Recent Applications of Hidden Markov Models in Computational Biology

Khar Heng Choo[1], Joo Chuan Tong[1], and Louxin Zhang[2]*

[1] *Department of Biochemistry, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260;*
[2] *Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543.*

This paper examines recent developments and applications of Hidden Markov Models (HMMs) to various problems in computational biology, including multiple sequence alignment, homology detection, protein sequences classification, and genomic annotation.

Key words: Hidden Markov Models, sequence alignment, homology detection, protein structure prediction, gene prediction

Introduction

Hidden Markov Models (HMMs), being computationally straightforward underpinned by powerful mathematical formalism, provide a good statistical framework for solving a wide range of time-series problems, and have been successfully applied to pattern recognition and classification for almost thirty years.

The study of Markov Chains (MCs) was initiated in early 1900s by Markov (*1*), who laid the foundation for the theory of stochastic processes. From 1940s to 1960s, HMMs had been investigated as a representation of stochastic functions of MCs (*2–5*). Its initial development was predominated by theoretical reasonings that attempt to solve problems pertaining to the issues of uniqueness and identifiability. HMMs did not gain much popularity until early 1970s when Baum *et al* successfully applied the technique to speech recognition by developing an efficient training algorithm for HMMs (*6*).

In the late 1980s and early 1990s, HMMs were subsequently introduced to computational sequence analysis (*7*) and protein structural modeling (*8*, *9*) in molecular biology. However, HMMs have gained their popularity in the computational biology community only after three groups explored HMM-based profile methods for sequence alignment (*10–12*). In his excellent survey papers, Eddy addressed what HMMs are, their strength and limitation, and how profile HMMs were beginning to be used in protein structural modeling and sequence analysis (*13*, *14*). Our article emphasizes on recent HMM applications appearing in computational biology in the last five years since the last review of the field (*14*).

*** Corresponding author.**
E-mail: matzlx@nus.edu.sg

Hidden Markov Model

A wonderful description of the HMM theory has been written by Rabiner (*15*). In a nutshell, HMMs are composed of two components. Associated with each HMM is a discrete-state, time-homologous, first-order MC with suitable transition probabilities between states and an initial distribution. In addition, each state emits symbols according to a pre-specified probability distribution over emission symbols or values. Emission probabilities are dependent only on the present state of the MC, regardless of previous states. Starting from some initial states with the initial probability, a sequence of states is generated by moving from one state to another according to the state-transition probabilities until a final state is reached, creating an observable sequence of symbols as each state emits a symbol when it is visited.

The key idea is that an HMM is a sequence "generator". It is a finite model describing a probability distribution over a set of possible sequences. A simple HMM for generating a DNA sequence is specified in Figure 1A.

In the model, state transitions and their associated probabilities are indicated by arrows; and symbol emission probabilities for A, C, G, T at each state are indicated below the state. For clarity, we omit the

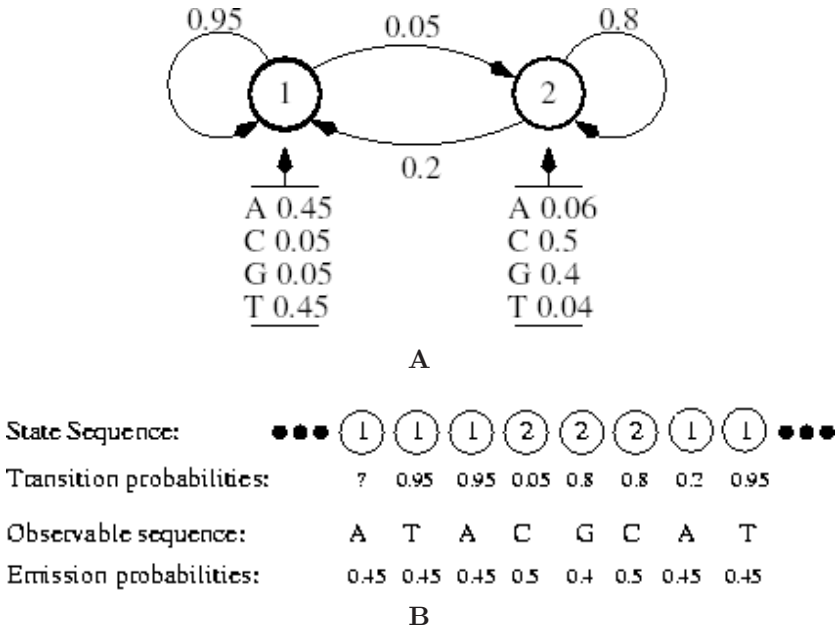


Fig. 1 A. a simple HMM model for generating DNA sequences; **B.** a generated state sequence and the associated DNA sequence

initial and final states as well as the initial probability distribution. For instance, this model can generate the state sequence given in Figure 1B and each state emits a nucleotide according to the emission probability distribution.

When producing sequences of emissions, only the output symbols can be observed. The sequences of states underlying MC are hidden and cannot be observed, hence the name Hidden Markov Model. Any sequence can be represented by a state sequence in the model. The probability of any sequence, given the model, is computed by multiplying the emission and transition probabilities along the path.

HMM topologies

The topology of an HMM refers to the set of states, and in particular the permitted and prohibited transitions between the states of the underlying MC, that is, the respective non-zero and zero entries of the transition matrix. To date, many different HMM topologies have been proposed, which include the fully connected model, circular model and left-right model.

