to all data. From this result, we can say that in discriminating given new peptides, the performance of our supervised learning of HMMs exceeds that of a backpropagation neural network, which so far has been regarded as the most effective approach to this problem.

Parameter updating in learning each of HMMs and neural networks is repeated until the changes in their parameters become smaller than a certain preset amount. Figure 3 shows an actual example of

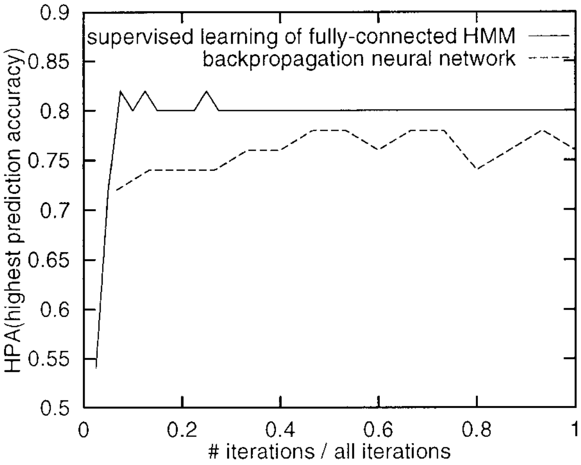

Fig. 3. Variation of HPA as number of iterations increases. The number of iterations is represented as the ratio to the number of all iterations obtained when a preset stopping condition is satisfied.

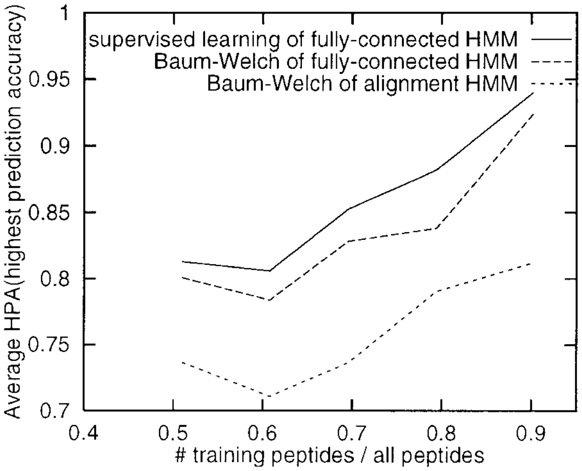

Fig. 4. Comparing our supervised learning with Baum-Welch.

the variation of HPAs of test data until the parameter updating is stopped. In the example, the two learning methods use an identical set of training and test data, at which the size of the training data is approximately the same as that of the test data. The figure shows that the HPA of each method reaches its upper limit before the parameter updating is stopped and that the HPA of an HMM is always better than that of a backpropagation neural network, despite minor training problems such as overtraining.

Figure 4 shows the learning curves of our supervised learning algorithm of fully connected HMMs and the Baum-Welch algorithm of fully connected or alignment HMMs. This figure indicates that the fully connected HMM is able to improve greatly the average HPA obtained by the alignment HMM, and that our supervised learning can further improve the HPA obtained by the Baum-Welch. The average HPA of fully connected HMMs trained by our method is always approximately 2–15% better than those of fully connected and alignment HMMs trained by the Baum-Welch.

Figures 2–4 clearly demonstrate that our method surpasses all the methods used for comparison purposes.

Predicting Peptides That Bind to MHC Molecules

Data

We focus on HLA-A2 protein, but we also attempt to use peptides that can bind to HLA-DR1 and HLA-DR4. The number of peptides relevant to activity is considerably smaller than that for binding to any MHC protein, and hence we here consider binding peptides only. Thus, the data used here have only three types of labels, i.e., YH, YM, and YL, and constitute a set of positive examples, which can bind to MHC molecules.

Experimental procedure

We train an HMM by our supervised learning algorithm using all data of an MHC molecule, and we repeat this training 100 times with random different initial parameter values. Out of the 100 trained HMMs, we choose the one that provides the minimum value of function *E* [see Eq. (5)] for all training data of peptides that can bind to an MHC molecule.

Next, we perform a random walk on the chosen HMM trained by peptides that bind to HLA-A2 protein. We start at the starting state of the HMM and randomly choose a state to transit depending on the transition probabilities attached to the edges from the starting state; after moving to a state, we again randomly choose a symbol depending on the symbol generation probability distribution attached to the state. We repeat this state transition and symbol generation until the transition reaches the finishing state. This random walk finally generates a string (symbol sequence) and the score of the sequence, which is obtained by multiplying all the probabilities used for generating the sequence on state transition and symbol generation of the walk. Roughly speaking, we can regard the score as the probability of the sequence given the model, as the Viterbi algorithm is used in predicting the probability.

