LPB: A New Decoding Algorithm for Improving the Performance of an HMM in Gene Finding Application

Ahmed M.Khedr, Mohamed H. Ibrahim, and Amal Al Ali

***Abstract*—Hidden Markov models (HMMs) are applied to many problems of computational Molecular Biology. In a predictive task, the HMM is endowed with a decoding algorithm in order to assign the most probable path of states, and in turn the class labelling, to an unknown sequence. In this paper we introduce a novel decoding algorithm Log-posterior-best (LPB) which combines the log-odd posterior probability and 1-best algorithms. LPB is a two steps process: first the Log odd probability of each state is computed and then the best allowed label path through the model is evaluated by a 1-best algorithm. We show that our LPB decoding performs better than other existing algorithms in some computational biological problems such as gene finding in prokaryotes.**

***Index Terms*—Hidden Markov Model; LPB decoding algorithm; DNA sequences.**

I. Introduction

A HIDDEN Markov Model consists of two stochastic processes. The first stochastic process is a Markov chain that is characterized by states and transition probabilities [1]. The states of the chain are externally not visible, therefore ”hidden”. Another stochastic process will generate emissions which is observable at every instant. It is dependent on state probability distribution [2]. In case of Hidden Markov Model, the term “hidden” doesn't indicate a parameter of the model, but it indicates the state of the Markov Chain [3], [4], [5], [6].

A Hidden Markov Model is a generalization of a Markov chain, in which each ”internal” state is not directly observable (hence the term hidden) but produces ”emits” an observable random output ”external” state, also called “emission”, according to a given stationary probability law [8]. In this case, the time evolution of the internal states can be induced only through the sequence of the observed output states.

If the number of internal states is N, the transition probability law is described by a matrix with N times N values; if the number of emissions is M, the emission probability law is described by a matrix with N times M values. A model is considered defined once given these two matrices and the initial distribution of the internal states. The study by Rabiner [9], [10] is widely well appreciated for clarity in explaining HMMs. It is a powerful class of model used in many fields including gene finding, profile searches, error correction coding, multiple sequence alignment, speech recognition and regulatory site identification.

One of the important goals is to find genes in a genome sequences of prokaryote organisms. Finding or predicting a gene in prokaryote is needed for knowing the function, activity, and characteristics of cells. Also finding genes will provide an important way for identifying diseases that could have occurredin animals, plants and humans, and aid scientists to discover a useful drugs.

Gene Finding: The term ”gene finding” indicates the action of finding genes within a DNA sequence, but is often used with a more general meaning of labeling DNA tracts [11], for example labeling them as coding, intergenic, introns, etc. In this last sense gene finding can be considered a special case (the most important in bioinformatics) of the more general action known as sequence labeling (also for non-DNA sequences). Determining the DNA and RNA sequences are relatively cheaper than determining protein sequences. One of the most successful class of techniques for analyzing biological sequences has been the use of HMM. HMM are mainly used for predicting protein sequences since it provides a good probabilistic framework for the same [12], [25]. Considering the work done, there are mainly two tasks for bioinformatics: Deducing the protein DNA sequence and Comparing protein sequences to an existing database of protein sequences, both of which utilize Hidden Markov Models. In [13], the authors introduced the use of HMMs for discriminating coding and intergenic regions in E. coli genome. The program GeneMark [19] finds genes in E. coli DNA using a Markov model for the coding region. It is a parser with a complex intergenic model. The more complex HMM, intergenic model consists of several parts in addition to the start and stop codon models. After generating the stop codon, the model chooses either the transition to the long intergenic HMM or the short intergenic HMM, with appropriate probabilities. The short intergenic HMM tends to generate intergenic regions of lengths from 1 to 14 or so, with statistics determined from examples of such short intergenic regions in actual E. coli contigs. Similarly, the parameters of the long intergenic model are adjusted to capture the statistics of longer intergenic regions. The parameters of the two intergenic models were estimated from a set of known intergenic regions by a learning procedure known as the forward-backward algorithm. As a result of the training process, the long intergenic region develops patterns, without having to explicitly encode them. The complex parser has a better accuracy.

Generally, there are two principal methods for finding

Manuscript received Feb. 21, 2020; revised July 03, 2020.

Professor Ahmed M. Khedr is with Department of Computer Science, College of Computing & Informatics, University of Sharjah, Sharjah 27272, UAE, Email: akhedr@sharjah.ac.ae.

Dr Mohamed H. Ibrahim is with Department of Mathematics, Zagazig University, Zagazig, Egypt

Dr Amal Al Ali is with department of information system, College of Computing & Informatics, University of sharjah, Sharjah 27272, UAE, Email: aialali@sharjah.ac.ae

genes in prokaryotes (e.g, E. coli), both of which have been incorporated into systems that analyse eucaryotic DNA. The first approach locates signals in DNA like promoter sequences and splice junctions using techniques such as neural networks [25], [26] or statistical methods [27]. The second approach scores a certain window of DNA in various ways in order to decide whether the window belongs to a coding (*where coding regions represent a gene*) or anon-coding region (reviewed in [29]). Staden and McLachlan [23], [30] proposed deviation from average codon usage as a way of determining the probability that the window is coding or not. Later, Gribskov et al. [31] used a similar measure as a part of their 'codon preference plot', but their measure did not require the knowledge of an average codon usage from other sources. Most other scoring methods are related to codon usage in some way [23]. Recently, the most powerful machine learning techniques such as neural networks [34] and HMM [37] have been used to analyze coding (and non-coding) regions. In particular, the GeneMark Model [19] finds genes in E.coli DNA using a HMM for the coding region related to the one discussed in this paper.