Fully connected model

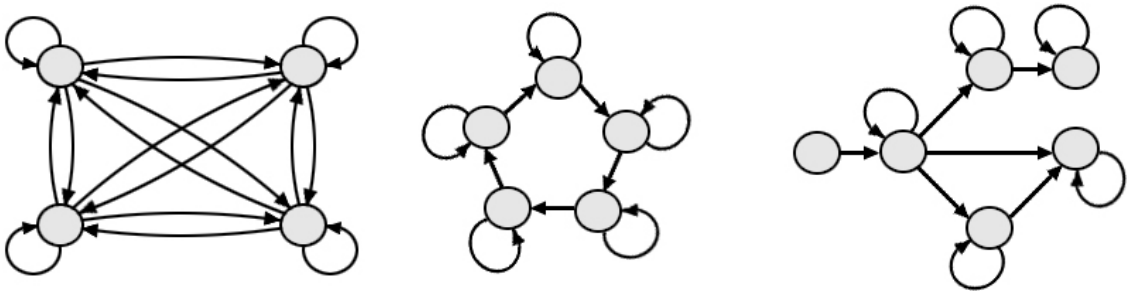
An HMM is termed a fully connected model (Figure 2A) when the states are pairwise connected such that the underlying digraph is complete. There are no distinguishable starting and terminating states and the transition matrix does not contain any zero entries with the exception of diagonal entries that correspond to loops or self-transitions.

Circular model

In a circular model (Figure 2B), the underlying directed graph is ergodic where the probability that any state will recur with the exception of states with zero probability. It is insensitive to size changes and there are no unique starting and terminating states.

Left-right model

When the underlying directed graph is acyclic, with the exception of loops, hence supporting a partial order of the states, it is known as left-right model (Figure 2C). In principle, there is one start state and one end state, which can be attained through the use of a special symbol for the end of an observation sequence and silent states (states with no output). Transitions from state to state proceed from left to right through the model, with the exception of loops. A more stringent form of this topology is defined by the strict left-right model that forbids the existence of loops and only permits transitions from a state of graph-theoretical distance d to distance $d+1$.



A **B** **C**

Fig. 2 Some existing HMM topologies. **A.** a fully connected HMM; **B.** a circular HMM; **C.** a left-right HMM.

HMM models

Standard HMMs

The standard HMM formalization utilizes a number of simple assumptions with the intention of making the approach viable both mathematically and computationally. State sequences are modeled as a first-order MC. Each state generates one output.

Let $X_1, X_2, \ldots, X_i, \ldots$ denote the state variables in a standard HMM with state space $S = \{s_1, s_2, \ldots, s_N\}$. The initial state is selected according to the initial distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ and the transition probabilities are

$$a_{ij} = P(X_{t+1} = s_j \,|\, X_t = s_i).$$

Let Y_1, Y_2, ..., Y_i, ... denote the observed process generating symbols depending on the current state with the following probabilities

$$b_j(Y_{t+1} \,|\, Y_1, Y_2, \ldots, Y_t) =$$

$$P(Y_{t+1} \,|\, Y_1, Y_2, \ldots, Y_t, X_{t+1} = s_j).$$

Note that the output Y_{t+1} depends on the entire previous process, not just the current state X_{t+1}. However, in most applications in computational biology, Y_{t+1} depends only on the current state X_{t+1}.

Generalized HMM (GHMM)

A Generalized HMM (GHMM), also known as a hidden semi-Markov model, is structurally and operationally similar to standard HMMs but with a generalized distribution on the duration of a state, which is defined as the time the HMM stays at the particular state. In a standard HMM, the duration is geometrically distributed, that is, if p denotes the probability of self-transition in a state, then, the probability that l outputs are generated from the state is $p^{l-1}(1-p)$. However, in a GHMM, the duration d of a state X is usually selected from some generalized distribution, commonly derived from the training data and then called an empirical distribution. Each state generates outputs by first choosing the length according to some duration distribution, and then producing an output sequence of that duration. In addition, the positions in the output sequence from the state need not to be identically and independently distributed.

The GHMM model has been successfully implemented in gene finding programs, such as GENSCAN (*16*) and GENIE (*17*), and has been adopted by others for cross-species gene finding (*18*) since the exon lengths are not geometrically distributed.

Pair HMM (PHMM)

It represents yet another variant to the standard HMM and has been widely adopted for the generation of pairwise alignment of two sequences (*19*). The operational mechanism of PHMM is the same as standard HMM with the exception that each state outputs a pair of symbols. The probability of generating any particular alignment can be derived by taking the product of the probabilities at each step. A common problem encountered in sequence alignment is the difficulty in identifying the correct alignment when similarity is weak. Using PHMM, the probability that a given pair of sequences is related can be computed independent of a specific alignment by summing all possible alignments using the forward algorithm.

Generalized pair HMM (GPHMM)

It is a hybrid probabilistic model (*20*) that generalizes both GHMM and PHMM. A GPHMM can be considered as a sequence machine, generating a pair of observed sequences with different lengths in tandem.

Let $S = \{s_1, s_2, \ldots, s_m\}$ denote the state space of a GPHMM and $X_1, X_2, \ldots, X_L$ denote the sequence of hidden states that the GPHMM follows as it generates the pair of observed sequence $Y = Y_1, Y_2, \ldots, Y_T$ and $Z = Z_1, Z_2, \ldots, Z_U$, where $L \leq T, U$. As a standard HMM, the first state X_1 is distributed according to the initial distribution π_{X_1}, and moving from a state to another state occurs according to the associated transition probability. With each hidden state X_i, we associate a pair of duration lengths (d_i, e_i) generated from some joint distribution, representing the number of symbols in each observed sequence generated from the state. Let $p_i = \sum_{1 \leq k \leq i} d_k$ and $q_i = \sum_{1 \leq k \leq i} e_k$ denote the partial sum of the duration. Then, in state X_i, the GPHMM generates the sequences $Y[p_{i-1}+1, p_i]$ and $Z[q_{i-1}+1, q_i]$, according to joint distribution

$$b_{X_i}\big(Y[p_{i-1}+1, p_i], Z[q_{i-1}+1, q_i] \big| Y[1, p_{i-1}], Z[1, q_{i-1}]\big).$$

Here, we use the notation $Y[a, b]$ to represent the subsequence $Y_a, Y_{a+1}, \ldots, Y_b$ of Y.