We repeat the random walks 100,000 times, and out of the 100,000 sequences generated, we remove those that have already been noted in the MHCPEP database and those that are composed of only one type of amino acid. Out of the sequences generated, we extract only the sequences that are nine residues long, because, as mentioned earlier, such peptides of nine residues occupy more than 60% of all the peptides relevant to HLA-A2 and thus we expect that most experimental researchers are interested in the nonamer peptides. We sort the processed se-

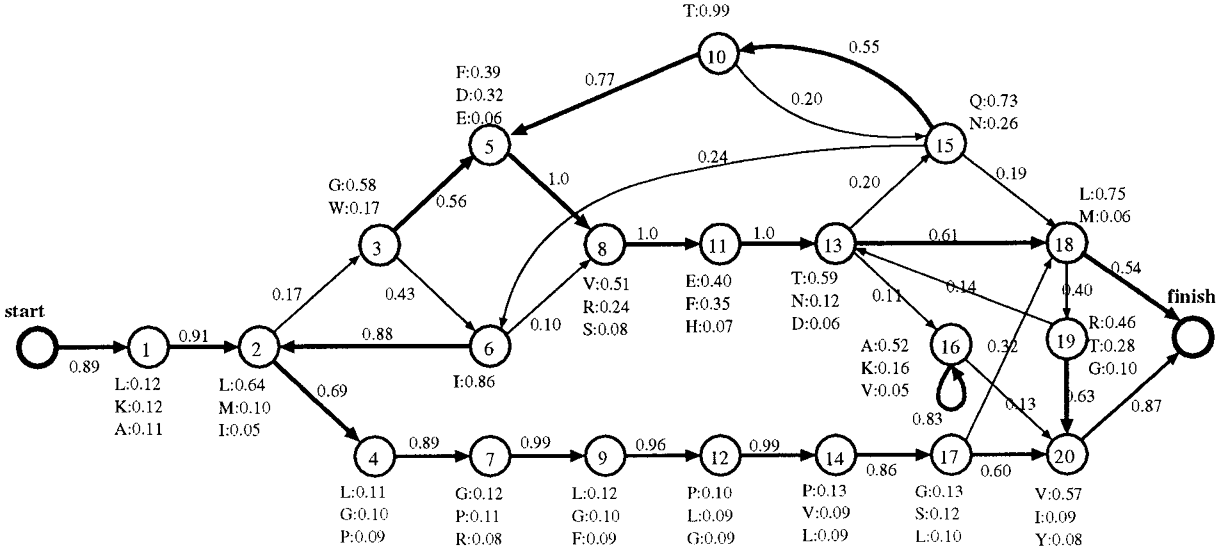

Fig. 5. Main part of HMM representing peptides binding to HLA-A2. Only edges whose transition probabilities exceed 0.1 and the top three symbols (at maximum), whose symbol generation probabilities exceed 0.05 at each state, are shown. The edge having the largest transition probability of the probabilities attached to edges starting from a state is shown by a thick line, and states are numbered 1 to 20 from left to right and from top to bottom.

quences in descending order of scores and select 10,000 of them from the top down.

Finally, we repeat the above process five times. We sort the five sequences of 10,000 obtained in descending order of scores and select the top 100.

HLA-A2

Figure 5 shows the HMM that provides the lowest value of function *E* for all training peptides that bind to HLA-A2 protein, in 100 HMMs trained by our supervised learning algorithm using the same data.

Note that the HMM automatically extracts roughly two different patterns hidden in the peptides used as training data. One major pattern is states 1 → 2 → 4 → 7 → 9 → 12 → 14 → 17 → 20, and the other relatively minor pattern is states 1 → 2 → 3 → 5 → 8 → 11 → 13 → [(18(→ 19 → 20)) or (16 → 20)]. As shown in Figure 5, a number of variations can be incorporated in the second pattern, but no change is allowed in the first pattern except for the last state.

States 2 and 20, which can be used in the two patterns, coincide with two anchor residues of a nonamer HLA-A2-restricted motif reported by Falk et al. (1991). In the motif, anchors are Leu at position 2 and Val at position 9, and these have high probabilities at states 2 and 20, respectively.

In the first pattern, all states except for the two states corresponding to anchor residues have broad symbol generation probability distributions, in which the largest probability value is at maximum 0.13. Furthermore, such distributions at states 4, 7, 9, 12, and 14 are similar to each other, and in them, Gly, Leu, and Pro always have relatively high probabilities. This result is consistent with a report by Sette et al. (1991), in which positions 3–5 in a nonamer HLA-A2 motif have the same amino acid propensity. This indicates, however, that neither the motif nor the first pattern can capture any distinct feature of this portion, and thus it will be difficult for them to predict accurately (or discriminate) peptides that bind to HLA-A2 protein.

On the other hand, the second pattern presents a clearer sequence pattern hidden in the training data. In particular, the transition of states 5 → 8 → 11 → 13 is connected by edges, at any of which a transition probability of 1.0 is attached, and this indicates that the transition is *certainly* hidden in training peptide data. Actually, an epitope found in influenza matrix protein (Gotch et al., 1988) contains the amino acid sequence Phe-Val-Phe-Thr, which can be generated with a high probability by this transition. The sequence is found in 49 of the total 472 peptides used as training data, and this is one of the most frequent patterns in HLA-A2 binding peptides. Note that the sequence Phe-Val-Phe-Thr is found in a different position in each of the 49 training peptides. Out of the 49 peptides, the numbers in which the sequence starts at the fourth, fifth, sixth, seventh, and eighth positions are 3, 23, 20, 2, and 1, respectively.