When HMMs are implemented for Gene prediction problem, a decoding algorithm is needed. Decoding phase is the process of finding the most suitable path of states that represent a given sequence and in turn the class labelling, to an unknown sequence.

There are many argued decoding algorithms such as Viterbi, 1-best, posterior. The decoding algorithm takes the observation sequence as input and return the best path of states that represent it in the model. The Viterbi and the Posterior decoding algorithms are the most common algorithms used in decoding. The former is very efficient when one path dominates, while the second, even though does not guarantee to preserve the automaton grammar, is more effective when several concurring paths have similar probabilities. The third alternative is 1-best, which was shown to perform equal or better than Viterbi for getting most probable label path.

II. Related Work

Many extensions to the original "pure" HMM have been developed for gene finding. For example, [14] designed separate HMM modules, each one appropriate for a specific region of DNA. Separate HMM modules were designed and trained for specific regions of DNA: exons, introns, intergenic regions and splice sites. In order to form a biologically viable topology, the models were coupled. Additionally, the integrated HMM was trained on a set of eukaryotic DNA sequences and then tested by using an unknown DNA sequence. [15], [16] used a Generalized HMM (GHMM or "hidden semi-Markov Model") that allows more than one emission for each internal state. Gene finding has many applications regarding to human health [44], [45], [46], [47], [48], [49].

A pair HMM [2] is having large no of similarity with standard HMM. Basic difference is that it emits pairwise alignment, where as standard HMM emits a single sequence. This method provides parse only alignments between two sequences but, with suitable enhancements, it is sometimes applied to gene finding. For example, [17] presented a new method that predicts the gene structure starting from two homologous DNA sequences, identifying the conserved sub-sequences. A useful open-source implementation is described [18]. [19] proposed a new algorithm (GeneMark.hmm) that improves the gene finding performance of the old GeneMark algorithm by means of a suitable coupling with an HMM model. [19] introduced an Evolutionary Hidden Markov Model (EHMM), based on a suitable coupling of an HMM and a set of evolutionary models based on a phylogenetic tree.

In [33], the authors combined all methods for locating or predicting genes (the search for initiation signals, the scoring of possible coding regions, and the final dynamic programming to get the best path of states) in a single simple framework of Hidden Markov Models. The HMM we use in this paper to find genes in E.coli is the one used in Krogh [33]. Since only one strand is modelled, the HMM is applied twice, once to the direct strand and then to the complementary strand.

After the Model is built for E.coli DNA sequences, this Model will be used to predict the genes in E. coli DNA sequences. When performing a prediction task using HMMs, the problem is endowed with a decoding algorithm in order to assign the most probable state path, and in turn the class labelling, to an unknown DNA sequence. The decoding algorithm takes the unknown DNA sequence as input and returns the best path of states that represents it in the model, this path used to identify genes in this DNA sequence. In E.coli gene finding problem discussed here, the Viterbi [33], Posterior, and 1-best decoding algorithms were the most common algorithms used in decoding process.

The Viterbi decoding is very efficient in determining the best allowed path when one path dominates the rest, but the disadvantage of Viterbi is that if two or more paths have the similar probability for the DNA sequence, it lacks ability to choose the best between them. While the Posterior decoding in contrary is more effective when several concurring paths have similar probabilities, but its disadvantage is that it does not guarantee to preserve the automaton grammar, or in other words it may get a path that might be not allowed. The third alternative is 1-best, which is very efficient in determining the best allowed label path when one label path dominates the rest. Labels almost represent a particular signals in DNA sequences. The disadvantage of 1-best is that if there are two or more labels having the similar probability for the DNA sequence, it doesn't have the ability to choose the best between them. 1-best was shown to perform equal or better than Viterbi. This problem can be solved in case the data sets are distributed among a number of sites which will be our future work [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63].

III. Decoding Algorithms

- **Viterbi Decoding:** Viterbi decoding finds the path π through the model which has the maximal probability with respect to all the others [23], [27]. This means that we look for the path which satisfies

$$\pi^v = argmax_\pi P(\pi|O, M) \quad (1)$$

where $O = O_1, \ldots, O_L$ is the observed sequence of length L and M is the trained HMM model. Since the

$P(O|M)$ is independent of a particular path π, Equation 1 is equivalent to:

$$\pi^v = argmax_\pi P(\pi, O|M) \tag{2}$$

where $P(\pi, O|M)$ can be easily computed as:

$$P(\pi, O|M) = (\prod_{i=1}^{L} a_{\pi(i-1),\pi(i)} e_{\pi(i)}(O_i)) a_{\pi(L),End} \tag{3}$$

where by construction $\pi(0)$ always is the Begin state. Defining $v_k(i)$ as the probability of the most likely path ending in state k at position i, and as the trace-back $P_i(k)$ pointer, π^v can be obtained by running the following dynamic programming called Viterbi decoding.