In practice, only the sequences Y and Z observed and variables $L, X, \{(d_i, e_i) \mid i \leq L\}$ are hidden to us. Assume that we have all the observed sequences by the time the final state X_L is reached, then, we have $p_L = T$ and $q_L = U$. The probability of a particular combination of hidden and observed sequences is calculated as

$$P\big(X, Y, Z, \{(d_i, e_i) | \, i \leq L\}\big) = \pi_{X_1} f_{X_1}(d_1, e_1) b_{X_1}$$

$$\big(Y[1, p_1], Z[1, q_1]\big) \prod_{i=2}^{L} a_{X_{i-1} X_j} f_{X_i}(d_i, e_i) b_{X_i}$$

$$\big(Y[p_{i-1}+1, p_i], Z[q_{i-1}+1, q_i] \big| Y[1, p_{i-1}], Z[1, q_{i-1}]\big),$$

where $f_{X_i}(,)$ is the duration distribution at state X_i and a_{ij} is the transition probability from state i to state j.

Profile HMMs

They are linear, left-right models commonly used for detecting structural similarities and homologies. The profile HMM architecture (*21*) consists of three classes of states: the match state, the insert state and the delete state; and two sets of parameters: transition probabilities and emission probabilities. The match and insert states always emit a symbol, whereas the delete states are silent states without emission probabilities. Emitted symbols are assumed to be conditionally independent given the states. Match states model conserved positions of an alignment; insert states model insertions of residue(s) at a specific position, while delete states are responsible for deleting the consensus residue. The model always begins from the start state and finishes with the end state. Transitions from state to state progress from left to right through the model, with the exception of self-loops on insertion states. The gap penalties for insertions and deletions, by which positions of the conserved regions are controlled, are provided by transition probabilities back and forth the insert and delete states. A profile HMM topology widely used in protein sequence analysis is illustrated in Figure 3.

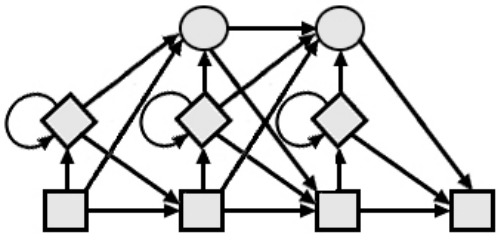


Fig. 3 A profile HMM topology. The square states are match states, the diamond states are insert states and the circles are delete states. State transition probabilities are indicated as arrows.

One main drawback of profile HMMs is that both signal and noise are treated equally, resulting in a large number of estimated emission parameters. This overfitting problem is typically avoided by using a regularizer (*22*) which replaces the observed amino acid distribution by its estimator as described in the next section.

In general, in almost all applications of HMMs, we are requested to solve one or more of the following questions:

1) Given an existing HMM and an observed sequence, what is the probability that the HMM could generate the sequence?

2) What is the optimal state sequence that the HMM would use to generate the observed sequence?

3) Given a large amount of data, how to find the structure and parameters of the HMM that best accounts for the data?

Both 1) and 2) can be solved in polynomial time using dynamic programming technique. The respective algorithms, called Forward and Viterbi, have a worst-case time complexity $O(NM^2)$ and space complexity $O(NM)$, for a sequence of length N and an

HMM of M states. However, there are only several heuristic algorithms for 3). Here, we omit the detailed description of these algorithms due to the space limit. For details of these algorithms, the reader is referred to the survey paper by Rabiner (*15*) or books written by Ewens and Grant (*23*) and Durbin *et al* (*21*).

Estimation of HMM Emission Probabilities

Overfitting occurs when the HMM adapts too well to the training data and includes random disturbances in the training set as being significant. As these disturbances do not reflect the underlying distribution, the performance of the HMM on the given dataset is affected. A variety of approaches known as regularization have been developed to address it. In general, regularizers can be broadly classified into two main categories: (1) substitution matrices and (2) statistical techniques.

The uses of substitution matrices for regulating the emission of noise and signals from HMMs have been widely adopted by several groups. The Gribskov profile (*24*) or average-score method (*25*) computes the weighted average of scores from a score matrix, such as the Dayhoff matrices (*26*) or the BLOSUM matrices (*27*). With this approach, each of the amino acid residues at every position along the peptide for a group of sequences previously aligned by structural or sequence similarity is assigned a weight to produce a matrix. Within each matrix, each row corresponds to a position of a certain length of protein sequence, and each column corresponds to an amino acid. An additional column contains a penalty for insertions or deletions at that position. Each entry of the matrix indicates a score for finding the amino acid at the position specified by a row and a column respectively. Scores are assigned by summing up the position specific weights, based on their sequence and the appropriate matrix. The work of Tatusov *et al* (*25*) involves using an evolving position-dependent weight matrix derived from a coevolving set of aligned conserved segments to perform iterative database scans. At each step, a cutoff score is obtained from the expected distribution of matrix scores for the chance inclusion of either a fixed number or a fixed proportion of false positive segments in the following iteration. Another approach known as feature-alphabet (*28*) divides the set of amino acids into disjoint feature sets and treats the contents of each feature sets equivalently. There are several ways to generate feature alphabets, such as computing their scores based only on the set of amino acids previously seen in a context (*29*), or together with the frequency of occurrences of amino acids.

Statistical techniques which include zero-offset, pseudocounts (*25*), and likelihood-based approaches such as Dirichlet mixture distribution (*30*) and efficient emission probability (EEP) estimation (*31*) represent an alternative way for regularization. The simplest statistical method is the zero-offset technique (*22*) that prevents probabilities from being estimated as zero by introducing the addition of a small positive zero-offset z to each count $s(i)$, the number of occurrences of amino acid i, to generate the posterior counts $X_s(i)$:

$$X_s(i) \leftarrow s(i) + z$$

However, a poor estimation to the amino acid distribution may result if the estimated probability distribution is constant due to non-occurrences of amino acid i in the sample. Hence, the pseudocount method represents a slight variant to the zero-offset technique that aims to overcome this problem by introducing a positive constant $z(i)$ for each amino acid:

