Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 7 (12): 1626-1632, 2007
ISSN 1812-5654

Improving the Performance of an HMM for Protein Family Modelling

Mohamed Hamza El-Sayed and Ahmed M. Khedr
Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt

Abstract: A hidden Markov model for protein modelling consists of sub-models for alpha-helix, beta-sheets, coil and possibly more. It is described how to estimate the model parameters as a whole from labeled sequences instead of estimating the parameters from the individual parts independently from subsequences. It is argued that the standard maximum likelihood ML estimation is not the optimal for training such model. In this study a new method is used where instead of estimating the parameters of model that maximizing the probability of the protein sequences (ML), we maximize the probability of the correct labels prediction, such a criterion is called conditional maximum likelihood CML. The advantage of this method is to optimize recognition of model. We tested our method on some of protein families such as L-asparagines, we noted that the performance of HMM is improved in prediction process.

Key words: Protein family, class hidden Markov model, conditional maximum likelihood estimation

INTRODUCTION

As the proteomics project evolve automated annotation of the protein sequences becomes increasingly important. One of the difficult tasks is to model protein family. Several groups are applying probabilistic modelling using HMM, neural networks, comparative modelling, Bayesian network and more. However HMM gained great respect because it satisfies great results in speech recognition. The 2 central ideas of working in this field are to integrate or improve the models. In this paper, we improve the performance of profile HMM in the application of protein family modelling by applying new training technique (CML) instead of using the traditional one (ML).

Protein family: A protein family is a group of evolutionary related proteins sequences. Although there is a divergence between protein sequences (primary structure), but all sequences have the same general function and a common three dimensional structure. It can be considered also as large collection of multiple sequence alignments. Figure 1 is an example of protein family (l-asparagines).

Hidden Markov models are usually being estimated by maximum likelihood ML and this is true also for the HMM based protein modelers reported previously (Anders *et al*., 1993). The standard ML method optimizes the probability of the observed sequences (i.e., protein sequences in family), but for prediction (i.e., decoding process) one would rather want to maximize the probability of the correct prediction. In this study a method is described for estimating HMM from labeled sequences which maximize the probability of the correct labeling instead of observed sequence only and this substantially improve the performance of protein modelling with HMM. This method is an integral part of HMM protein modelling which is an HMM for protein sequence prediction. Here we focus on the description of the method and the results of HMM protein modeling will be used to illustrate the power of this method.

Profile HMM (PHMM): An HMM can be considered as a structure that generates protein sequences by random process. This structure and the corresponding random process of HMM can be described as follows: HMM contains a sequence of M states, which we call match states, corresponding to positions in proteins or columns in a multiple alignment. Each of these states can emit a letter x (x is amino acid) from 20 amino acids, according to the probability distribution $p(x/m_k)$ (which is called emission probability $E_{mk}(x)$, k = 1,..., m), each of the match states (m_k, k = 1,..., m), has distinct probability distributions or emissions. For each match state m_k there is a delete state d_k that doesn't emit any amino acid (dummy state used to skip m_k) and there are a total m + 1 insert states to either side of the match states which emit amino acid in exactly the same way as the match states, but using the probability distributions $p(x/i_k)$ at state i_k. For convenience, there are dummy BEGIN and END states, denoted as m_0 and m_{m+1}, respectively. For each state, there are three possible transitions to other

Corresponding Author: Ahmed M. Khedr, Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt Tel: +20 010 209 3552