We can find other frequent sequence patterns in the second. For example, the longer sequence Leu-Gly-Phe-Val-Phe-Thr, which can be generated by states 2 → 3 → 5 → 8 → 11 → 13 with a high probability, is found in 36 peptides in the training data. Similarly, the sequence Thr-Leu-Thr-Val, which can be generated by states 13 → 18 → 19 → 20, is in 33 peptides in training data, and Ala-Ala-Ala, i.e., $(Ala)_3$, generated by state 16 only, is found in 38 of

TABLE II. Patterns That Can Be Generated by HMMs of HLA-A2, HLA-DR1, and HLA-DR4 With High Probabilities and That Are Frequently Seen in Peptides of Training Data[†]

Pattern	State transition	No. of peptides
a. HLA-A2		
GILGF	3 → 6 → 2 → 3 → 5	33
ILGFVF	6 → 2 → 3 → 5 → 8 → 11	34
LGFVFT	2 → 3 → 5 → 8 → 11 → 13	36
GFVFTL	3 → 5 → 8 → 11 → 13 → 18	36
FTLTV	11 → 13 → 18 → 19 → 20	30
b. HLA-DR1		
AAAAA	2 → 2 → 2 → 2 → 2	39
PKYVKQN	3 → 5 → 6 → 24 → 18 → 20 → 22	34
KYVKQNT	5 → 6 → 24 → 18 → 20 → 22 → 25	33
YVKQNTL	6 → 24 → 18 → 20 → 22 → 25 → 26	34
VKQNTLK	24 → 18 → 20 → 22 → 25 → 26 → 27	31
KQNTLKL	18 → 20 → 22 → 25 → 26 → 27 → 28	31
QNTLKLA	20 → 22 → 25 → 26 → 27 → 28 → 29	32
NTLKLAT	22 → 25 → 26 → 27 → 28 → 29 → 25	33
QYIKANS	3 → 5 → 8 → 5 → 9 → 12 → 15	31
YIKANSK	5 → 8 → 5 → 9 → 12 → 15 → 18	31
IKANSKF	8 → 5 → 9 → 12 → 15 → 18 → 20	31
KANSKFI	5 → 9 → 12 → 15 → 18 → 20 → 8	30
NSKFIG	12 → 15 → 18 → 20 → 8 → 11	32
SKFIGI	15 → 18 → 20 → 8 → 11 → 14	32
FIGITE	20 → 8 → 11 → 14 → 29 → 25	30
EKASSVF	3 → 5 → 9 → 12 → 15 → 17 → 20	32
KASSVFN	5 → 9 → 12 → 15 → 17 → 20 → 22	31
ASSVFNV	9 → 12 → 15 → 17 → 20 → 22 → 17	31
SSVFNVV	12 → 15 → 17 → 20 → 22 → 17 → 24	31
EKKIA	3 → 5 → 7 → 14 → 29	32
KKIAKM	5 → 7 → 14 → 29 → 18 → 21	30
KIAKME	7 → 14 → 29 → 18 → 21 → 3	30
IAKMEKA	14 → 29 → 18 → 21 → 3 → 5 → 9	30
AKMEKAS	29 → 18 → 21 → 3 → 5 → 9 → 15	30
c. HLA-DR4		
AAAAAA	1(3) → 1(3) → 1(3) → 1(3) → 1(3) → 1(3,5)	39
AAYAAA	1(3) → 1(3) → 1 → 1(3) → 1(3) → 1(3,5)	43
AAAKAAA	1(3) → 1(3) → 1(3) → 1 → 1(3) → 1(3) → 1(3,5)	37
KAAAAAA	1(3) → 1(3) → 1(3) → 1(3) → 1(3) → 1(3) → 1(3,5)	32
KYVKQNTL	4 → 6 → 9 → 11 → 15 → 17 → 18 → 19	34
YVKQNTLK	6 → 9 → 11 → 15 → 17 → 18 → 19 → 11	35
VKQNTLKL	9 → 11 → 15 → 17 → 18 → 19 → 11 → 15	33
KQNTLKLA	11 → 15 → 17 → 18 → 19 → 11 → 15 → 17	34
QNTLKLAT	15 → 17 → 18 → 19 → 11 → 15 → 17 → 18	34

[†]The patterns presented here are those that are longer than three and are found in more than 30 peptides in the respective training data. If longer patterns, including those that satisfy the above requirement, are found in more than 30 sequences, only the longest one of them is described.

training data. All frequent sequence patterns, which are revealed by the HMM of Figure 5, are shown in Table IIa. The table indicates that each portion of the second pattern in the HMM captures hidden features in the training data.

Table III shows the top 100 peptides obtained by our random generation process described above and whose binding ability is so far unknown. Most of the 100 peptides obtained have the pattern that is supposed to be generated by states 5 → 8 → 11 → 13, and this indicates that most of the peptides generated belong to the second pattern presented by the HMM of Figure 5.

HLA-DR1 and HLA-DR4

Figure 6 shows the HMM that provides the lowest *E* value for all training peptides that bind to HLA-DR1 protein. Note that the figure shows only 31 states, and the remaining state is not used to represent peptides that bind to HLA-DR1.