- Initialization:
$$v_{Begin}(0) = 1,\ v_k(0) = 0 \ \ for \ \ k \neq Begin$$
- Recursion:
$$v_k(i) = [\ max_{\{s\}}(v_s(i-1)a_{s,k})]e_k(O_i)$$
$$P_i(k) = argmax_{\{s\}}(v_s(i-1)a_{s,k})$$
- Termination:
$$P(O, \pi^v|M) = max_{(s)}[\ v_s(L)a_{s,End}]$$
$$\pi^v_L = argmax_{(s)}[\ v_s(L)a_{s,End}]$$
- Traceback:
$$\pi^v_{i-1} = P_i(\pi^v_i)$$
- Label assignment:
$$\mu_i = Label(\pi^v_i), \ \ for \ \ i = 1, \ldots, L$$

- **Posterior Decoding:** The Posterior decoding finds the path which maximizes the product of the Posterior probability of the states [23], [27]. Using the usual notation for forward $f_k(i)$ and backward $b_k(i)$ we have

$$P(\pi_i = k|O, M) = f_k(i).b_k(i)/P(O|M) \tag{4}$$

The path π^P which maximizes the Posterior probability is then computed as:

$$\pi^P_i = argmax_{\{s\}}\ P(\pi_i = s|O, M), \quad for \quad i = 1, \ldots, L \tag{5}$$

The corresponding label assignment is:

$$\mu_i = label(\pi^P_i), \quad for \quad i = 1, \ldots, L \tag{6}$$

If we have more than one state sharing the same label, labelling can be improved by summing over the states that share the same label (*Posterior sum*). In this way we can have a path through the model which maximizes the Posterior probability of being in a state with label η when emitting the observed sequence element , or more formally:

$$\mu_i = argmax_{\{\eta\}} \sum_{label(s)=\ \eta} P(\pi_i = s|O, M) \tag{7}$$

The drawback of Posterior-decoding is that the state path sequences π^P_i or μ may be not allowed paths. However, this decoding can perform better than Viterbi, when more than one high probable path exits [23], [27]. In this case a post-processing algorithm that recasts the original topological constraints is recommended [36].
In the sequel, if not differently indicated, with the term Posterior we mean the Posterior sum.

- **1-best Decoding:** The 1-best labelling algorithm described here is the Krogh's previously described variant of the N-best decoding [32]. Since there is no exact algorithm for finding the most probable labelling, 1-best is an approximate algorithm which usually achieves good results in solving this task [33]. Differently from Viterbi, the 1-best algorithm ends when the most probable labelling is computed, so that no trace-back is needed.
For sake of clarity, we define H_i as the set of all labelling hypothesis surviving as 1-best for each state s up to sequence position i. In the worst case the number of distinct labelling-hypothesis is equal to the number of states. h^s_i is the current partial labelling hypothesis associated to the state s from the beginning to the i^{th} sequence position. In general, several states may share the same labelling hypothesis. Finally, we use $\bigoplus$ as the string concatenation operator, so that 'AAAA' $\bigoplus$ 'B'='AAAAB'. 1-best algorithm can then be described as follows:

  - Initialization:
$$v_{Begin}(0) = 1, \quad v_k(0) = 0 \quad for \quad k \neq Begin$$
$$v_k(1) = a_{Begin,k}.e_k(O_i),\ H_1 = \{label(k) : a_{Begin,k} \neq 0\}$$
$$H_i = \emptyset, \quad for \quad i = 2, \ldots, L$$
  - Recursion:
$$v_k(i+1) = max_{(h \in H_i)}[\textstyle\sum_s v_s(i).\gamma(h^s_i, h).a_{s,k}].e_k(O_{i+1})$$
$$h^k_{i+1} = argmax_{(h \in H_i)}[\textstyle\sum_s v_s(i)\gamma(h^s_i, h).a_{s,k}] \bigoplus label(k)$$
$$H_{i+1} \longleftarrow \ H_{i+1} \cup \{h^k_{i+1}\}$$
  - Termination:
$$\mu = argmax_{h \in H_L} \textstyle\sum_s v_s(L).\gamma(h^s_L, h).a_{s,End}$$

where $\gamma(a, b)$=1 when a =b, 0 otherwise, also $a_{s,k} = 0$ when there is no transition between states s and k, none-zero otherwise. We must note that in 1-best decoding we do not need to keep trace-back matrix since it is computed during the forward steps.

Contribution: In this paper, we introduce a novel decoding algorithm Log-Posterior-best(LPB) which combines the log-odd Posterior probability and 1-best algorithms. LPB is a two steps process: first the Log odd probability of each state is computed and then the best allowed label path through the model is evaluated by a 1-best algorithm.

We show that our LPB performs better than other existing algorithms in some computational biological problems such as predicting coding regions in prokaryotic's DNA.

IV. Log-Posterior-Best Decoding Algorithm (LPB)

Most computational molecular biology prediction problems endowed to decoding of most probable path, so it's great to develop in decoding algorithms. Approximately in all prediction problems such as gene prediction, protein structure prediction, protein, gene classifications, and more others include dealing with labels. Labels almost represent a biological properties in DNA or protein sequences. As a

result in the decoding process we aim to find the decoding of most probable label path more than most probable state path.