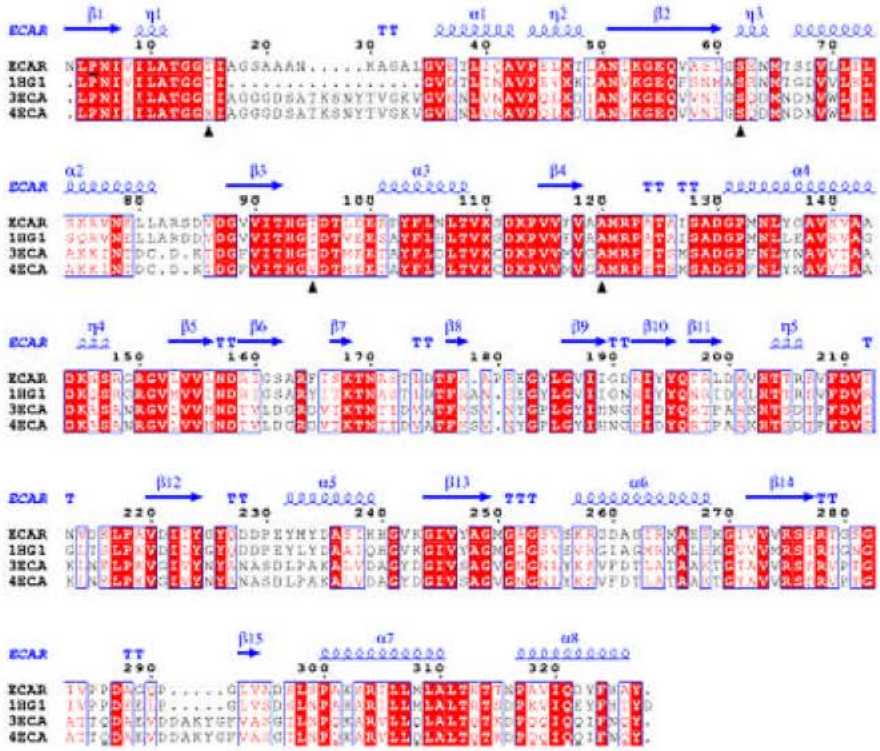

Fig. 1: Sequence alignment of the type II l-asparagines. NCBI accession number for l-asparagines are in brackets: seq1 *E. carotovora* (AY560097); seq2 *E. chrysanthemi* (CAA32884); seq3 *Wolinella succinogenes* (P50286); seq4 *P. pseudomonas* 7A (AAB31750); seq5 *Escherichia coli* K12 (NP417432); seq6 Pombe (NP590286)

state as: transition into match or delete states always move forward in the model, whereas transitions into insert states do not. Note that multiple insertions between match states can occur, since the self-loop on the insert state allows transitions from the state to itself. The transitions probability from state q to state r is called $\Gamma(r/q)$ or shortly a_{qr}.

A protein sequence can be generated by a random walk through the model as follows: commencing at state m_0, choose a transition to m_1, i_0 or d_1 randomly according to the transition probability $\Gamma(m_1/m_0)$, $\Gamma(i_0/m_0)$ and $\Gamma(d_1/m_0)$. If m_1 is chosen, emit the first amino acid x_1 from the probability distribution $p(x_1/m_1)$ and choose a transition to the next state according to probability transitions $\Gamma(*/m_1)$, where * indicates any possible next state, if the next state is the insert state i_1, generate amino acid x_2 from emission probabilities $p(x_2/i_1)$ and select the next state from $\Gamma(*/i_1)$, if the next state is delete state d_2 then emit no amino acid and choose the next state from transition probabilities $\Gamma(*/d_2)$. Continue using this manner all the way to the END state (Fig. 3), to generate the sequence of amino acids (x_1, x_2,..., x_L) by the path of states ($s = q_0, q_1,..., q_{N+1}$). Because the delete states emit no amino acid then $N \geq L$. The probability of the event that the path q_0, q_1,..., q_{N+1} is taken and the sequence x_1, x_2,..., x_L is generated can be obtained using Eq. 1.

$$p(x_1....x_L, q_0.....q_{N+1}/\text{Model}) = \Gamma(m_{N+1}/q_N) \times \prod_{i=1}^{N} \Gamma(q_i/q_{i-1}) p(x_i/q_i) \quad (1)$$

Where N is the number of states in path, we set $p(x_1/q_1) = 1$, if q_1 is a delete state. The probability of any sequence x_1, x_2,..., x_L of amino acids is the sum of all possible paths that could generate that sequence using Eq. 2.

$$p(x_1....x_L/\text{Model}) = \sum_{\text{Paths } q_0 ... q_{N+1}} p(x_1.....x_L, q_0.....q_{N+1}/\text{Model}) \quad (2)$$

By this way, the probability distribution of the sequences space is defined. The goal is to find a model (i.e., a model length and probability parameters) that accurately describes a family of proteins by assigning large probabilities to sequences in that family.