A nonamer motif of HLA-DR1 binding peptides presented by Hammer et al. (1992, 1993) has Tyr or Phe at position 1 and Leu or Met at position 4. We guess that this motif appears in our HMM by regarding states 6 and 16 as positions 1 and 4 in the motif, respectively. However, a state transition including

TABLE III. Top 100 Peptides of Nine Residues That Bind to HLA-A2 Protein, Generated by HMM of Figure 5

	Peptide	Probability		Peptide	Probability
1	GILGFVETL	−5.13036	51	AGFRETLRV	−5.86142
2	GILGDVETL	−5.21152	52	GGFVETLTV	−5.86222
3	GILGDVFTL	−5.26656	53	KLGFVFTLL	−5.86632
4	LGFVETLRV	−5.47421	54	LLGDVFTQL	−5.86781
5	KGFVETLRV	−5.47863	55	GILGFVFSL	−5.87013
6	GILGFRFTL	−5.50243	56	GFVFTLLRV	−5.87366
7	GILGDRETL	−5.52856	57	GDVFTQLRV	−5.87957
8	KGFVFTLRV	−5.53366	58	FGDVFTLRV	−5.88431
9	LGDVETLRV	−5.55537	59	LLGDVETLL	−5.88803
10	KGDVETLRV	−5.55979	60	LLGFVFTRV	−5.88970
11	GILGDRFTL	−5.58359	61	GILGDVESL	−5.89625
12	GFVETLRTL	−5.58546	62	AGDVFTLTV	−5.89778
13	LGDVFTLRV	−5.61040	63	LLLGFVETL	−5.89835
14	KGDVFTLRV	−5.61482	64	GDVETLLRV	−5.89978
15	GFVFTLRTL	−5.64049	65	GFRETLRTL	−5.90249
16	GGFVETLRV	−5.64502	66	ALGFVETRV	−5.90484
17	AGDVFTLRV	−5.68058	67	IGDVFTLRV	−5.90625
18	LGFVETLTV	−5.69142	68	GILGDVENL	−5.90660
19	KGFVETLTV	−5.69584	69	GGFVFTLTV	−5.91726
20	GGFVFTLRV	−5.70005	70	KLGDVETRV	−5.92025
21	GILWFVFTL	−5.71087	71	GDVETQTQL	−5.94344
22	GDVFTLRTL	−5.72165	72	RGFVETLRV	−5.94488
23	GGDVETLRV	−5.72618	73	GILAAAAAV	−5.94896
24	KLGFVETQL	−5.73604	74	GIMGFVETL	−5.95025
25	GILWDVETL	−5.73699	75	TGDVETLRV	−5.95214
26	WILGDVETL	−5.73699	76	GLGFVFTQL	−5.95746
27	GFVETQLRV	−5.74338	77	ALGDVETLL	−5.95820
28	FGFVETLRV	−5.74811	78	GFVETQLTV	−5.96058
29	KGFVFTLTV	−5.75087	79	GILGDVFNL	−5.96164
30	AGFVETLTV	−5.76159	80	GGFRETLRV	−5.96205
31	IGFVETLRV	−5.77005	81	FGFVETLTV	−5.96532
32	LGDVETLTV	−5.77257	82	LLGDVFTRV	−5.97086
33	GGDVFTLRV	−5.78121	83	KLGDVFTRV	−5.97528
34	KLGFVFTQL	−5.79107	84	GILGEVETL	−5.97586
35	LGFRETLRV	−5.79124	85	GFVETLRVL	−5.97601
36	GILWDVFTL	−5.79203	86	GLGFVETLL	−5.97768
37	KGFRETLRV	−5.79566	87	LLLGDVETL	−5.97951
38	ALGFVETQL	−5.80179	88	YLAAAAAAV	−5.98183
39	FGFVFTLRV	−5.80315	89	GDRETLRTL	−5.98365
40	LLGFVETLL	−5.80687	90	KLLGDVETL	−5.98393
41	KLGFVETLL	−5.81129	91	ALGDVETRV	−5.98600
42	LLGDVETQL	−5.81278	92	GILGDVHTL	−5.99664
43	AGFVFTLTV	−5.81663	93	GGDVFTLTV	−5.99842
44	KLGDVETQL	−5.81720	94	GDVFTQTQL	−5.99848
45	GDVETQLRV	−5.82453	95	LWFVETLRV	−5.99968
46	LGDVFTLTV	−5.82761	96	RGFVFTLRV	−5.99992
47	KGDVFTLTV	−5.83203	97	KWFVETLRV	−6.00410
48	FLAAAAAAV	−5.85104	98	FLGFVETQL	−6.00552
49	IGDVETLRV	−5.85121	99	GFVFTQLTV	−6.01562
50	GFVFTLTTL	−5.85770	100	CLAAAAAAV	−6.02506

these two states, i.e., states 6 → 10 → 13 → 16, is merely a part of the entire HMM shown in Figure 5. Thus, we can say that the HMM finds other patterns that are different from the existing motif.

Actually, each portion of the HMM presents a sequence pattern that is frequently found in training data. For example, the sequence Glu-Lys-Ala-Ser, which can be generated by states 3 → 5 → 9 → 12 with a high probability, is contained in 40 peptides, the sequence Ser-Ser-Val-Phe-Asn, from states 12 → 15 → 17 → 20 → 22, is found in 34 peptides, and the longer sequence Asn-Thr-Leu-Lys-Leu-Ala, from states 22 → 25 → 26 → 27 → 28 → 29, is found in 38 peptides.

