Cai, Jinhai (2008) *Enhanced HMM for the Recognition of Sigma70 Promoters in Escherichia coli.* In: Digital Image Computing: Techniques and Applications (DICTA) 2008, 1-3 Dec 2008, Canberra, Australia.

Enhanced HMM for the Recognition of Sigma70 Promoters in *Escherichia coli*

Jinhai Cai
School of Engineering Systems
Cooperative Research Centre for Spatial Information
Queensland University of Technology
Brisbane, QLD 4001, Australia
j.cai@ieee.org or j.cai@qut.edu.au

Abstract

In this paper, we propose an enhanced HMM for the recognition of sigma70 promoters in E. coli. HMMs for -10 and -35 boxes have been proposed to model the positional dependency of motifs which is lost in methods based on weight matrices. We also propose to use a set of spacer states sharing the observation densities to achieve the desired spacer duration probability functions. We have conducted two sets of experiments on recognizing promoters and locating DNA binding sites and the proposed method has achieved very promising results in comparison with earlier neural network approaches.

1 Introduction

Bioinformatics is an evolving interdisciplinary field of great importance to medical and biological researchers. Recently, techniques in signal processing and pattern recognition have been widely applied in gene sequence alignment and promoter recognition. Promoters help to regulate gene transcription and their identification is crucial if expression of individual genes and gene operons is to be understood [1, 2].

In the past, many approaches have been developed to search for protein binding sites, including methods based on consensus sequences [3], regular expressions [4], weight matrices [5], neural networks [6], and hidden Markov models [7]. The consensus sequence [3] is a concise and powerful representation of all sequences of a family and is easily displayed by sequence logo. However, this representation causes significant information loss. Regular expressions [4] provide another representation, which is easily understood by the human reader. However, the approach may be too rigid to evaluate partial matched sequences and and may involve the loss of important statistical information. However, the most significant disadvantage of using regular expression for promoter identification is that there is no reliable method to obtain a suitable regular expression - the choice being very sensitive to sequence alignment. The weight matrix [5] (also called the position specific scoring matrix) is a matrix of scores where each score is determined for every possible sequence character at a given position. In this approach, the position of a character in a motif sequence is independent. As weight matrix retains adequate 0th order statistical information, it covers all regular expression type signatures and also is suitable for evaluating partial matched motif sequences. Consequently, the weight matrix has been more widely used in applications [9]. However, the approach is not without its problems: 1) due to the Markov chain assumption, information on the correlation between character positions in a sequence is lost; and 2) information on the gap between motifs can not be directly modelled by the weight matrix. Horton and Kanehisa[6] used a neural network (perceptron model) for prediction of E. coli promoter sites. This perceptron model for locating *E. coli* promoters can be viewed as an enhanced version of the weight matrix, but its computational efficiency is not as high as that of weight matrix. A hidden Markov model (HMM) is a stochastic finite state automation. Hidden Markov model (HMM) has been widely used in automatic speech recognition. HMM provides a powerful statistical framework for modeling statistics of input signals. HMMs are especially suitable for modelling sequential signals such as speech signals, financial data, and gene sequences, etc. The profile HMM[7],[10] has been widely exploited in computational biology, although its efficacy is limited as the length of the training sequences increases. However, the HMM does have the advantage of being trainable directly on unaligned sequences, and readily supporting graded scoring of partial matches. In contrast to the weight matrix methods and conventional HMMs, the proposed HMM provides the additional facility that gaps can be handled systematically.

While the HMMs approach has previously been em-

978-0-7695-3456-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DICTA.2008.79

IEEE computer society

ployed in the recognition of sigma70 promoters in *Escherichia coli*, this paper provides significant improvements as a result of a substantially new model architecture. HMM models for DNA-binding sites are usually based on two 6-state boxes, which are equivalent to weight-matrices with systematically modelled gaps. As a result, statistical information is inevitably lost. The new architecture introduced by this paper directly addresses this information loss.