Estimating the parameters of profile HMM from training sequences: The most traditional approach is to estimate the parameters entirely automatically from a set of unaligned primary sequences using EM algorithm (i.e., learning process) (Anders *et al*., 1993). This approach can in principle find the model that best describes a given set of sequences. So given a set of training sequences s_1, s_2,..., s_n, one can see how well a model fits them by calculating the probability that it generates them. This probability is simply the product of terms of the form given by Eq. 2 which yields Eq. 3:

$$p(s_1, s_2, ..., s_n / \text{Model}) = \prod_{a=1}^{n} p(s(a) / \text{Model}) \quad (3)$$

Where each term p(s(a)/Model) is calculated by substituting x_1, x_2,..., x_L as s(a) in Eq. 2. This is called likelihood of the model. One would like this value in Eq. 3 to be high which is called maximum likelihood (ML) method. There are some algorithms that give arbitrary starting point, find a local maximum by iteratively re-estimating the model in such a way that the likelihood increases in each iteration. The most common algorithm is Baum Welch (forward backward algorithm) (Rabiner, 1989) which is a version of the general EM algorithm.

Baum-Welch algorithm: Is the standard algorithm for maximizing the HMM parameters when the paths through the model for each training sequence are unknown. This algorithm has two main steps, in the first step, the expected number of times of each transition and emission which are occurred in the training sequences will be computed (expectation step). In the second step, the transitions and emissions probabilities are updated using re-estimation formulas. The main steps of Baum Welch algorithm can be briefly described as follows:

- Shuffle the training protein sequences to avoid hill climbing,
- Initialize the transitions and emissions probabilities,
- Apply the forward algorithm to each sequence for obtaining forward probabilities values,
- Apply the backward algorithm to each sequence for obtaining backward probabilities values,
- Use the forward and backward values to calculate transitions and emissions counts (expected values) expectation step,
- Use the transitions and emissions counts for calculating transitions and emissions probabilities (θ),
- Calculate the target function (log p(training sequences/θ)), where training sequences are s_1, s_2,..., s_n,
- Repeat steps (1-7) iteratively and after each iteration replace θ_{i-1} by θ_i.

The iteration here will be terminated when there is convergence between target function values and not between θ values (Rabiner, 1989).

Class HMM: The protein modeling problem is part of a general class of problems which has a class label associated with each observation or sequence. In protein modeling the observations are amino acids and the label classes are for instance alpha helix, beta-sheets, loops and the rest. The traditional approach to handle this problem is to separately estimate models for sequences representing each class and latter merge them into one large model. However in HMM framework it is possible to estimate the whole combined model directly from the labeled sequences if the states of model are labeled, such a model called a Class Hidden Markov Model (CHMM) (Anders, 1994). In our work, we label protein sequence as the following: a for alpha helix, b for beta-sheets, l for coils and c for the rest as shown in Fig. 2.

We limit our work to one label for each state, which is probably the most useful approach in protein modelling. This means that:

- Some states are labeled with a for alpha-helix,
- Some states are labeled with b for beta-sheets,
- Some states are labeled with l for coils and
- Some states are labeled with c for the rest.

and the emission probability of label at each state will be one or zero. Therefore, the emission probability of a symbol (any amino acid) and its label (e.g., a) will be:

$$E_i(\text{symbol}, a) = E_i(\text{symbol}) \times E_i(a) = \begin{cases} E_i(\text{symbol}); \text{if the state i is labeled a} \\ 0 \quad \text{Otherwise} \end{cases} \quad (4)$$

The results will be the same with the other labeled (b, c, or l) states.

The aim of CHMM is that a decoding of amino acids sequences yields the correct labels. That is the objective of training is to optimize the probability of the labeling, [Probability (label sequences/amino acids sequences, model)], rather than the probability of the amino acids sequence, [Probability (amino acids sequences/model)] which is optimized in profile HMM training.

A CHMM has 2 modes, the decoding mode in which an unlabelled sequence of amino acids has to be labeled and it works exactly like a profile HMM and the training mode where the labels of amino acids sequences are taken

Fig. 2: Represent a segment of one amino acids sequence in family and it's label sequence

into account. One should notice that in profile HMM, a state can emit symbol with probability distribution over letters of symbols (amino acids) but in CHMM it can emit symbol with associated label as in Fig. 2.

To estimate the parameters of a CHMM, a set of labeled sequences is needed. Let us call a sequence $X = x_1, x_2,..., x_l$ and the corresponding label sequence $Y = y_1, y_2,..., y_l$ The estimation of the parameters by standard ML, if the model parameters are called θ and we assume there is only one sequence in the training set, then the ML solution will be:

$$\theta^{ML} = \arg\max p(x, y/\theta) \quad (5)$$

The probability p(x, y/θ) can be calculated by a trivial modification of the standard forward algorithm (Anders, 1994) where only the valid paths through the model are allowed. A valid path is one in which the state labels agree with the observed labels. For instance, a part of the sequence labeled by a for alpha-helix can only match a state also labeled a. The same holds for the backward algorithm and thus the forward-backward (or Baum Welch) re-estimation procedure can be used to maximize Eq. 5.