In this paper, we introduce a novel decoding algorithm called Log-Posterior-best algorithm (abbreviated as LPB). In our LPB we combined 1-best decoding algorithm with Posterior decoding algorithm.

Advantage of LPB algorithm is that it solves the problem in 1-best algorithm when there are several label paths for single unknown sequence by including Posterior probability. Another advantage of LPB algorithm is that it solves the problem of Posterior decoding algorithm by selecting a path that might be not allowed, including 1-best.

Log-Posterior-Best (LPB) algorithm is two process:

1) firstly, we compute the log-odd of Posterior probability for each state at each position in the unknown query sequence.
2) secondly, we apply 1-best decoding algorithm to get the most probable label path.

We must note that, in our LPB, we take log-odd to avoid the problem of underflow, that may occur with very long sequences as we stated above and also compute the Posterior probability in more accurate manner. finally the LPB is compute π^{LPB} performed as follows:

- Initialization:

$$v_{Begin}(0) = 1, \quad v_k(0) = 0 \quad for \quad k \neq Begin$$
$$v_k(1) = a_{Begin,k}.e_k(O_1), \; H_1 = \{label(k) : a_{Begin,k} \neq 0\}$$
$$H_i = \emptyset, \quad for \quad i = 2, \ldots, L$$

- Recursion:

$$v_k(i+1) = max_{(h \in H_i)}[\textstyle\sum_s v_s(i).\gamma(h_i^s, h).a_{s,k}] \cdot logit(P(\pi_i = k|O, M)))$$
$$h_{i+1}^k = argmax_{(h \in H_i)}[\textstyle\sum_s v_s(i)\gamma(h_i^s, h).a_{s,k}] \bigoplus label(k)$$
$$H_{i+1} \longleftarrow H_{i+1} \cup \{h_{i+1}^k\}$$

- Termination:

$$\mu = argmax_{h \in H_L} \textstyle\sum_s v_s(L).\gamma(h_L^s, h) \cdot a_{s,End}$$

where log-odd of Posterior probability is computed as:

$$P(\pi_i = k, O) = P(\pi_i \mid O, M) \times P(O \mid M). \tag{8}$$

$$f_k(i) \times b_k(i) = P(\pi_i \mid O, M) \times P(O \mid M). \tag{9}$$

$$P(\pi_i \mid O, M) = \frac{f_k(i) \times b_k(i)}{P(O \mid M)}. \tag{10}$$

$$logit(P(\pi_i|O, M)) = logit(\frac{f_k(i) \times b_k(i)}{P(O \mid M)}). \tag{11}$$

$$logit(P(\pi_i|O, M)) = logit(f_k(i) \times b_k(i)) - logit(P(O \mid M)). \tag{12}$$

$$logit(P(\pi_i|O, M)) = log(\frac{f_k(i) \times b_k(i)}{1 - (f_k(i) \times b_k(i))}) - log(\frac{P(O \mid M)}{P(O \mid NullModel)}) \tag{13}$$

V. Results and Discussions

A. Data Sets

The data used to score our algorithm LPB with other decoding algorithms (Viterbi, 1-best, Posterior) are:
Some DNA sequences which are available at (*http://www.ncbi.nlm.nih.gov/sites/entrez*) for dUTPase coli, in which we use a set that includes 400 DNA sequences whose length range from 800 bp to 1250 bp. Aligning these DNA sequences show that their conserved region represent less than 25% of their total length, in other words these DNA sequences are less than 25% identical. Among these 400 DNA sequences (300 were used randomly to train the HMM, the other 100 DNA sequences are used as test set). Here our focus is to test the ability of Viterbi, Posterior, 1-best, and LPB decoding algorithms to solve the problem of the coding regions prediction in these DNA sequences.

B. Accuracy's Measures

The measures used here to test the accuracy of predictions will be at *nucleotide level*:
in which each nucleotide of a test sequence is classified as predicted positive (PC) if it is in a predicted coding region, predicted negative (PN) otherwise, and also as actual positive (**AC**) or actual negative (AN) according to annotation of the DNA sequence. These assignments are then compared to calculate the number of true positives (**TC**), false positives (FC), true negatives (TN) and false negatives (FN). Accuracy is then measured by:

$$\begin{aligned} Sensitivity\ (SEN) &= \frac{TC}{TC + FN} \\ Specificity\ (SPE) &= \frac{TC}{TC + FC} \end{aligned} \tag{14}$$