2 HMM for *E. coli* sigma70 promoter recognition

2.1 Basics of the HMM

A DNA sequence is regarded as a time-varying stochastic process. In a short segment, the signal can be considered as a stationary process with minor fluctuations. Therefore, we may use individual states of the HMM to model the steady statistical information for a short segment. Significant changes of properties over the whole input sequence can be modeled by state transitions. A stochastic process is a random process, which evolve over time with probabilities measured on the space of paths. Let X be a random process, where $X = \{x_k\}$ and $k = 1, 2, \cdots$. If X is an Hth order causal Markov process, then we have

$$Pr(x_k|x_{k-1}, \cdots, x_{k-G}) = Pr(x_k|x_{k-1}, \cdots, x_{k-H}), \quad (1)$$

where $G \geq H$. It is very interesting to note the special case when $H = 1$:

$$Pr(x_k|x_1, x_2, \cdots, x_{k-1}) = Pr(x_k|x_{k-1}). \quad (2)$$

This is a very useful property which allows us to use HMM for promoter recognition if we regard x_k as a state and letr the gene sequence position take the role of time.

Generally, an HMM with N states. To formally describe the HMM for promoter recognition, we use the following notation:

- $\mathbf{S}$ is a set of states $\{s_i\}$, where s_i denotes state i.
- $\mathbf{A}$ is a matrix of state transition probabilities $\{a_{i,j}\}$, where $a_{i,j}$=$Pr(s_j|s_i)$ is the probability of transition from state i to state j.
- $\mathbf{B}$=$\{b_i(y)\}$ is an array of the output probability density functions, where $b_i(y) = Pr(y|s_i)$ is the probability for state i to emit symbol y; here $y = A, C, G, orT$.
- π=$\{\pi_i\}$ is the initial state distribution, where $\pi_i = Pr(s_i)$ at the starting time (or sequence position).

For convenience, we represent the model using the compact notation $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$.

Given the definition of HMMs, we must consider the following two problems:

Recognition: Given the observation sequence $\mathbf{O}$ and the models λ, how can we find the model that can generate the observation sequence with the highest probability?

Training: Given a set of observation sequences and the initial model, how can we adjust the model parameters to best account for the given observation sequences?

Recognition based on HMMs

For statistical or Bayesian paradigms, the decision-making is based on the concept of the maximum *a posteriori* probability, $Pr(\mathbf{O}|\lambda)$ where $Pr(\mathbf{O}|\lambda)$ is the conditional probability of the observation sequence $\mathbf{O}$ for the given model λ. The problem of recognition now is how to calculate $Pr(\mathbf{O}|\lambda)$, which can be rewritten as

$$Pr(\mathbf{O}|\lambda) = \sum_{\text{all } \mathbf{S}} Pr(\mathbf{O}, \mathbf{S}|\lambda), \quad (3)$$

where $\mathbf{S}$ is a state sequence of HMM,

$$Pr(\mathbf{O}, \mathbf{S}|\lambda) = Pr(\mathbf{S}|\lambda)Pr(\mathbf{O}|\mathbf{S}, \lambda) = \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t),$$

and $a_{s_0 s_1} = \pi_{s_1}$.

The Viterbi algorithm

The amount of direct computation of $Pr(\mathbf{O}|\lambda)$, which is of the order of $O(TN^T)$ [12], is clearly infeasible in practice. Fortunately, there are two very efficient algorithms, namely, the Baum-Welch algorithm [12] and the Viterbi algorithm [13], available to calculate $Pr(\mathbf{O}|\lambda)$. In our system, we adopted the Viterbi algorithm due to the following reasons:

1. The state sequence of HMM is not observable. However, the single best state sequence can be found by the Viterbi algorithm. This is particularly useful in estimating state duration probabilities.
2. The Viterbi algorithm makes it possible to replace the multiplication in probability computation with summations. As a consequence, the scaling procedure, which is necessary in Baum-Welch algorithm, can be avoided. This results in better computational performance.