The model obtained by this way is in most cases identical to a model combined of sub-models estimated from each class independently, which is the usual approach and the main advantage is that the combined model can be estimated in one go from scratch. Another advantage is that the transitions between the sub-models are also automatically estimated in the CHMM, in the other hand the estimation of the several transitions between sub-models for the individual classes is not straight forward when the sub-models are trained independently.

In the estimated CHMM model, the prediction of labels for a sequence $X = x_1, x_2,..., x_l$ can be done by using viterbi algorithm (Rabiner, 1989) to find the most probable state sequence $\pi^* = \pi_1^*, \pi_2^*, \ldots, \pi_L^*$ that satisfies the following equation:

$$\pi^* = \arg\max p(x, \pi/\theta), \forall\pi \quad (6)$$

then by taking off the label of each state in the path, we will get the sequence labels.

Conditional maximum likelihood estimation: In the ML approach the probability that sub-model can emit the protein sequences in a certain class (e.g., alpha-helix) is maximized. Even after maximizing this probability it may well happen that these sequences have a higher probability given a model estimated for different class (e.g., beta-sheets). ML is not discriminative, so the probability of the observed protein sequence under the model is not the relevant quantity for prediction; one really only cares about getting the predicted labeling correct. Therefore, maximizing the probability of the labeling instead seems more appropriate, this is called conditional maximum likelihood (CML) (Rabinar, 1991). Therefore instead of Eq. 5 the CML estimation will be:

$$\theta^{CML} = \arg\max p(y/x, \theta) \quad (7)$$

This probability can be rewritten as:

$$p(y/x, \theta) = \frac{p(x, y/\theta)}{p(x/\theta)} \quad (8)$$

The numerator is the same probability as in Eq. 5 which can be calculated as described above. The denominator is even simpler: it is the probability of the unlabelled protein sequence, i.e., the probability usually calculated in an HMM disregarding any labeling, this probability is calculated by standard forward algorithm.

The probability p(x, y/θ) is the sum over all valid paths through the model and p(x/θ) is the sum of the probability over all valid and invalid paths. Therefore, maximizing Eq. 8 corresponds to making probability p(x, y/θ) comes as close as possible to p(x/θ); i.e., to minimize the probability of all the invalid paths.

To actually maximize Eq. 8 is trickier than the maximization of Eq. 5, because an expectation-maximization EM algorithm like the Baum Welch doesn't exist. In this study, the extended Baum Welch algorithm (Normandin and Morgan, 1991) is used. The CML method is obviously not limited to HMM, it has proven successful in a neural network and HMM hybrid called hidden neural network or HNN (Riis and Krogh, 1997).

Simulation class HMM for protein family modeling: We constructed a model structure of 2n match states, 2n + 2

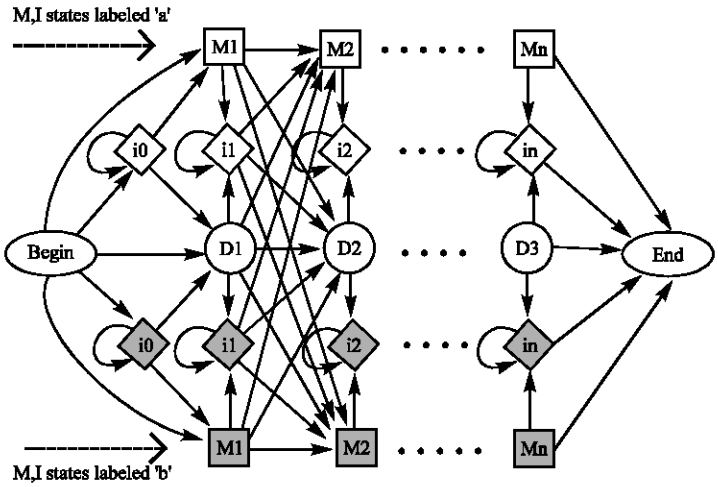

Fig. 3: The combined two sub models (CHMM), (1) the top sub model represents alpha-helix and (2) the bottom sub model represents beta-sheets

insert states and n delete states. The model consist of 2 sub-models, one corresponding to alpha helix and the other to beta sheet regions. We assigned n match states for class label a which represents alpha helix and n match states for class label b which represents beta sheets, the same thing for 2n + 2 insert states. The two sub models share the n delete states and begin and end states as shown in Fig. 3.