$$X_s(i) \leftarrow s(i) + z(i)$$

The Dirichlet mixture method (*22*, *32*, *33*) offers a similar but more complex alternative to the pseudocount methods. Dirichlet mixtures are constructed by analyzing the amino acid distributions at specific positions in a large set of proteins using Dirichlet density functions. A Dirichlet density is a probability density function over all possible combinations of amino acids appearing in a particular position. It gives high probability to certain distributions (for example, conserved distributions or common features at a specific location) and low probability to others. The posterior counts of Dirichlet mixtures are defined as:

$$X_s(i) \leftarrow \sum_{1 \le c \le k} q_c \frac{\beta(z_c + \varepsilon)}{\beta(z_c)} \big(z_c(i) + s(i)\big),$$

where the vector $z_c + \varepsilon$ refers to the component-wise sum of the two vectors, β refers to the generalization of the binomial coefficients and is defined as

$$\beta(a) = \frac{\prod_i \Gamma\big(a(i)\big)}{\Gamma\big(\sum_i a(i)\big)},$$

in which Γ refers to the continuous generalization of the integer factorial function $\Gamma(n) = n!$ and $a(i)$ is the i-th coordinate of the vector a.

An alternative likelihood-based approach is presented by the EEP technique (*31*) that takes into account conservation of the alignment. Here, amino acids are first divided into the subset J_1 of effective (or conserved) amino acids and the subset J_2 of ineffective (noise) ones and then the estimation is based on the assumption that ineffective residues follow a background distribution. EEP explicitly models the conserved residues in the alignment instead of only considering the general characteristics of the amino acids by using the log-likelihood function of the multinomial distribution:

$$l = \sum_{i \in J} n_j \, log \, b_j \, ,$$

where n_j is a frequency of an amino acid j, b_j is the residue with the largest relative frequency with respect to its background probability b_j^o. The constraints of the log-likelihood function are determined as

$$\frac{b_i}{b_i^o} = \frac{b_e}{b_e^o}$$

$$\frac{\sum_{j \in J_1} b_j}{\sum_{j \in J_2} b_j} \leq c \frac{\sum_{j \in J_1} b_j^o}{\sum_{j \in J_2} b_j^o}$$

$$\sum_{j \in J_1} b_j + \sum_{j \in J_2} b_j = 1 \, ,$$

where $i, e \in J_2$ and c is a constant. The first constraint ensures that the mutual ratios of the ineffective residues remain the same as the background distribution. The second condition is only needed to make sure that the total proportion of the effective residues compared to the proportion of the ineffective ones does not increase too much when compared to the proportions in the background distribution. The optimization part is performed with the Lagrange multipliers method.

An important advantage of the EEP method over other regularization techniques is the reduction in the dimension of the parameter space. This decrease is significant for protein sequence alignments because only a small number of residues can be considered effective in conserved positions. Based on a study of 20 well-defined protein families by Ahola *et al* (*31*), it was shown that the EEP method is capable of detecting sequences with an average of 98% sensitivity and 99% specificity. The sensitivity proved to be better than the Dirichlet mixture distribution method, even if the number of emission parameters was reduced down to 11% of the original. As a consequence of the reduction of the parameter space, the variance of the ineffective residues decreases without influencing variance of the effective residues. This improvement is significant when shortening confidence intervals for emission probabilities and improves the sensitivity of database search results. However, despite the high accuracy of EEP, the technique does suffer from a major disadvantage of being unable to account for the physical and chemical characteristics of the amino acids, and thus, it ignores the relationships among the amino acids.

Applications of HMMs in Computational Biology

Algorithms such as BLAST (*32*) or FASTA (*34*) used in sequence comparison to infer biological function of a protein work well for highly similar sequences, nonetheless produce mediocre results for highly divergent sequences. Profile or motif based analyses that exploit information such as residual position and conserved residues derived from multiple sequence alignments to construct and search for sequence patterns were developed to address this deficiency. The following sections review recent applications of HMMs in the different areas of computational biology.

Pairwise sequence alignment

Pairwise sequence alignment involves aligning two sequences based on similarity between them to infer functional similarity. Using PHMM, Smith *et al* viewed the alignment problem as random process and adopted a probability model to tackle the problem (*19*). Most importantly, they presented a unique training method for estimating parameters (or probability) and extended the alignment model to allow multiple parameters sets, all of which are selected using HMM.

For training, one specifies a collection of pairs of sequences. After some initializations of the parameter values are assigned, training then takes place iteratively to learn the parameters that will produce overall maximal forward probabilities for the set of training pairs.

Suppose two sequences Y and Z with length $M = (M_1, M_2)$ are observed in a PHMM with state space $S = \{s_1, s_2, \ldots, s_m\}$. A position in the observation is specified by coordinates $r = (r_1, r_2)$ such that $1 \leq r_i \leq M_i$ for $i = 1, 2$. Then, the observation corresponding to the position r is the pair of subse-

quences $Y_1, Y_2, \ldots, Y_{r_1}$ and $Z_1, Z_2, \ldots, Z_{r_2}$. This pair of subsequences is denoted by $O[1 \to r]$. Moreover, a move from one position to another denoted by ε is one of *(0, 1)*, (1, 0), or (1, 1). For a position r, a move ε indicates a move from the position r to the position $r + \varepsilon$ if this is valid. The output corresponding to this valid move is denoted by $O[r \to r + \varepsilon]$, which is $(-, Z_{r_2+1}), (Y_{r_1+1}, -)$ or (Y_{r_1+1}, Z_{r_2+1}), depending on $\varepsilon = (0, 1), (1, 0)$ or $(1, 1)$, where '$-$' denotes a gap.