For a given model, the standard Viterbi algorithm used in score evaluation is:

Step 1: Initialization.
$\alpha_1(i) = \pi_i b_i(O_1),$
$\Phi_1(i) = 0 \qquad 0 \leq i \leq N - 1.$

Step 2: Recursion.

$$\alpha_t(j) = \max_i[\alpha_{t-1}(i)a_{ij}b_j(O_t)],$$
$$\Phi_t(j) = \arg\max_i[\alpha_{t-1}(i)a_{ij}] \qquad 2 \leq t \leq T.$$

Step 3: Termination. (S_F is the final state set.)

$$Pr^*(\mathbf{O}|\lambda) = \max_{s \in S_F}[\alpha_T(s)],$$
$$s_T = \arg\max_{s \in S_F}[\alpha_T(s)].$$

Step 4: State path backtracking.

$$s_t = \Phi_{t+1}(s_{t+1}).$$

In the recognition phase, the state-path backtracking and the indices $\Phi_t(j)$ are not needed in conventional HMM-based methods. However they are necessary in the training phase and in modeling structures, so we introduce them here. Note, the Viterbi algorithm efficiently finds the maximum of $Pr(\mathbf{O}, \mathbf{S}|\lambda)$ denoted by $Pr^*(\mathbf{O}|\lambda)$.

2.2 HMM for *E. coli* sigma70 promoter recognition

In order to illustrate the need for a new architecture for our application, we first introduce the major conserved features in *E. coli* sigma70 promoter sequences.

There are three major conserved features in *E. coli* sigma70 promoter sequences: the ”-10 box”, the ”-35 box” and the transcription initiation site. The -10 and -35 boxes are conserved hexanucleotide elements that are named according to the approximate position of their central nucleotides relative to the transcriptional start site (TSS) [10]. The well known consensus sequences for the -10 and -35 boxes are TATAAT and TTGACA, respectively. At the initiation site, a pyrimidine (C or T) is followed by a purine (A or G). In order to model these major conserved features, we design an HMM as illustrated in Figure 1, where a circle stands for the state of an HMM and a rectangle stands for a set of states of an sub-HMM. One of the most important merits of using HMM is that we can put all available information into one framework. In this model, there are two sub-HMMs - for the -10 and -35 boxes, respectively, two sub-HMMs for the spacers between -35 and -10 boxes and between -10 box and TSS, respectively, and finally one sub-HMM for elements at the initiation sites. Usually, the spacer between the -10 and -35 boxes consists of 13 to 21 nucleotides but the most likely value is 17. The spacer between the -10 box and the TSS may consist of 3 to 12 nucleotides but is most likely to occupy 6 or 7 nucleotides.

HMMs for -10 and -35 boxes

In traditional methods, an HMM with a set of six left-to-right states is used to model the single hexamer box [10]. This approach is equivalent to the position weight matrix approach, with the same deficiency in that information on the correlation between neighboring nucleotides is lost. In order to improve the performance of HMM-based methods, a ”super-HMM” was proposed [10]. The idea of the super-HMM is to discover sub-classes of a promoter family. A super-HMM consists of several basic sub-HMMs in parallel with each sub-HMM representing a single sub-class. For a given sequence, the optimal path goes through only one sub-HMM. However each sub-HMM retains only 0-order statistical information, therefore sub-HMMs can not solve the problems identified in traditional HMMs. Moreover, the equivalent number of training sequences for each sub-HMM is much smaller than the total number of training sequences. It is well known that a substantial number of training sequences is required to obtain a properly trained HMM, while the available number of sigma70 promoter sequences is limited. As a consequence, the super-HMM can not produce better results than that using one simple HMM with six states if there are only a small number of training sequences available.

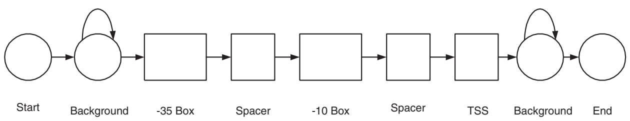

Figure 1. HMM for sigma70 promoter recognition

We propose a new HMM architecture with a lattice structure as illustrated in Figure 2. For a given state, transitions are permitted to any of the next few states, thereby allowing us to model the bigram statistical information. The structure of a trained lattice HMM is thus determined by the state transition probability distribution and both traditional HMMs and super-HMMs are special cases of our lattice-HMM.

However, there is one issue on applying lattice-HMM for promoter recognition: how to determine a suitable initial model of the lattice-HMM. Random initialization of a super-HMM is likely to result in the promoter sequences being classified as a single sub-class, associated with one of the two sub-HMMs [10] due to the reinforcement of each competitive learning iteration. It is well known that the re-estimation procedure in HMMs will lead the initial model to a local optimum. As a consequence of random initialization, the local optimum may be far away from the global optimum.