Notice that the model length is n because the length of sequences in multiple alignment is n, or the average length of unaligned sequences is n. This model is oversimplified to be scientifically useful, as it should include match and insert states for corresponding coil, loops and the rest.

The main characteristics of our model are the following:

- Transition probabilities in our model
  - Each match state has the following four transition probabilities
    a) Forward transition to the next match state with same label,
    b) Forward transition to the next match state with different label,
    c) Forward transition to the insert state in its layer and
    d) Forward transition to the next delete state.
  - Each insert state has the following four transition probabilities
    a) Forward transition to the next same label match state,
    b) Forward transition to the next match state with different label,
    c) Forward transition to the next delete state and
    d) Forward transition to itself.
  - Each delete state has the following five transition probabilities
    a) Forward transition to the next match state with first label,
    b) Forward transition to the next match state with second label,
    c) Forward transition to the next insert state in its layer with the first label,
    d) Forward transition to the next insert state in its layer with the second label,
    e) Forward transition to the next delete state.

Therefore the total number of transition probabilities in the model are $21 \times n+1$.

- At each match and insert state, there are two emission probabilities, one corresponding to amino acid denoted by ϕ and the other for class label a or b called membership probability and denoted by Ψ. Therefore the total emission probability for any state I will be:

$$E_i(\text{amino acid}, \text{label}) = \phi_i(\text{amino acid}) \times \psi_i(\text{label}).$$

We notice that membership probability for state I with class labeled a is: $\Psi_i(a) = 1$ and $\Psi_i(b) = 0$ and the contrary for class labeled b. Delete states have no emissions probability.

RESULTS AND DISCUSSION

In this section, the conditional maximum likelihood estimation method was tested on the model we constructed previously. Our data sets collected here are the following three protein families (*L-asparaginase*, *Micrococcus lutes* K-3type glutamines from *A. orayzae*, *D-aminoacylase* (Georgia and Labrou, 2005; Masuo and Yoshimune, 2004; Yoshimune *et al*., 2005). All of these subsets (protein families) have approximately equal size which is almost 40 amino acid sequences. The tests were done using 3 fold cross validations. The model was estimated using 2 subsets and tested on the last one.

The experiment main steps are the following:

Step 1: The model was estimated in usual way using Baum Welch algorithm to satisfy ML and the performance tested using viterbi algorithm, the results are shown in Fig. 4 and second column in Table 1.

Step 2: To speed up our training, from the ML estimated model in the previous step the training will be continued using the extended Baum Welch algorithm to Satisfy CML and the performance tested also using viterbi algorithm, the results of this step are shown in Fig. 5 and the third column of Table 1.

The following ten numbers are calculated and taken into account for valuating the performance:

- **Amino acid sensitivity:** The percentage of amino acids in alpha-helix regions that are correctly predicted as alpha-helix (i.e., labeled with a).
- **Amino acid determination:** The percentage of amino acids in alpha-helix regions predicted as alpha-helix (i.e., labeled with a) that actually alpha-helix.
- **Alpha-helix sensitivity:** The percentage of alpha-helix regions that are correctly predicted.
- **Alpha-helix determination:** The percentage of predicted alpha-helix regions that are correct.
- **Missing alpha-helix:** The number of real alpha-helix regions that are not overlapping a Predicted one.
- **Wrong alpha-helix:** The number of predicted alpha-helix regions that are not overlapping a real one.
- **Beta-sheet sensitivity:** The percentage of Beta-Sheet regions that are correctly predicted.
- **Beta-sheet determination:** The percentage of predicted Beta-Sheet regions that are correct.
- **Missing beta-sheet:** The number of real Beta-Sheet regions that are not overlapping a predicted one.
- **Wrong beta-sheet:** Number of predicted Beta-Sheet regions that are not overlapping a real one.