C. Testing

We use 400 DNA sequences with their labels, the results obtained using the four different decoding algorithms (i.e., Viterbi, Posterior, 1-best, LPB) are evaluated, where the performance of the four algorithms are tested on 300 DNA sequences (cross validations), and the latter 100 DNA sequences are used for blind-test to reconstruct the correct labelling. Figures 1 and 2 give the performance of the decoding algorithms in terms of average sensitivity and specificity as obtained in cross-validation and blind test. The results indicate that the proposed LPB scheme provides better performance than the other decoding algorithms. In addition, Figures 3 and 4 illustrate the quality of predictions of the decoding algorithms at the nucleotide level. By combining 1-best decoding algorithm with Posterior decoding algorithm, LPB algorithm solves the limitation of 1-best algorithm that occurs when there are several label paths for single unknown sequence. Moreover, LPB algorithm solves the drawback of Posterior decoding algorithm that selects a path that might be not allowed by including 1-best. When comparing the LPB algorithm with Viterbi, Posterior and 1-Best, it is clear that the LPB provides a great improvement to the prediction and provides better quality prediction. The obtained summary of results is tabulated in Table 1. It is obvious that, the Viterbi decoding and the 1-best are unreliable, since the blind test

is less than 20%. On the other hand, Posterior decoding is more efficient and can correctly assign more than 50 % and 25 % of the DNA sequences, in cross-validation and blind test, respectively. Our proposed LPB decoding has the best performance since its decoding achieving 90 % and 81 % in cross-validation and blind test, respectively. We must note that LPB is perform better than Viterbi Posterior and 1-best because it solves the problem of similar probabilities for two state paths or label paths by including Posterior probability and also solves the problem of selecting not allowed path that occurred in Posterior by including 1-best.

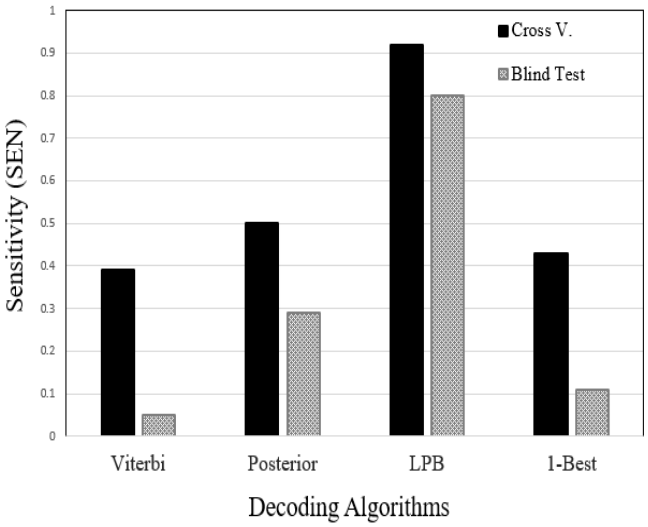

Fig. 1. The performance of the decoding algorithms in terms of average sensitivity.

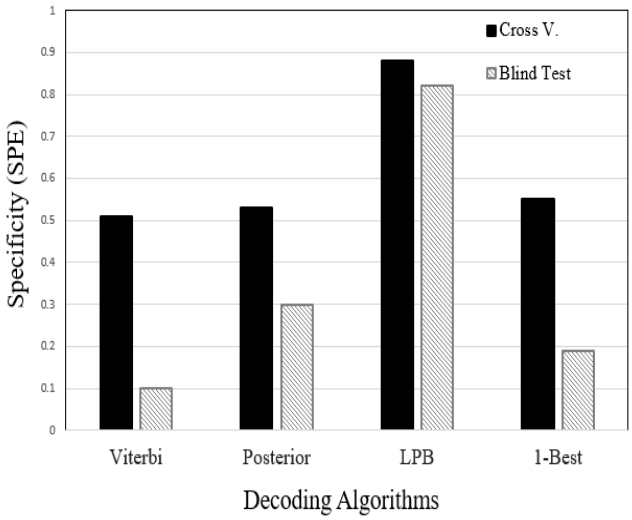

Fig. 2. the performance of the decoding algorithms in terms of average specificity.

We must note that LPB performs better than Viterbi and 1-best because it solves the problem of similar probabilities for two state paths or label paths by including Posterior probability.

Also, it performs better than Posterior because it solve the problem of not allowed path that occurred in Posterior by including 1-best.

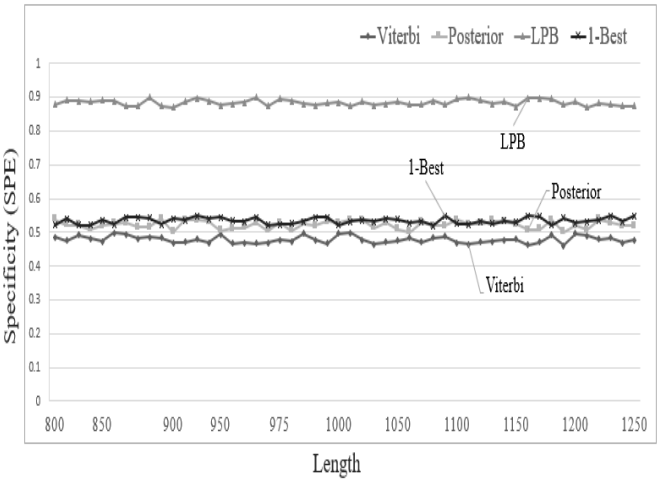

Fig. 3. The quality of predictions of the decoding algorithms at the nucleotide level-Specificity.