Finally, assume $X_1, X_2, \ldots, X_t$ is the hidden state sequence that the PHMM follows as it generates the observed pairs $P_1, P_2, \ldots, P_{t'}$ with the reduced sequence pair $O_{t'} = O$. Set

$$\xi_r(s_i, \varepsilon) = P(O_t = O[1 \to r], P_t = O[r - \varepsilon \to r], X_t = s_i \,|\, t \le t');$$

$$\eta_r(s_i, s_j) = P(O_t = O[1 \to r], X_t = s_i, X_{t+1} = s_j \,|\, t \le t').$$

Then, both $\xi_r(s_i, \varepsilon)$ and $\eta_r(s_i, s_j)$ can be computed easily given $P(O)$, the probability of observing O, which can be computed using the forward-backward algorithm in turn. Then, the training formulas are

$$\overline{\pi}_i \propto \sum_{\varepsilon} \xi_{\varepsilon}(s_i, \varepsilon)$$

$$\overline{a}_{ij} \propto \sum_{1 \le r \le M} \eta_r(s_i, s_j)$$

$$\overline{b}_i(x) \propto \sum_{\varepsilon, \varepsilon \le r \le M} \xi_r(s_i, \varepsilon)\,,$$

where the proportionality signs are used to indicate that the estimates are to be normalized to define probabilities.

Using this approach, multiple mutation matrices selection is made possible and estimation of model parameters given a training set of paired sequences can be done. However, this approach does suffer from various limitations including huge consumption of memory and time taken.

Multiple sequence alignment

Multiple sequence alignment (MSA) is commonly used in finding conserved regions in protein families and in predicting protein structures. Profile HMMs, in particular, have been applied with much success and continue to gain momentum. Multiple alignments from a group of unaligned sequences are automatically created using the Viterbi algorithm (*15*). Viterbi algorithm computes the probability of the maximum path by finding the most likely path through the HMM for each sequence. Each match state in the HMM corresponds to a column in the multiple alignment. A delete state is represented by a dash. Amino acids from insert states are either not shown or are displayed in lower case letters. It is this best alignment to the model that is used to produce multiple alignments of a set of sequences. Some popular implementations of profile HMMs include SAM (*35*, *36*) and HMMER (*14*).

The Sequence Alignment and Modeling system (SAM) is a collection of software tools for multiple protein sequence alignment and profiling using HMMs (*33*). SAM provides programs and scripts for SAM-T2K, which is an iterative HMM-based method for finding proteins similar to a single target sequence and aligning them. It aligns sequences to an HMM and improves the alignment by retraining the HMM on the sequences. A multiple alignment can be used to build an HMM, which can then be used to search for new members of the family. When new members are found, the HMM can be retrained to include them, new multiple alignments are made, and the process is repeated.

Alexandersson *et al* (*37*) implemented a cross-species gene finding and alignment program SLAM using GPHMM, which simultaneously aligns and predicts genes in two orthologous sequences. The input to SLAM consists of two sequences and an approximate alignment (*20*). The approximate alignment is used to reduce the search space for the Viterbi algorithm and allows for improvement in speed and reduction in memory usage. The main components of SLAM consist of a splice-site detector, an intron/intergene model, an exon pair scoring model, and a conserved noncoding sequence model. The accuracy of the technique is validated on the ROSETTA testset of 117 single-gene sequences as well as multigene lloxA cluster. SLAM compares favorably to other gene finders including GENSCAN (*16*), ROSETTA (*38*), SGP-1 (*39*), SGP-2 (*40*), TWINSCAN (*41*), particularly with regard to the false-positive rate.

Protein homology detection

In the protein homology problem, the goal is to determine which proteins are derived from a common ancestor. The common ancestor model makes the assumption that, at some point in the past, each protein sequence in a family was derived from a common

ancestor sequence. That is, at each amino acid position in the sequence, the observed amino acid occurs due to a mutation (or set of mutations) from a common amino acid ancestor. There are many protein sequences sharing similarity but there are many with varying divergence as well such that structural and functional similarity is hard to detect based on sequence data alone.

Pairwise sequence comparison methods such as BLAST accept two sequences and calculate a score for their optimal alignment. This score may then be used to decide whether the two sequences are related. Park *et al* (*42*) showed that profile-based methods, particularly profile-based HMMs (*10*, *13*), which consider profiles of protein families, perform much better than pairwise methods. A more recent study by Lindahl and Elofsson (*43*) compared the relative performance of pairwise and profile methods.

Examples of popular profile HMM software packages include SAM (*35*, *36*) and HMMER (*14*). HMMER (*14*) provides the necessary model building and scoring programs for homology detection. It contains a program that calibrates a model by scoring it against a set of random sequences and fitting an extreme value distribution to the resultant raw scores; the parameters of this distribution are then used to calculate accurate E-values for sequences of interest.

Truong *et al* (*44*) utilized the HMMER package to classify unknown protein sequences into subfamilies within structurally and functionally diverse superfamilies. Their technique begins with an MSA of the subfamily followed by constructing an HMM database representing all sliding windows of the MSA of a fixed size. Finally, they constructed an HMM histogram of the matches of each sliding window in the entire superfamily. The complete set of HMMs created from all subfamily signatures is concatenated to build the HMM database for the protein superfamily. The analysis of a query sequence follows a two-step process. First, search the query sequence for the conserved domain of the protein superfamily. If the conserved domain is found, then search for subfamily signatures. If the subfamily signatures are found, the sequence belongs to the subfamily whose signature has the lowest e-value. Otherwise, the sequence is classified to a new protein superfamily. The classification system has achieved an equivalent level of success as most profile and motif databases. This technique was applied to find subfamily signatures in the cadherin and the EF-hand protein superfamilies. The HMM histograms of the analyzed subfamilies revealed information about their Ca^{2+} binding sites and loops.

Protein structure prediction

The strong formalism and underlying theory of HMMs and extensive applications in sequence alignment have prompted researchers to apply them to the domain of protein structure prediction (*36*, *45*). Identification of homologous proteins becomes important since these proteins descending from common ancestry root share similar overall structure and function.