In this paper, we divide the training procedure into three steps. We begin with the initialisation of the lattice-HMM for the -35 box, which requires us to set non-zero elements in the state transition matrix to a uniform value and set

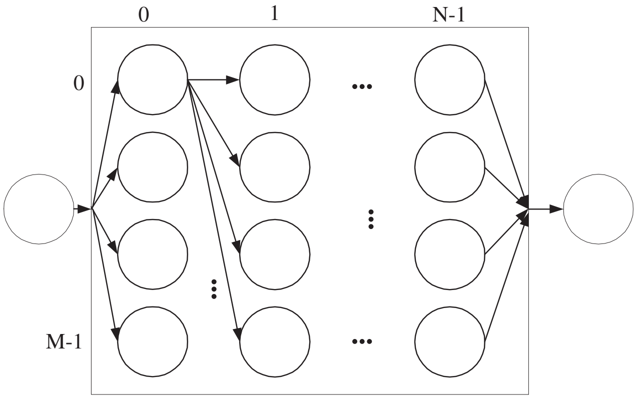

Figure 2. New HMM architecture: Lattice-HMM

the observation symbol probability distribution to a uniform value with small random distortion. The second step is to refine the initial model of the lattice-HMM. We use DNA-binding site matrices [8] instead of promoter sequences to refine the initial model. As the random distortion may result in a good or poor initial model by chance, we select the initial model with the maximum output probability from many repetitions of the second step. The last step is to train the whole HMM by Viterbi Algorithm using promoter sequences.

Spacer states

The modelling of state duration is one major weakness of the conventional HMMs. The inherent duration probability density $Pr_i(d)$ associated with state i can expressed as follows [12]:

$$Pr_i(d) = a_{i,i}^{d-1}(1 - a_{i,i}), \tag{4}$$

where d is the duration. With this model, the probability of the state duration decreases exponentially with its duration. Obviously, this exponential distribution of state duration is inappropriate for most physical signals. Several methods have been proposed to solve this problem [12], however, they require more than many times computation required in the standard HMMs. Vaseghi [14] proposed to a state distribution-dependent transition probability to cope with this problem. The major advantage of Vaseghi's method is that it does not increase the computational overhead. As the maximal duration of background states is unknown in our application, we have to treat background states and spacer states differently if the Vaseghi's method is used. This will increase the complexity of the HMM. We propose a simple solution to this problem, which is illustrated in Figure 3. If the maximum duration of the spacer states is $M - L + 1$ and the minimum duration of the spacer states is $M - N + 2$, the duration probability density $Pr_{spacer}(d)$ is given by:

$$Pr_{spacer}(d) = a_{L,M+2-d}. \tag{5}$$

Now, we can treat all states in the same way as in conventional HMM and we can obtain the desired duration probability density functions. The only drawback is that the proposed method needs more states than other methods to achieve its goal. However, this problem can be lessened if all spacer states share the same observation probability distribution.Form eqn.(5) you can find that it is really easy to initialize $a_{i,j}$ for spacer states with prior information on $Pr_{spacer}(d)$, such as 90to 19 nucleotides.

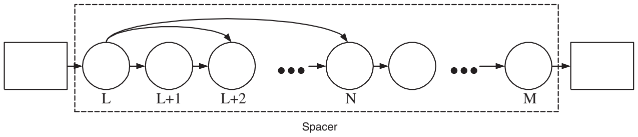

Figure 3. The spacer states

2.3 Dealing with the small number of training sequences

In speech recognition, the number of samples for a particular word is adequate for the researcher obtain a reasonable model of the word. However, the number of gene sequences for a particular category is usually small. This problem is immediately apparent if we apply neural network techniques or statistical methods in bioinformatics. For example, once $a_{i,j}$ becomes 0 in one training iteration - often due to a small number of training samples - it remains as 0 forever regardless of the number of training iterations conducted. As a result, the trained model - while locally optimal - may perform poorly on unseen data. The solution to this problem is to control the rate of convergence:

$$\lambda = (1 - \alpha)\lambda_{pre} + \alpha\lambda_v, \tag{6}$$

$$\lambda_{pre} = \lambda, \tag{7}$$

where α is a control constant, λ is the currently trained model, λ_{pre} is the previously trained model and λ_v is the currently trained model using the Viterbi algorithm. It is easy to observe from eqn.(6) that $a_{i,j}$ in λ can be not equal to 0 even if it is equal to 0 in λ_v.