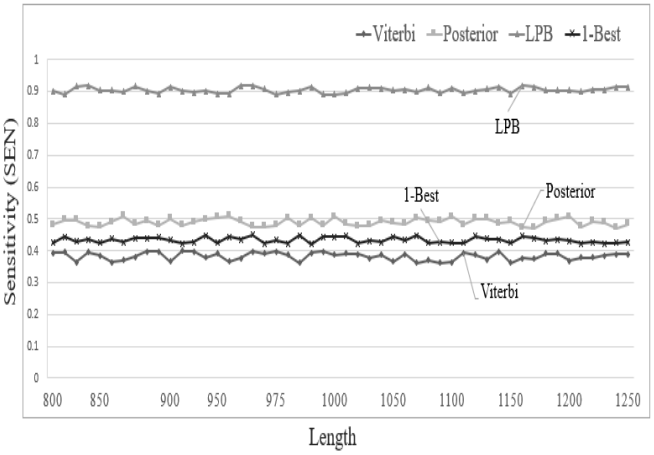

Fig. 4. The quality of predictions of the decoding algorithms at the nucleotide level-Sensitivity.

Table 1: The Performance of the four Decoding Algorithms on DNA sequences

algo. / Meas.	Viterbi		Posterior		1-Best		LPB	
	Cross V.	Blind Test	Cross V.	Blind Test	Cross V.	Blind Test	Cross V.	Blind Test
SEN	0.39	0.05	0.50	0.29	0.43	0.11	0.92	0.80
SPE	0.51	0.10	0.53	0.30	0.55	0.19	0.88	0.82

D. Complexity

- **Time Complexity:** the time complexity for Viterbi, 1-best, and Posterior are $[O(N^2.\ L)]$, although LPB takes a time longer than other algorithms where it combines the time complexity of log-odd Posterior probability $plus$ 1-best algorithm, but its time-complexity still remains the same as other decoding algorithms which is $[O(N^2.\ L)]$.
  where L and N are the DNA sequence length and the number of states respectively.

$$time\ (Viterbi) \leq time\ (Posterior) \leq time\ (1-Best) \leq time\ (LPB) \quad (15)$$

- **Space Complexity:** the space-complexity of Viterbi and Posterior algorithms, while 1-best requires less memory space [33], LPB is the same as 1-best, in which both of LPB and 1-best have less memory space is that there is no tracking back and they depend on length of coding regions rather than total DNA sequences.

$$space\ (1-best) \leq space\ (Viterbi) \leq space\ (Posterior) \leq space\ (LPB) \quad (16)$$

VI. Conclusion

We proposed a new decoding algorithm called Log-Posterior-best (LPB). LPB decoding algorithm is applicable to all the possible HMMs with an arbitrary number of labels, we show that LPB perform better than other decoding algorithms, especially when dealing with biological problems that needs labelling such as gene finding in prokaryote.

References

[1] M. Ahmad, A. Abdullah and K. Buragga, "A novel optimized approach for gene identification in DNA sequences," J. Applied Sci., vol. 11, pp. 806-814, 2011.

[2] I. M. Meyer, and R. Durbin, "Comparative ab initio prediction of gene structures using pair HMMs," Bioinformatics, vol. 18, pp. 1309-1318, 2002.

[3] M. H. El-Sayed and A. M. Khedr, "Improving the performance of an HMM for protein family modelling," J. Applied Sci., vol. 7, pp. 1626-1632, 2007.

[4] M. H. Ibrahim1, and A. M. Khedr, "Leveraging Pruning Techniques for Improving Generalized HMM Decoding in Gene Classification," International Journal of Biomedical Data Mining, vol. 7, no. 1, pp. 1-6, 2018.

[5] A. M. Khedr, "Improving Protein Tertiary Structure Prediction using HMM," Kuwait J. of Sci. and Eng. vol. 38, no. 2A, pp.1-14, 2011.

[6] A. M. Khedr and M. H. Ibrahim, "Log-odd: A New Method for Improving Hidden Markov Model Decoding for Gene Finding," Kuwait J. Sci. vol. 39, no. (2A), pp. 103-118, 2012.

[7] T. T. Tran, F. Zhou, S. Marshburn, M. Stead, S.R. Kushner and Y. Xu, "De novo computational prediction of non-coding RNA genes in prokaryotic genomes," Bioinformatics, vol. 25, pp. 2897-2905, 2009.

[8] A. P. Oron, Z. Jiang and R. Gentleman, "Gene set enrichment analysis using linear models and diagnostics," Bioinformatics, vol. 24, pp. 2586-2591, 2008.

[9] L. Jing, H. Hua and L. Sufang, "Voice identification based on HMMs," Trends Applied Sci. Res., vol. 1, pp. 79-82, 2006.

[10] M. Frikha, Z. B. Messaoud and A. B. Hamida, "Towards discriminative training estimators for HMM speech recognition system," J. Applied Sci., vol. 7, pp. 3891-3899, 2007.

[11] C. Burge, and S. Karlin, "Prediction of complete gene structures in human Genomic DNA," J. Mol. Biol., vol. 268, pp. 78-94, 1997.

[12] D .C. Nath and A. Bhattacharjee, "A bayesian approach for autoregressive models in longitudinal data analysis: An application to type 2 diabetes drug comparison," Asian J. Applied Sci., vol. 4, pp. 640-648, 2011.