Karplus *et al* (*45*) made protein structure prediction for target sequences in CASP3 relying solely on sequence information using the method SAM-T98. This iterative method steps through the template library and target models several times. The first step involves building an HMM from a sequence or a multiple sequence alignment. The resulting HMM is used to score a non-redundant database. Sequences that exceed certain threshold are collected to form the training set. This threshold is relaxed in each iteration to include less similar sequences that may still be homolog. Scoring is based on log odds where the likelihood of HMM-generated sequence is compared to that of null model generated sequence. Null model in this case is taken as the reverse of the HMM. Re-estimation of the HMM using these sequences is based on sequence weighting and Dirichlet mixture prior follows. The final step realigns the training set using the re-trained HMM. The multiple alignments from this step serve as initial input in next iteration. Database searching is then carried out based on the HMM constructed from the final multiple alignment, known as SAM-T98 alignment. SAM-T98 considered only sequence information and hence yielded poor results in more difficult targets. It was subsequently augmented to include structural information in SAM-T02. Karplus *et al* also extended the use of SAM-T98 multiple alignments of the target sequences to secondary structure prediction where favorable results were observed.

A coiled-coil structure is formed by the intra- or extra-molecular association of two or more alpha-helices, which wrap around each other. Each of these single helices is referred as a coiled-coil domain (CCD). CCDs are frequently involved in protein-protein interactions, and play central roles in diverse processes including signaling and transcription. Most CCDs have a "heptad" repeat that is a periodic sequence pattern of seven characteristic residues: the

two hydrophobic core positions are designed a and d; they are separated by two positions b and c; and b and c are separated by three positions (e, f, and g) in turn that are occupied by mainly hydrophilic and often charged residues.

Delorenzi and Speed (*46*) developed a 64-state circular HMM for recognition of proteins with a CCD that outperforms traditional Position Specific Scoring Matrix (PSSM) using 150-fold cross-validation on datasets extracted from various protein databases including CCDs, SWISSPROT and PDB. This approach initializes the background state to 0 and the remaining 63 states are assigned a group number 1–9 with a letter that refers to the heptad position. Groups 1–4 model the first four residues in a CCD (the N-terminal helical turn); Group 5 models internal coiled-coil residues; while Groups 6–9 model the last four residues (the C-terminal turn). In the model, a CCD has a minimal length of nine, one residue per group.

In a more recent work, Bagos *et al* came up with an HMM method based solely on amino acid sequence capable of predicting the transmembrane β-strands of the outer membrane proteins of gram-negative bacteria, and discriminating those from water-soluble proteins in large datasets (*47*). The model maximizes the probability of correct predictions instead of likelihood of the sequences. This method fares equally good in terms of true positives and overall topologies as compared to some of the best method (*48*, *49*) proposed so far for the prediction of transmembrane β-barrel proteins.

Numerous previous works on structural studies (*50*, *51*) were based on single dimensional HMM profile encoding structural information in symbols (that is, H for helix), none of which work with 3D coordinates. Alexandrov and Gerstein used 3D HMMs to explicitly model spatial coordinates to compare protein structures (*52*). Conventional dynamic programming fails when attempting to match query structure of the model due to the assumption that the best match between query and model in any region of the alignment is independent and does not affect the optimum match before it. They made the core structures using ellipsoidal Gaussian distributions by centering on aligned Cα positions. Each Gaussian distribution is then normalized to 1 to obtain probability distribution based on coordinates. The cores are essentially structural profiles similar to sequence profiles, each representing a statistical distribution of potential coordinates. Each match state denotes the probability of a given Cα position falling within a prescribed volume, where the probability is the coordinate differences. Score increases if the aligned Cα of the query is closer to the centroid and vice versa. The 3D HMMs were tested on globin family and IgV fold and other SCOP domains. Their results are promising.

Genomic Annotation

With many genomes having been sequenced, HMMs have been increasingly applied in computational genomic annotation. In general, computational genome annotation includes structural annotation for genes and other functional elements, and functional annotation for assigning functions to the predicted functional elements.

The sequences of entire chromosomes consist of a collection of genes separated from each other by long stretches of "junk" sequences. The computational approach for gene identification involves bring together a large amount of diverse information. Up to now, the most popular and successful gene finder probably is GENSCAN (*16*). It is based on generalized HMMs. We sketch it below in order to illustrate the basic concept of an HMM-based gene finder.

Roughly speaking, a protein-coding gene consists of a consecutive sequence of the DNA that is transcribed into RNA, called premessenger RNA (or pre-mRNA for short). This pre-mRNA consists of an alternating sequence of exons and introns. After transcription, the introns are edited out, and the final molecule, called mRNA, is translated into protein.

The region of the DNA before the start of the transcribed region is called the "upstream region". This is where the promoter of the gene locates. In the promoter region, transcription factors bind and initiate transcription. The 5′ untranslated region (5′UTR) follows the promoter. This stretch does not get translated into protein. Near the end of 5′UTR is a signal that indicates the start of translation, called the translation initiation signal (TIE); TIE just locates before the first codon in the first exon. TIE is followed either by a single exon or by a sequence of exons separated by introns. An intron may break a codon in any position. Finally, following the final exon is the 3′ untranslated region (3′UTR), which is another stretch of sequence that is transcribed but not translated. Near the end of the 3′UTR are poly-A signals indicating the end of transcription. Each poly-A signal is six bases long with the typical sequence AATAAA.

GENSCAN model has two identical components

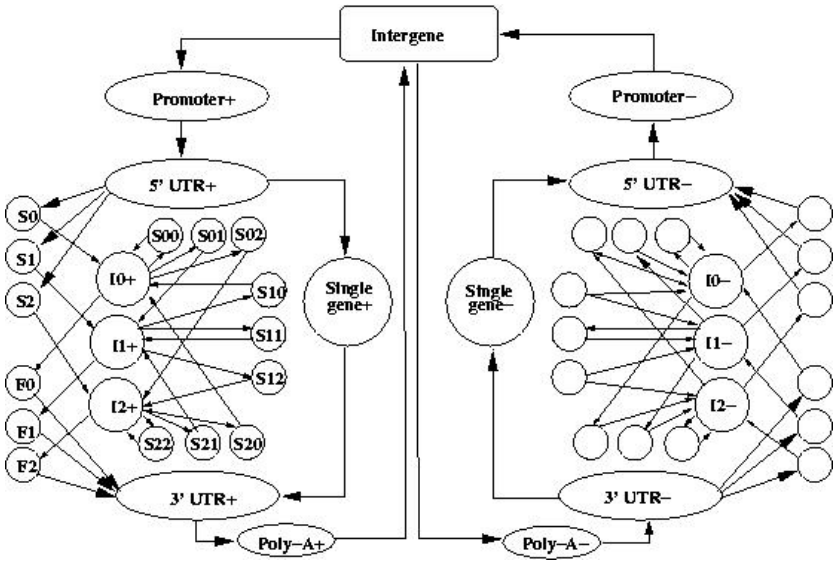


Fig. 4 The complete GENSCAN model.