3 Experimental Results

3.1 Promoter Recognition

In this experiment, our task is to determine whether a given DNA sequence is a sigma70 promoter. We used two sets of DNA sequences: true sigma70 promoter test sequences and computer generated DNA sequences with the same bigram statistical information as the true sigma70 promoter sequences. To evaluate the performance of the proposed algorithm, three measures were used: precision, specificity and sensitivity. These are repectively defined as [15]

$$\begin{aligned} Precision &= \frac{N_c}{N_t} * 100\%, \\ Specificity &= (1 - \frac{N_{fp}}{N_{ng}}) * 100\%, \\ Sensitivity &= \frac{N_{tp}}{N_{po}} * 100\%, \end{aligned} \tag{8}$$

where N_c is the number of test sequences classified correctly, N_t is the total number of test sequences, N_{fp} is the number of false positives, N_{ng} is the total number of negative test sequences, N_{tp} is the number of positive test sequences classified correctly and N_{po} is the total number of positive test sequences. The performance of the proposed method on promoter recognition is given in Table 1. The performance of the proposed method can be further improved by incorporating weak motifs, combining scores of top candidates instead of the best one [11] and combining classifiers [15].

Table 1. The performance of the proposed method

Precision	83.2%
Specificity	84.2%
Sensitivity	81.2%

3.2 Locating binding sites

For given sigma70 promoter sequences, it is desirable to locate DNA binding sites. However, However, only a limited subset of putative binding sites have been confirmed through laboratory experiment, making evaluation difficult. In this experiment, we replaced nucleotides upstream of TSS with random signals, which have the same bigram statistical information as the true sigma70 promoter sequences. Then we insert DNA binding site matrices from [7] into the sequences with the spacers between -10 boxes and the TSS varied from 3 to 12 nucleotides. The experiment showed that the proposed method can achieve an accuracy of 93.2% on locating these binding sites. Figure 4 shows some results, where the located binding sites are in blue color, the true binding sites are underlined and the error is highlighted in yellow color. This result is very encouraging.

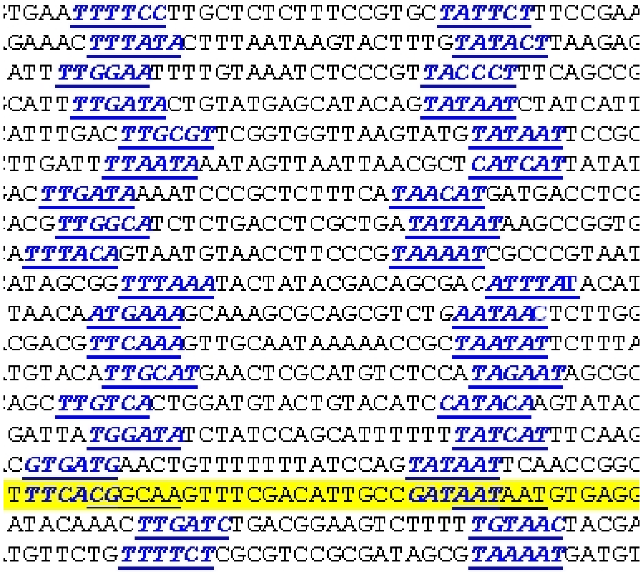
TGAATTTTCCTTGCTCTCTTTCCGTGCTATTCTTTCCGAA
GAAACTTTATACTTTAATAAGTACTTTGTATACTTAAGAG
ATTTTGGAATTTTGTAAATCTCCCGTTACCCTTTCAGCCG
CATTTTGATACTGTATGAGCATACAGTATAATCTATCATT
ATTTGACTTGCGTTCGGTGGTTAAGTATGTATAATTCCGC
TTGATTTTAATAAATAGTTAATTAACGCTCATCATTATAT
ACTTGATAAAATCCCGCTCTTTCATAACATGATGACCTCG
ACGTTGGCATCTCTGACCTCGCTGATATAATAAGCCGGTG
ATTTACAGTAATGTAACCTTCCCGTAAAATCGCCCGTAAT
ATAGCGGTTTAAATACTATACGACAGCGACATTTATACAT
TAACAATGAAAGCAAAGCGCAGCGTCTGAATAACTCTTGG
CGACGTTCAAAGTTGCAATAAAAACCGCTAATATTCTTTA
TGTACATTGCATGAACTCGCATGTCTCCATAGAATAGCGC
AGCTTGTCACTGGATGTACTGTACATCCATACAAGTATAC
GATTATGGATATCTATCCAGCATTTTTTTATCATTTCAAG
CGTGATGAACTGTTTTTTTATCCAGTATAATTCAACCGGC
TTTCACGGCAAGTTTCGACATTGCCGATAATAATGTGAGG
ATACAAACTTGATCTGACGGAAGTCTTTTTGTAACTACGA
TGTTCTGTTTTCTCGCGTCCGCGATAGCGTAAAATGATGT