[13] A. Krogh, I. S. Mian and D. Haussler, "A hidden Markov model that finds genes in Escherichia coli DNA," Nucleic Acids Res., vol. 22, pp. 4768-4778, 1994.

[14] J. Henderson, S. Salzberg and K. H. Fasman, "Finding genes in DNA with a hidden markov model," J. Comput. Biol., vol. 4, pp. 127-141, 1997.

[15] D. Kulp, D. Haussler, M.G. Reese and F.H. Eeckman, "A generalized hidden Markov model for the recognition of human genes in DNA," Proc. Int. Conf. Intell. Syst. Mol. Biol., vol. 4, pp. 134-142, 1996.

[16] C. B. Burge, and S. Karlin, "Finding the genes in genomic DNA," Curr. Opin. Struct. Biol., vol. 8, pp. 346-354, 1998.

[17] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids," Cambridge University Press. Cambridge, 1998.

[18] W. H. Majoros, M. Pertea and S.L. Salzberg, "Efficient implementation of a generalized pair hidden Markov model for comparative gene finding," Bioinformatics, vol. 21, pp. 1782-1788, 2005.

[19] A. V. Lukashin and M. Borodovsky, "Genemarkhmm: New solutions for gene finding," Nucleic Acids Res., vol. 26, pp. 1107-1115, 1998.

[20] Pedersen, J.S. and J. Hein, "Gene finding with a hidden Markov model of genome structure and evolution," Bioinformatics, vol. 19, pp. 219-227, 2003.

[21] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, Miller,W. and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," Nucleic Acid Research, vol. 25, pp. 3389-3402, 1997.

[22] P. G. Bagos, T. D. Liakopoulos, I. C. Spyropoulos, and S. J. Hamodrakas, "PRED-TMBB: a web server for predicting the topology of beta barrel outer membrane proteins," Nucleic Acids Research, vol. 32, pp. 400-404, 2004.

[23] Baldi,P. and Brunak,S., "Bioinformatics: the Machine Learning Approach," MIT Press, 2001.

[24] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure, "Hidden Markov Models of Biological Primary Sequence Information," PNAS USA, vol.91, pp. 1059-1063, 1994.

[25] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, s. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. Sonnhammer, "The Pfam Protein Families Database," Nucleic Acids Research, vol. 30, pp. 276-280, 2002.

[26] H. R. Bigelow, D. S. Petrey, J. Liu, D. Przybylski, and B. Rost, "Predicting transmembrane beta-barrels in proteomes," Nucleic Acids Research, vol. 32, pp. 2566-2577, 2004.

[27] R. Durbin, S. Eddy, A. Krogh and G. Mitchinson, "Biological sequence analysis: probabilistic models of proteins and nucleic acids," Cambridge Univ. Press, Cambridge, 1998.

[28] I. Holmes, and R. Durbin, "Dynamic programming alignment accuracy," Annual Conference on Research in Computational Molecular Biology, New York, United States, RECOMB vol. 04, pp. 493-504, 1998.

[29] Q. Liu, Y. S. Zhu, Wang, B.H. and Y. X. Li, "A HMM-based method to predict the transmembrane regions of beta-barrel membrane proteins," Computational Biology and Chemistry, vol. 27, no. 1, pp. 69-76, 2003.

[30] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and Haussler, D., "Hidden Markov models in computational biology: Applications to protein modeling," Molecular Biology, vol. 235, pp. 1501-1531, 1994.

[31] A. Krogh, "Hidden Markov models for labeled sequences," In Proceedings 12^{th} International Conference on Pattern Recognition. IEEE Comp. Soc. Press, Singapore, pp. 140-144, 1994.

[32] A. Krogh, "Two mehods for improving the performance of an HMM and teir application for gene finding," In Proc. Fifth Int. Con. Intellegent System of Molecular Biology, pp. 179-186, 1997.

[33] A. Krogh, I. Saira Mian1, and D. Haussler2, "A hidden Markov model that finds genes in E.coli DNA," Nucleic Acids Research, vol. 22, pp. 4768-4778, 1994.

[34] A. Krogh, B. Larsson, V. G. Heijne and EL. Sonnhammer, "Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes," Molecular Biology, vol. 305, pp. 567-580, 2001.

[35] H. Mamitsuka, "Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models," Proteins, vol. 33, pp. 460-474, 1998.

[36] P. L. Martelli, P. Fariselli, A. Krogh, R. Casadio, "A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins," Bioinformatics, vol. 18, pp. 46-53, 2002.

[37] P. L. Martelli,P. Fariselli, and R. Casadio, "An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins," Bioinformatics, vol. 19, pp. 205-211, 2003.

[38] G. E. Tusnady, and I. Simon, "Principles governing amino acid composition of integral membrane proteins: application to topology prediction," Molecular Biology, vol. 283, pp. 489-506, 1998.

[39] H. Viklund, and A. Elofsson, "Best alpha-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information," Protein Sci., vol. 13, no. 7, pp. 1908-1917, 2004.