(Figure 4) for finding genes in both the forward (5′ to 3′) and reverse directions in one pass. In the left component corresponding to the forward direction, the intergenic, promoter, 5′UTR , 3′UTR and poly-A regions are modeled with a state separately. However, modeling the exons and introns is more complicated. It uses 19 states drawn between the 5′UTR and 3′UTR states. There are two paths from the 5′UTR state to the 3′UTR state. The path through the single gene state corresponds to single exon genes. The reason for considering single exon genes separately is that the distribution of their lengths is quite different from that of the multiexon genes. In a multiexon gene, a single codon can be split between two exons. Therefore, 18 states are used for copying these different combinations.

In this generalized HMM model, all the transition probabilities from a state to itself are zero, and when the process visits a state, it produces a sequence of length following a distribution such as geometric distribution.

With the model, given an uncharacterized genomic sequence, GENSCAN applies a generalized Viterbi algorithm to obtain an optimal parse. The parse gives a list of the states visited and the lengths of the sequences generated at those states. Thus, a decomposition of the original sequence into gene predictions is obtained.

Recently, Meyer and Durbin (*53*) developed DOUBLESCAN, a pair HMM model, for *ab initio* prediction of gene structures using two different algorithms: the Viterbi algorithm and the stepping stone algorithm. The emission probabilities are based on match exon states in orthologous genes with identical coding lengths derived from a subset of the data set in Jareborg *et al* (*54*) and are estimated using Dirichlet distribution. Marginalization is performed for all states except the stop state to introduce symmetry with respect to the two sequences into the emission probabilities and avoid potential compositional bias. Transition probabilities are initialized to values estimated from event frequencies and manually refined. Transitions into splice site states are controlled by posterior probabilities generated using a splice site predictor (*55*) while transitions between the match intergenic and the START are controlled by a weight matrix model. This method performs well with a higher sensitivity and specificity as compared to GENSCAN.

Walker *et al* (*56*) employed two HMMs simultaneously to identify prokaryotic translation initiation sites. Specifically, the HMM-termed product hidden Markov model (PROD-HMM) with a total of 100 states attempts to model species-specific trinucleotide frequency patterns in two orthologous DNA sequences adjacent to a translation start site and to detect the contrasting amino acid substitution rates that differentiate prokaryotic coding from intergenic regions.

Conclusion

This paper has explored various topologies of HMMs and estimation probabilities. Subsequently, we presented several of the variant models from the stan-

dard HMMs. We then reviewed recent applications using HMMs in areas like sequence alignment, homology detection, and so on. Hopefully, this review can provide an insight into applications of HMMs in computational biology. In general, application of HMMs is not straightforward, since the architecture of the model often has to be expressly designed.

Finally, despite many of these models have proven to be successful, these models suffer from certain limitations. The linear nature of HMM also makes it difficult to capture higher-level information or correlations among amino acids. Prediction of actual distance when a protein folds as opposed to when it is spread out, and prediction of chemical and electrical interactions are just some examples. These limitations have prompted research into new kinds of statistical models (*21*).

Appendix—HMM Software

Name	Description	URL
HMMER	It produces profile hidden Markov models for homolog search in a database.	http://hmmer.wustl.edu/
SAM	A suite of tools for biological sequence analysis including homology detection, secondary structure prediction and so on.	http://www.cse.ucsc.edu/research/compbio/sam.html
TMHMM	It models and predicts the location and orientation of alpha helices in membrane-spanning proteins.	http://www.cbs.dtu.dk/services/TMHMM/
SignalP	A signal peptide prediction program. It was originally developed using artificial neural network and later updated to HMMs.	http://www.cbs.dtu.dk/services/SignalP/
Phobius	It predicts transmembrane regions and signal peptide.	http://phobius.cgb.ki.se/
Meta-MEME	A motif-based hidden Markov model used for database search for homologs.	http://metameme.sdsc.edu/
SATCHMO	It aligns sequences and constructs tree using HMMs in situation where sequence identity is low.	http://www.drive5.com/lobster/index.htm
COACH	It performs pairwise alignment or profiles of alignments.	http://www.drive5.com/lobster
HMMSPECTR	A protein structure prediction tool.	http://biology.sdsc.edu/HMM-SPECTR/
HMMSTR	A tool to predict the structure (including secondary, local, supersecondary, and tertiary) of proteins from their sequences.	http://www.bioinfo.rpi.edu/ bystrc/hmmstr/server.php

Listed in this table are programs mentioned in this survey that employ HMMs and are freely available for use or download.

References

1. Sheynin, O. 1988. A Markov's work on probability. *Arch. Hist. Exact Sci.* 39: 337-377.
2. Blackwell, D. and Koopmans, L. 1957. On the identifiable problem for functions of finite Markov chains. *Ann. Math. Stat.* 28: 1011-1015.
3. Burke, C.J. and Rosenblatt, M. 1958. A Markovian function of a Markov chain. *Ann. Math. Stat.* 29: 1112-1120.
4. Gilbert, E.J. 1959. On the identifiablity problem for functions of finite Markov chains. *Ann. Math. Stat.* 30: 688-697.