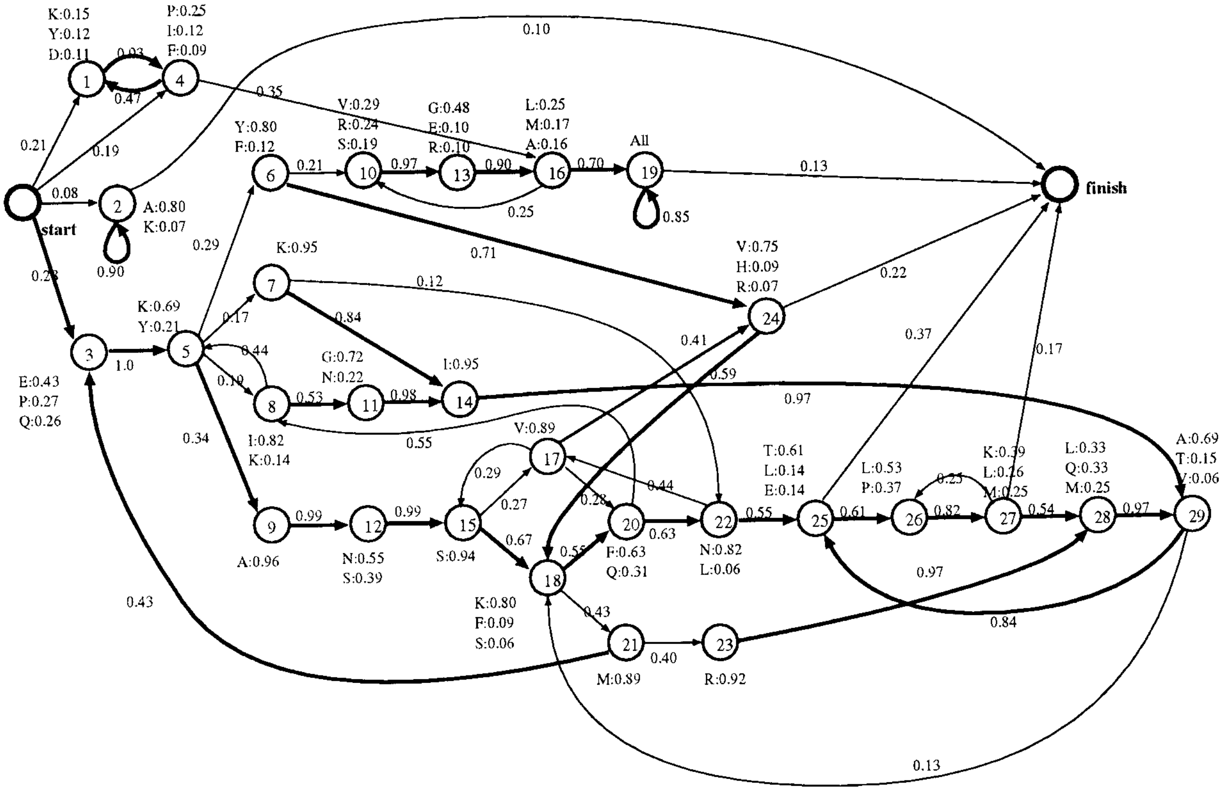

Fig. 6. Main part of HMM representing peptide binding to HLA-DR1. The thresholds used for omitting edges and symbols are the same as those used in Figure 5. We attach ''All'' to a state at which no symbol generation probability exceeds 0.1.

The strength of a dependency between two states in an HMM that are connected by an edge is represented by a transition probability attached to the edge, and this indicates that the HMM can capture a pair of two neighboring amino acids, i.e., a two-letter string, which is frequently seen in training data. This feature allows an HMM to represent these frequent sequence patterns by a sequence of states.

It is interesting to note that the combinations of the sequence patterns generated by HMMs with high probabilities are not necessarily found in the training data. For example, although a combination of the first and second examples described above, i.e., Glu-Lys-Ala-Ser-Ser-Val-Phe-Asn, is found in 28 peptides of the training data, a combination of the second and third, i.e., Ser-Ser-Val-Phe-Asn-Thr-Leu-Lys-Leu-Ala, cannot be found in the training data and even the center part of this combination, i.e., Phe-Asn-Thr, is not contained in the training data. This is because a state in an HMM depends only on the states from which edges with high transition probabilities connect to the state. Thus, there is no substantial dependency between two states in a sequence of states generating a string, if the two do not generate two neighboring amino acids in a frequent sequence. However, we can say that HMMs that allow us to extract frequent two-letter strings in training data provide *new combinations* of the strings that correspond to unknown peptides expected to bind to an MHC protein.

Figure 7 shows the HMM that provides the lowest *E* value for all training data of peptides that bind to HLA-DR4 protein. This figure also uses only 26 states to represent the peptides that bind to HLA-DR4, and roughly two patterns, i.e., states 1, 3 and 5 and others, are shown in the figure.

The first pattern is represented by only three states, any of which generates Ala with the highest probability among all 20 types of amino acids. This feature is similar to state 2 in Figure 6. Actually, $(\text{Ala})_n$ is one of the popular sequence patterns in training data for HLA-DR1 and HLA-DR4. A partial sequence $(\text{Ala})_5$ is found in 39 and 51 peptides in training data of HLA-DR1 and HLA-DR4, respectively, and $(\text{Ala})_6$ is also found in 29 and 39 peptides, respectively.