[40] A. Q. Oliveros, C. Roberto C., T. Marta, A. Willy, "On the Application of Hidden Markov Model and Bayesian Belief Network to seismic noise at Las Canadas Caldera, Tenerife, Spain," Chaos, Solitons & Fractals, In Press, Corrected Proof, Available online 13 November 2006.

[41] J. L. Fleiss, "Statistical Methods for Rates and Proportions," Biometrics, vol. 37, pp. 50-62, 1981.

[42] J. Martin Bland and D. G. Altman, "Statistics Notes - The odds ratio," British Medical Journal, vol. 318, pp. 1209-1209, 1999.

[43] P. Fariselli, P. Luigi Morelli, and R. Casadio, "A new decoding algorithm for hidden Markov models improves the prediction of the topology of all-beta membrane proteins," BMC Bioinformatics, vol. 6, no. 12, pp. 70-78, 2005.

[44] W. Velasquez, A. Munoz-Arcentales, and J. Salvachua, "E-Health Services Role in a Smart City," A View after a Natural Hazard Engineering Letters, vol. 27, no. 4, pp. 686-695, 2019.

[45] J. S. Sukono, B. Subartini, J. H. Fransiska Purba, Sudradjat Supian, and Y. Hidayat, " An Application of Genetic Algorithm Approach and Cobb-Douglas Model for Predicting the Gross Regional Domestic Product by Expenditure-Based in Indonesia," Engineering Letters, vol. 27, no. 3, pp. 411-420, 2018.

[46] Z. Tian, G. Wang, and Y. Ren, "AMOAIA: Adaptive Multi-objective Optimization Artificial Immune Algorithm," IAENG International Journal of Applied Mathematics, vol. 49, no. 1, pp. 14-21, 2019.

[47] Q. Zhu, L. Li, and C. Gan, "Modeling and Analysis of the Impact of Adaptive Defense Strategy on Virus Spreading," IAENG International Journal of Applied Mathematics, vol. 48, no.2, pp. 146-151, 2018.

[48] C. Patrone, A. Khodabakhsh, M. Lattuada, and R. Revetria, "Internet of Things Application in the Healthcare Sector," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2018, 23-25 October, 2018, San Francisco, USA, pp. 449-455, 2018.

[49] L. Chang, "A Cell-centered Scheme for Solving Anisotropic Diffusion Equations on Skewed Meshes," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2015, 18-20 March, 2015, Hong Kong, pp. 404-408, 2015.

[50] A. M. Khedr, and R. Bhatnagar, "New Algorithm for Clustering Distributed Data using k-means," Computing and Informatics, Vol. 33, pp. 1001-1022, 2014.

[51] A. M. Khedr, "Decomposable Naive Bayes Classifier for Partitioned Data," Computing and Informatics, vol. 31, pp. 1511-1531, 2012.

[52] A. M. Khedr, "Nearest Neighbor Clustering over Partitioned Data," Computing and Informatics, vol. 30, pp. 1001-1026, 2011.

[53] A. M. Khedr and Salim A., "Decomposable Algorithms for Finding the Nearest Pair," J. Parallel Distrib. Comput., vol. 68, pp. 902-912, 2008.

[54] A. M. Khedr, "Learning k-Classifier from Distributed Databases," Computing and Informatics Journal, vol. 27, pp. 355-376, 2008.

[55] A. M. Khedr and Bhatnagar, R., "Agents for Integrating Distributed Data for Complex Computations," Computing and Informatics Journal, vol. 26, no.2, pp. 149-170, 2007.

[56] A. M. Khedr and Mahmoud, Rania, "Agents for integrating distributed data for function computations," Computing and Informatics, 31. 1101-1125, 2012.

[57] A. M. Khedr, "Decomposable Algorithm for Computing k-Nearest Neighbors across Partitioned Data," International Journal of Parallel, Emergent and Distributed Systems, vol. 31, no. 4, pp. 334-353, 2016.

[58] A. M. Khedr, Ibrahim Attia, "Classification of Vertically Distributed Data using Directed Acyclic Graph," ICGST Conference on Computer Science and Engineering, Istanbul, Turkey, pp. 207-215, 19-21 December 2011.

[59] M. M. Gadallah, Ahmed M. Khedr, and Rania Mahmoud, "Permeability Calculation from Well Logs Using New Learning Techniques," J. Appl. Geophys., vol. 6, no. 2, pp. 113-119, 2007.

[60] A. M. Khedr and Raj Bhatnagar, "A Decomposable Algorithm for Minimum Spanning Tree," Distributed Computing-Lecture Notes in Computer Science Springer-Verlag Hmidmlferg, vol. 29, 18l, pp. 33-44, 2004.

[61] A. M. Khedr, "EDPC: Decomposable Closest Pair Algorithm for Distributed Databases, Engineering Letter, vol. 28, no. 3, pp. 930-938, 2020.

[62] A. M. Khedr, Zaher AL Aghbari, and Ibrahim Kamel, "Privacy preserving Decomposable Mining Association Rules on Distributed Data," International Journal of Engineering & Technology, vol. 7, no. 3.13, pp. 157-164, 2018.

[63] A. M. Khedr, Walid Osamy, Ahmed Salim and Abdel-Aziz Salem, "Privacy Preserving Data Mining Approach for IoT based WSN in Smart City," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, 2019.