Table 1: Performance of our CHMM for protein modeling trained by ML and CML methods

Performance in %	ML estimate	CML estimate
Amino acid sensitivity	73	89
Amino acid determination	85	82
Alpha-helix sensitivity	56	62
Alpha-helix determination	62	71
Missing alpha-helix	18	5
Wrong alpha-helix	17	4
Beta-sheet sensitivity	53	60
Beta-sheet determination	68	75
Wrong beta-sheet	18	5
Missing beta-sheet	15	7

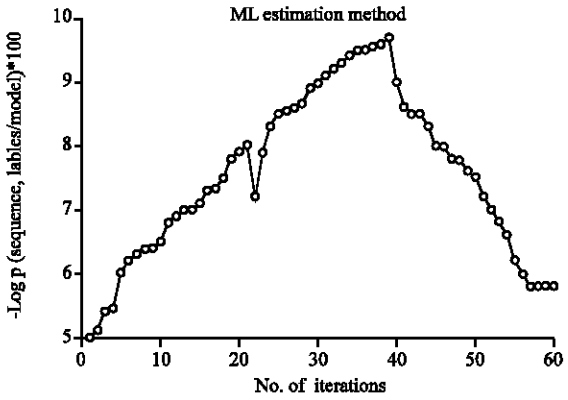

Fig. 4: Represent the score of model parameters at each iteration in ML estimation case

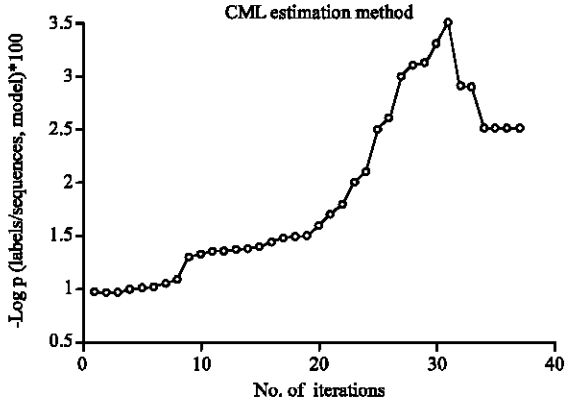

Fig. 5: Represent the score of model parameters at each iteration in CML estimation case

From Table 1, a great significant improvement is observed especially in missing and wrong alpha-helix and missing and wrong beta-sheets also in their sensitivity and determinations. Note that the decrease in the amino acid determination doesn't always mean a worse prediction, because if we assume that there are for example 500 alpha-helixes in the tested sets and only one predicted correctly then the determination is correct.

CONCLUSIONS

PHMM has been used for protein family identification. In this paper we proposed a method to improve the PHMM identification to protein family. The performance of PHMM in the prediction and classification process and multiple sequences alignment also improved by using our method. This method includes using CHMM which requires labeling and estimate parameters (i.e., training model) that satisfy CML which is more accurate than ML. The CML estimation designed to optimize prediction accuracy which obviously shown in our results, where the missing and wrong numbers of alpha-helix and beta-sheets decreased and their determination and sensitivity increased.

REFERENCES

Anders, K., M. Brown and I.S. Mian, 1993. Hidden Markov model in computational biology: Application to protein modeling. J. Mol. Biol., 235: 1501-1531.

Anders, K., 1994. Hidden Markov Model for Labeled Sequences. Proceedings of the 12th LAPR International Conference on Pattern Recognition. IEEE Computer Society Press, California, pp: 140-144.

Georgia, A.K. and N.E. Labrou, 2005. Cloning, expression and characterization of Erwinia Carotovora L-asparaginase. J. Biotechnol., 119: 309-323.

Masuo, N. and K. Yoshimune, 2004. Molecular cloning, over expression and purification of *Micrococcus luteus* K-3-type glutaminase from *Aspergillus oryzae* RIB40. J. Protein Expression Purification, 38: 272-278.

Normandin, Y. and S. Morgan, 1991. An Improved MMIE Training Algorithm for Speaker-Independent, Small Vocabulary, Continuous Speech Recognition. In: Proc. ICASSP'91, Toronto, pp: 537-540.

Rabinar, J., 1991. Hidden Markov models for speech recognition. J. Technometices, 338: 251-272.

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

Riis and A. Krogh, 1997. Hidden neural networks: A framework for HMM/NN hybrids. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, New York, USA., pp: 3233-3236.

Yoshimune, K., N. Esaki and M. Moriguchi, 2005. Site-directed mutagenesis alter DnaK-dependent folding process. J. Biochem. Biophys. Res. Commun., 326: 74-78.