On the other hand, an HLA-DR4-restricted nonamer motif presented by Hammer et al. (1993) and Sette et al. (1993), is Leu, Ile, Val, Trp, or Tyr at position 1 and Thr at position 5. We can find the motif in the second pattern of Figure 7, if we regard

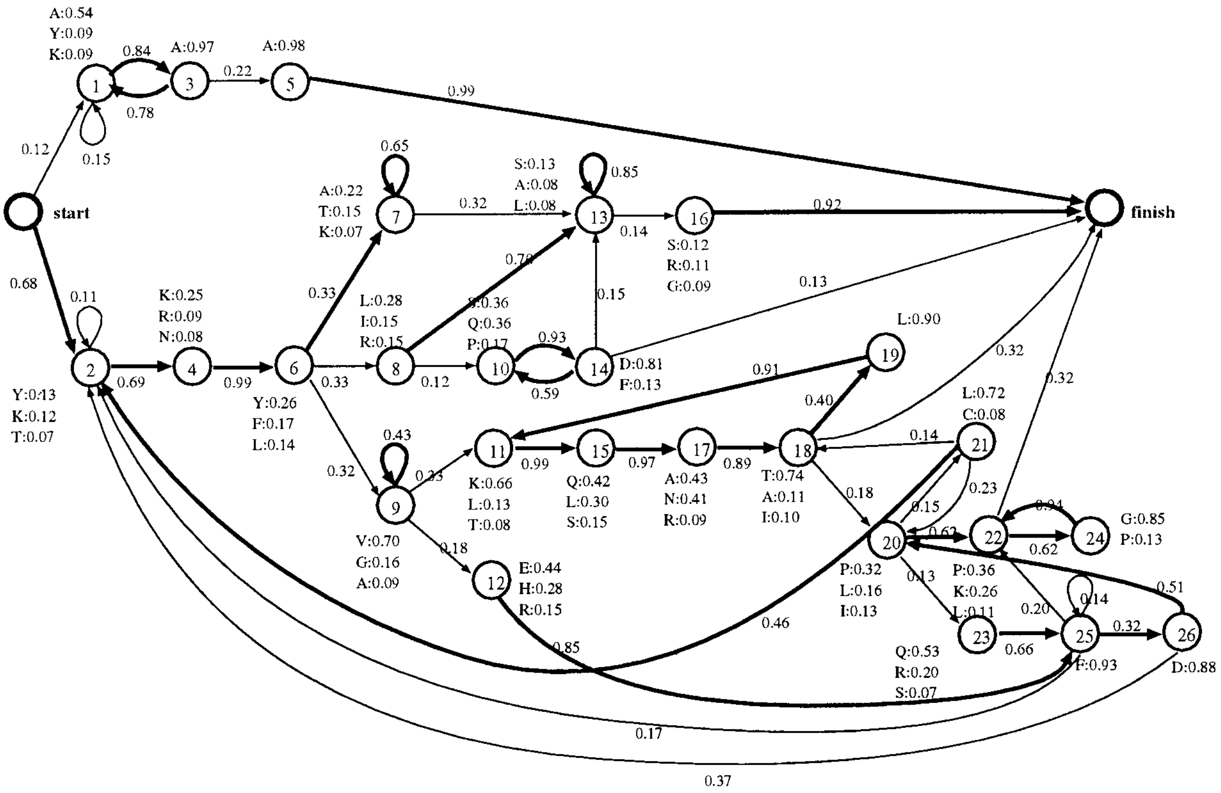

Fig. 7. Main part of HMM representing peptide binding to HLA-DR4. The thresholds used for omitting edges and symbols are the same as those used in Figure 5.

states 6 and 18 in the HMM as positions 1 and 5 in the motif.

Table IIb and c, respectively, show all frequent sequence patterns of length longer than three that are revealed by the HMMs of Figures 6 and 7.

DISCUSSION AND CONCLUSIONS

From our computer experiments in discriminating unknown sequences, we have shown that the performance of our supervised learning algorithm of a hidden Markov model (HMM) surpasses that obtained by either a backpropagation neural network, which up to now has been regarded as the most effective approach to predicting MHC binding peptides, or the Baum-Welch updates of an HMM. Furthermore, we trained a fully connected HMM by our supervised learning algorithm, and used it to predict new peptides that will bind to MHC molecules. We believe that the peptides predicted here and the HMMs trained by our supervised learning algorithm provide useful information for further research into the MHC molecules dealt with here.

Note that the patterns extracted and represented by the HMMs do not necessarily contain *all* patterns that can bind to the MHC molecules dealt with here. There are at least two reasons for this. First, the amount of data used here is extremely limited as well as biased, because existing peptide data have not been randomly experimentally investigated, even though the data used here are all derived from a currently available database. To put it concretely, some patterns that can actually bind to MHC molecules may not be contained in our training data, and thus the peptides predicted by the HMMs are limited to those having patterns similar to existing patterns having binding ability. Second, the structure and parameters of a trained HMM depend on its initial parameters, since our learning algorithm of an HMM is a local optimization algorithm. Thus, a trained HMM may not include a pattern hidden in a given set of data. To avoid this in our experiments as much as possible, we repeated the training of an HMM for a given set of training data, and we evaluated the performance by the average of the repetition or chose the best HMM from the results of the repetition, as the occasion demanded.