5. Heller, A. 1965. On stochastic processes derived from Markov chains. *Ann. Math. Stat.* 36: 1286-1291.
6. Baum, L.E., *et al.* 1972. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* 41: 164-171.
7. Churchill, G.A. 1989. Stochastic models for heterogeneous DNA sequences. *Bull. Math. Biol.* 51: 79-94.
8. Stultz, C.M., *et al.* 1993. Structural analysis based on state-space modeling. *Protein Sci.* 2: 305-314.
9. White, J.V., *et al.* 1994. Protein classification by stochastic modeling and optimal filtering of amino-acid sequences. *Math. Biosci.* 119: 35-75.
10. Krogh, A., *et al.* 1994. Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.* 235: 1501-1531.
11. Baldi, P., *et al.* 1994. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA* 91: 1059-1063.
12. Eddy, S.R., *et al.* 1995. Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol.* 2: 9-23.
13. Eddy, S.R. 1996. Hidden Markov models. *Curr. Opin. Struct. Biol.* 6: 361-365.
14. Eddy, S.R. 1998. Profile hidden Markov models. *Bioinformatics* 14: 755-763.
15. Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77: 257-286.
16. Burge, C. and Karlin, S. 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268: 78-94.
17. Reese, M.G., *et al.* 2000. Genie—gene finding in *Drosophila melanogaster*. *Genome Res.* 10: 529-538.
18. Kulp, D., *et al.* 1996. A generalized hidden Markov model for the recognition of human genes in DNA. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 4: 134-142.
19. Smith, L., *et al.* 2003. Hidden Markov models and optimized sequence alignments. *Comput. Biol. Chem.* 27: 77-84.
20. Pachter, L., *et al.* 2002. Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J. Comput. Biol.* 9: 389-399.
21. Durbin, R., *et al.* 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, UK.
22. Karplus, K. 1995. Evaluating regularizers for estimating distributions of amino acids. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 3: 188-196.
23. Ewens, W. and Grant, G. 2001. *Statistical Methods in Bioinformatics.* Springer-Verlag, New York, USA.
24. Gribskov, M., *et al.* 1987. Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA* 84: 4355-4358.
25. Tatusov, R.L., *et al.* 1994. Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc. Natl. Acad. Sci. USA* 91: 12091-12095.
26. Dayhoff, M.O., *et al.* 1978. A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure* (ed. Dayhoff, M.O.), Vol. 5, pp.345-352. Natl. Biomed. Res. Found., Washington DC, USA.
27. Henikoff, S. and Henikoff, J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* 89: 10915-10919.
28. Smith, R.F. and Smith, T.F. 1990. Automatic generation of primary sequence patterns from sets of related protein sequences. *Proc. Natl. Acad. Sci. USA* 87: 118-122.
29. Karplus, K. and Hu, B. 2001. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics* 17: 713-720.
30. Sjolander, K., *et al.* 1996. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* 12: 327-345.
31. Ahola, V., *et al.* 2003. Efficient estimation of emission probabilities in profile hidden Markov models. *Bioinformatics* 19: 2359-2368.
32. Altschul, S.F., *et al.* 1990. Basic local alignment search tool. *J. Mol. Biol.* 215: 403-410.
33. Brown, M., *et al.* 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 1: 47-55.
34. Pearson, W.R. and Lipman, D.J. 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA* 85: 2444-2448.
35. Hughey, R. and Krogh, A. 1996. Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Appl. Biosci.* 12: 95-107.
36. Karplus, K., *et al.* 1999. Predicting protein structure using only sequence information. *Proteins* Suppl 3: 121-125.
37. Alexandersson, M., *et al.* 2003. SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.* 13: 496-502.
38. Batzoglou, S., *et al.* 2000. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.* 10: 950-958.
39. Wiehe, T., *et al.* 2001. SGP-1, prediction and validation of homologous genes based on sequence alignments. *Genome Res.* 11: 1574-1583.
40. Guigo, R., *et al.* 2000. An assessment of gene prediction accuracy in large DNA sequences. *Genome Res.* 10: 1631-1642.
41. Korf, I., *et al.* 2001. Integrating genomic homology into gene structure prediction. *Bioinformatics* 17: S140-148.

42. Park, J., *et al.* 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.* 284: 1201-1210.
43. Lindahl, E. and Elofsson, A. 2000. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.* 295: 613-625.
44. Truong, K. and Ikura, M. 2002. Identification and characterization of subfamily-specific signatures in a large protein superfamily by a hidden Markov model approach. *BMC Bioinformatics* 3: 1.
45. Karplus, K., *et al.* 1997. Predicting protein structure using hidden Markov models. *Proteins* Suppl 1: 134-139.
46. Delorenzi, M. and Speed, T. 2002. An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics* 18: 617-625.
47. Bagos, P.G., *et al.* 2004. A Hidden Markov Model method, capable of predicting and discriminating beta-barrel outer membrane proteins. *BMC Bioinformatics* 5: 29.
48. Martelli, P.L., *et al.* 2002. A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. *Bioinformatics* 18: S46-53.
49. Liu, Q., *et al.* 2003. A HMM-based method to predict the transmembrane regions of beta-barrel membrane proteins. *Comput. Biol. Chem.* 27: 69-76.
50. Sonnhammer, E., *et al.* 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 6: 175-182.
51. Bystroff, C., *et al.* 2000. HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.* 301: 173-190.
52. Alexandrov, V. and Gerstein, M. 2004. Using 3D Hidden Markov Models that explicitly represent spatial coordinates to model and compare protein structures. *BMC Bioinformatics* 5: 2.
53. Meyer, I.M. and Durbin, R. 2002. Comparative *ab initio* prediction of gene structures using pair HMMs. *Bioinformatics* 18: 1309-1318.
54. Jareborg, N., *et al.* 1999. Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res.* 9: 815-824.
55. Levine, A. and Durbin, R. 2001. A computational scan for U12-dependent introns in the human genome sequence. *Nucleic Acids Res.* 29: 4006-4013.
56. Walker, M., *et al.* 2002. A comparative genomic method for computational identification of prokaryotic translation initiation sites. *Nucleic Acids Res.* 30: 3181-3191.

This work was partly supported by the Singapore BioMedical Research Council research grant BMRC01/1/21/19/140.