Figure 4. Locating binding sites.

4 Conclusion

In this paper, we have proposed an enhanced HMM for recognizing sigma70 *E. coli* promoters and locating DNA binding sites. We have proposed the lattice-HMM to deal with the bigram information loss. We have also proposed to use a set of spacer states instead of one spacer state to achieve desired spacer duration probability functions. Our initial experiments have produced very encouraging results, and in subsequent work we will examine its application to binding site identification across bacterial species.

References

[1] T.S. Rani, S. D. Bhavani and R. S. Bapi, “Analysis of E.coli promoter recognition problem in dinucleotide feature space”, Bioinformatics, Vol.23, No.5, pp.582-588, 2007.

[2] A.M. Huerta and J. Collado-Vides, “Sigma70 Promoters in *Escherichia coli*: Specific Transcription in Dense Regions of Overlapping Promoter-like Signals”, J. Molec. Biol., Vol.333, pp.261-278, 2003.

[3] T.D. Schneider and R.M. Stephens, “Sequence Logos: A New Way to Display Consensus Sequence”, Nucleic Acids Res. Vol.18, pp.6097-6100, 1990.

[4] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C.J. Sigrist, K. Hofmann, and A. Bairoch, “The PROSITE database, its status in 2002”, Nucleic Acids Res. Vol.30, pp.235-238, 2002.

[5] P. Bucher, “Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences”, J.Molec.Biol. Vol.212, pp.563-578, 1990.

[6] P.B. Horton and M. Kanehisa, “An Assessment of Neural Network and Statistical Approaches for Prediction of E. coli Promoter Sites”, Nucleic Acids Res. Vol.20, pp.4331-4338, 1992.

[7] A. Krogh, M. Brown, I.S. Mian, K. Sjölander and D. Haussler, “Hidden Markov Models in Computational Biology: Applications to Protein Modeling”, Journal of Molecular Biology, Vol.235, pp.1501-1531, 1994.

[8] K. Robison, A.M. McGuire and G.M. Church, “A Comprehensive Library of DNA-binding Site Matrices for 55 Proteins Applied to the Complete *Escherichia coli* K-12 Genome”, J. Mol. Biol., Vol.284, pp.241-254, 1998.

[9] E. Birney, “Hidden Markov Models in Biological Sequence Analysis”, IBM Journal of Research and Development Vol.45, No.3/4, 2001.

[10] A.G. Pedersen, P. Baldi, S. Brunak and Y. Chauvin, “Characterization of Prokaryotic and Eukaryotic Promoters Using Hidden Markov Models”, Proc. of the 4th International Conference on Intelligent Systems for Molecular Biology, pp.182-191, 1996.

[11] A. Krogh, “Two Methods for Improving Performance of An HMM and Their Application for Gene Finding”, Proc. of the 5th International Conference on Intelligent Systems for Molecular Biology, pp.179-186, 1997.

[12] L.R. Rabiner, “A tutorial on hidden Markov model and selected applications in speech recognition,” *Proceedings of the IEEE*, Vol.77, pp.257-286, 1989.

[13] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Transactions On Information Theory, Vol.13, No.2, pp.260-269, April 1967.

[14] S.V. Vaseghi, “State duration modeling in hidden Markov models,” *Signal Processing*, Vol.42, pp.31-41, 1995.

[15] Q. Ma, J.T.L. Wang, and J.R. Gattiker, ”Mining Biomolecular Data Using Background Knowledge and Artificial Neural Networks”, 2002.