However, the more training data are obtained, the larger the number of sequence patterns HMMs will be able to extract. This means we will be able to use HMMs and our supervised learning algorithm to extract frequent sequence patterns even in other MHC proteins of which only an extremely small amount of data is known at present, if a larger number of peptides that bind to MHC molecules are investigated in the future.

Figures 2–4 indicate that a fully connected HMM achieved a higher discrimination accuracy than those of both a backpropagation neural network and an alignment HMM. One important reason for this result is derived from the characteristics of both the data we used and the structures of the models that learn the data.

Concretely speaking, when four-letter substrings of ABCDE, i.e., ABCD and BCDE, are given as positive training data, an HMM, especially a fully connected HMM, can extract the common portion of them (i.e., BCD) with ease, while it is rather difficult for backpropagation neural networks to do so with complete accuracy because B, C, and D are all located at different positions in the two strings. Actually, on the binding of an MHC to a peptide, it is known that a peptide longer than nine can be entirely contained in an MHC groove (e.g., Collins et al., 1994). The data dealt with in this paper are also expected to include such sequences, and, as mentioned earlier, the Phe-Val-Phe-Thr pattern begins at variable positions in HLA-A2 binding peptide data. Hence, this result implies that a model dealing with only a peptide of a fixed length will not be an effective approach to the problem of predicting peptides that bind to MHC proteins.

Furthermore, when other four-letter substrings of FGHI J, i.e., FGHI and GHI J, are added to the positive training data, a fully connected HMM can extract two common patterns of them (i.e., BCD and GHI) separately. However, it is rather difficult for an alignment HMM to capture them separately, and thus it will learn a mixture of the two patterns, because the structure of the HMM was proposed for the purpose of aligning a number of sequences as a single pattern (i.e., a sequence profile), as seen in Figure 1. Note that if only two strings, ABCD and FGHI, are given as positive training data, both a backpropagation neural network and a fully connected HMM will be able to learn the two patterns separately, while it will be difficult for an alignment HMM to do so for the same reason. Our experimental results indicate that actually each set of peptides contains *multiple* (more than one) patterns, as shown in Figures 5–7.

In view of these considerations, we can say that a fully connected HMM is the most suitable representation model among the three we compared. Furthermore, in the test, our supervised learning algorithm worked effectively for the data we used, in which each peptide sequence has its own target value.

In our experiment for obtaining unknown peptides having a high possibility of binding to HLA-A2, we randomly generated peptides using 1 HMM chosen out of 100 HMMs, each of which was trained by all available peptides that bind to HLA-A2, and we presented the top 100 by sorting the peptides. The peptides ranked below 100, the peptides that are longer or shorter than nine, and the peptides generated by the HMMs of Figures 6 and 7 in the same procedure as done for HLA-A2 can be obtained from the author.

Finally, we would like to emphasize that the trained HMMs can be used to find epitopes in a given new sequence, e.g., a human immunodeficiency virus protein, just as existing motifs are used for the same purpose (Meister et al., 1995). We believe that the trained HMMs are useful for this purpose as well.

ACKNOWLEDGMENTS

The author thanks Dr. Keiko Udaka of Kyoto University for introducing the problem of predicting peptides binding to MHC molecules. Thanks are also due to Dr. Naoki Abe of the NEC Corporation for helpful discussions and constant encouragement.

REFERENCES

Adams, H.P., Koziol, J.A. (1995). Prediction of binding to MHC class I molecules. J. Immunol. Methods 185:181–190, 1995.

Altuvia, Y., Schueler, O., Margalit, H. Ranking potential binding peptides to MHC molecules by a computational threading approach. J. Mol. Biol. 249:244–250, 1995.

Baldi, P., Chauvin, Y., Hunkapillar, T., McClure, M. Hidden Markov models of biological primary sequence information. Proc. Natl. Acad. Sci. USA 91:1059–1063, 1994.

Bisset, L.R., Fierz, W. Using a neural network to identify potential HLA-DR1 binding sites within proteins. J. Mol. Recognit. 6:41–48, 1993.

Bjorkman, P.J., Burmesister, W.P. Structures of two classes of MHC molecules elucidated: crucial differences and similarities. Curr. Opin. Struct. Biol. 4:852–856, 1995.

Bouvier, M., Wiley, D.C. Importance of peptide amino and carboxyl termini to the stability of MHC class I molecules. Science 265:398–402, 1994.

Bowie, J.U., Luthy, R., Eisenberg, D. A method to identify protein sequences that fold into a known three-dimensional structure. Science 253:164–170, 1991.

Brusic, V., Rudy, G., Harrison, L.C. Prediction of MHC binding peptides using artificial neural networks. In: “Complex Systems: Mechanism of Adaptation.” Stonier, R.J., Yu, X.S. (eds.). Amsterdam: IOS Press, 1994:253–260.

Brusic, V., Rudy, G., Kyne, A.P., Harrison, L.C. MHCPEP, a database of MHC-binding peptides: update 1996. Nucleic Acids Res. 25:269–271, 1997a.

Brusic, V., Schönbach, C., Takiguchi, M., Ciesielski, V., Harrison, L.C. Application of genetic search in derivation of matrix models of peptide binding to MHC molecules. In: “Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (ISMB-97).” Halkidiki, Greece: AAAI Press, 1997b:75–83.

Churchill, G.A. Stochastic models for heterogeneous DNA sequences. Bull. Math. Biol. 51:79–94, 1989.

Collins, E.J., Garboczi, D.N., Wiley, D.C. Three-dimensional structure of a peptide extending from one end of a class I MHC binding site. Nature 371:626–629, 1994.

Davenport, M.P., Shon, I.A.P.H., Hill, A.V.S. An empirical method for the prediction of T-cell epitopes. Immunogenetics 42:392–397, 1995.

Eddy, S.R. Hidden Markov models. Curr. Opin. Struct. Biol. 6:361–365, 1996.

Falk, K., Rötzschke, O., Stevanović, S., Jung, G., Rammensee, H.-G. Allele-specific motifs revealed by sequencing of self-peptides eluted from MHC molecules. Nature 351:290–296, 1991.

Gotch, F., McMichael, A., Rothbard, J. Recognition of influenza A matrix protein by HLA-A2-restricted cytotoxic T lymphocytes. Use of analogues to orientate the matrix peptide in the HLA-A2 binding site. J. Exp. Med. 168:2045–2057, 1988.

Gulukota, K., Sidney, J., Sette, A., DeLisi, C. Two complementary methods for predicting peptides binding major histocompatibility complex molecules. J. Mol. Biol. 267:1258–1267, 1997.

Hammer, J., Takacs, B., Sinigaglia, F. Identification of a motif for HLA-DR1 binding peptides using M13 display libraries. J. Exp. Med. 176:1007–1013, 1992.

Hammer, J., Valsasnini, P., Tolba, K., Bolin, D., Higelin, J., Takacs, B. Promiscucous and allele-specific anchors in HLA-DR-binding peptides. Cell 74:197–203, 1993.

Kondo, A., Sidney, J., Southwood, S. Prominent roles of secondary anchor residues in peptide binding to HLA-A24 human class molecules. J. Immunol. 155:4307–4312, 1995.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D. Hidden Markov models in computational biology. Applications to protein modeling. J. Mol. Biol. 235:1501–1531, 1994.

Lee, K.F. "Automatic Speech Recognition: The Development of the SPHINX System." Boston, MA: Kluwer Academic Publishers, 1989.

Mamitsuka, H. A learning method of hidden Markov models for sequence discrimination. J. Comput. Biol. 3:361–373, 1996.

Mamitsuka, H. Supervised learning of hidden Markov models for sequence discrimination. In: "Proceedings of the First International Conference on Computational Molecular Biology (RECOMB-97)." Santa Fe, NM: ACM Press, 1997:202–209.

Margalit, H., Spouge, J.L., Cornette, J.L., Cease, K.B., DeLisi, C., Berzofsky, J.A. (1987). Prediction of immunodominant helper T cell antigenic sites from the primary sequence. J. Immunol. 138:2213–2229, 1987.

Matsumura, M., Fremont, D.H., Peterson, P.A., Wilson, I.A. Emerging principles for the recognition of peptide antigens by MHC class I molecules. Science 257:927–934, 1992.

Meister, G.E., Roberts, C.G.P., Berzofsky, J.A., Groot, A.S.D. Two novel T cell epitope prediction algorithms based on MHC-binding motifs; comparison of predicted and published epitopes from *Mycobacterium tuberculosis* and HIV protein sequences. *Vaccine* 13:581–591, 1995.

Parker, K.C., Bednarek, M.A., Coligan, J.E. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. J. Immunol. 152: 163–175, 1994.

Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77:257–286, 1989.

Rammensee, H.-G., Falk, K., Rotzschke, O. Peptides naturally presented by MHC class I molecules. Annu. Rev. Immunol. 11:213–244, 1993.

Rammensee, H.G., Friede, T., Stenovic, S. MHC ligands and peptide motifs: first listing. Immunogenetics 41:178–228, 1995.

Rothland, J.B., Taylor, W.R. A sequence pattern common to T cell epitopes. EMBO J. 7:93–100, 1988.

Rumelhart, D., Hinton, G., Williams, R. Learning representations by backpropagating errors. Nature 323:533–536, 1986.

Ruppert, J., Sidney, J., Celis, E., Kubo, R.T., Grey, H.M., Sette, A. Prominent role of secondary anchor residues in peptide binding to HLA-A2.1 molecules. Cell 74:929–937, 1993.

Sette, A., Sidney, J., Oseroff, C., del Guercio, M.F., Southwood, S., Arrhenius, T. HLA-DR4w4-binding motifs illustrate the biochemical basis of degeneracy and specificity in peptide-DR interaction. J. Immunol. 151:3163–3170, 1993.

Sette, A., Vitiello, A., Farness, P., et al. Random association between the peptide repertoire of A2.1 class I and several different DR class II molecules. J. Immunol. 147:3893–3900, 1991.

Sippl, M.J. Calculation of conformational ensembles from potentials of mean force, an approach to the knowledge-based prediction of local structures in globular proteins. J. Mol. Biol. 213:859–883, 1990.