# Gene Prediction Based On a Generalized Hidden Markov Model and Some Statistical Models of Related States: a Review

Rui GUO[1, a, *], Jian ZHANG[1], Ke YAN[1], Tian-Qi WANG[1]

[1]Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China

[a]email: 1456295870@qq.com

*Corresponding author

**Keywords:** GHMM, WMM, WAM, WWAM, MC, IMM, MDD.

**Abstract.** In recent years, the methods with a generalized hidden Markov model have gained significant application and development in gene prediction, which is predicting the location and structure of genes in genomic sequences, and produced an army of remarkable programs, such as Genie, GENSCAN, AUGUSTUS, etc. In spite of some limitations, the favorable performance and accuracy these programs still show have withstood the test of practice and time. Here, we provide a comprehensive review of the method of gene prediction with a novel hidden Markov model and some statistical models of related states included, just to share this knowledge with individuals interested in it.

## Introduction

With the development and gradual promotion of the third-generation gene sequencing technology [1], whose sequencing costs become lower while each sequencing length become much longer and sequencing accuracy much higher, we have accumulated more and more complete and clear whole-genome sequences for all kindsof organisms at a faster rate. Driven by this explosion of genome data, gene prediction programs have also proliferated, particularly those that are designed for specific organisms. Although there is still a considerable gap between the true prediction accuracy and the perfect state, the improvement these programs, such as Genie [2], GENSCAN [3], and AUGUSTUS[4], provided on the prediction accuracy is invariably exciting and encouraging, just take Human as an example, at the nucleotide level, more than 90% of genes are accurately predicted, at the exon level, not lower than 80% are predicted, and at the whole-gene level about 45%, which is largely beneficial to the improvement of the efficiency of genome annotation[5].

However, the background knowledge of these implemented programs is extremely complicated, here, and we simply state some. To any gene prediction programs, there are only two important aspects, one is the information made use of by them, which is generally divided into signal sensors and content sensors, and the other is the methods used to combine that information into a reliable predictor, for example artificial neural networks (ANN) [6]. Signal sensors are regarded as the basic and natural approach of finding the presence of functional sites, among the types of functional sites are promoters, start and stop codons, splice sites, branch points, etc., and many early approaches to gene prediction focused on it. As for content sensors, they are measures

 

that try to classify a DNA region into coding and noncoding, which are mainly based on extrinsic similarity with a biologically characterized sequence, e.g., protein sequence, cDNA or expressed sequence tag (EST) sequence and DNA sequence, and intrinsic statistical properties such as codon usage (a triplet of DNA bases), GC content, nucleotide composition, hexamer frequency, and base occurrence periodicity. The methods that use signal sensors or both signal and intrinsic content sensors are known as ab initio methods of gene prediction [7]. The Generalized Hidden Markov Model (GHMM), an approach developed from a HMM,is exactly prestigious one of the methods and also has proven a useful framework for the task of computational gene prediction in eukaryotic genomes, due to its flexibility and probabilistic underpinnings. So we provide a comprehensive review of this method and some classical models of related states involved to share with individuals interested in it.

## Modeling Gene with GHMM

Hidden Markov Models (HMM) have been used for decades in pattern recognition and their applicability to computational biology has gained widely recognition. But as we all know, a standard Hidden Markov model is just a state-based generative model which transitions stochastically from state to state, emitting a single symbol from each state. Although it can produce a certain effect in gene prediction, the recognition accuracy is still far from satisfactory. A GHMM, which is also known as semi-Markov model, generalizes this scenario by allowing individual states to emit a string of symbols rather than only one symbol at a time [8,9]. And it is generally parameterized by its transition probabilities, state duration (i.e., feature length) probabilities, and state emission probabilities. These probabilities influence the behavior of the model in terms of which sequences are most likely to be emitted and which series of states are most likely to be visited by the model as it generates its output.

As the prokaryotic structure is relatively simple, we decide to ignore it in this article. Eukaryotic gene prediction entails the parsing of a DNA sequence into a set of putative CDSs (coding segments) and their corresponding exon-intron structure [10]. Thus, the problem of eukaryotic gene prediction can be almost identical to obtained one of approved parsing sequences over the nucleotide alphabet $\sum = \{A, C, G, T\}$ according to the regular expression [11].

$$\sum^{*}(ATG\sum^{*}(GT\sum^{*}AG)^{*}\sum^{*}\Gamma)^{*}\sum^{*} \quad (1)$$

Here, $\sum^{*}$ denotes a string consisting of any number of the above-mentioned nucleotide alphabet, and it can refer to DNA sequences of intergenic region as well as intron and exon removed some specific endpoint regions, $ATG$ denotes a relatively common start codon, $GT$ and $AG$ denote split sites, respectively donors and acceptors, and $\Gamma = \{TAG, TGA, TAA\}$ denotes a stop codon. It is a fairly typical regular expression of gene structure analysis to state our concern; however, there are still

some crucial points to be supplemented in Eq. 1. Firstly, an additional constraint not explicitly represented in it is that the number of non-intron nucleotides between the start and stop codons of a single gene must be a multiple of three. Secondly, if these nucleotides are aggregated into a discrete number of non-overlapping triples, or codons, then none of these codons must be a stop codon, other than the stop codon which terminates the genes. Finally, note that the $\sum^{*}$ terms in Eq. 1 also permit the occurrence of pseudo-signals, e.g., an ATG triple which does not comprise a true start codon. Gene prediction with a GHMM thus entails parsing with an ambiguous stochastic regular grammar; the challenge is to find the most probable parse of an input sequence, given the GHMM parameters and the input sequence.

In the case of standard hidden Markov Models, this optimal parsing (or decoding) problem can be solved easily with the well-known Viterbi algorithm [12], a dynamic programming algorithm with run time linear in the sequence length, if given a fixed number of states. Since each state can now emit more than one symbol at a time in the case of GHMMs, the Viterbi algorithm is required to modify to solve the following optimization problem.

$$\begin{aligned}
\Phi_{optimal} &= \arg\max\ p(\Phi \mid S) \\
&= \arg\max \frac{p(\Phi, S)}{p(S)} \\
&= \arg\max\ p(\Phi, S) \\
&= \arg\max\ p(S \mid \Phi)\, p(\Phi) \\
&= \arg\max \prod_{i=1}^{n} p_e(S_i \mid q_i, d_i)\, p_t(q_i \mid q_{i-1})\, p_d(d_i \mid q_i)
\end{aligned} \tag{2}$$

Where $\Phi$ is a decoding of the sequence consisting of a series of states $q_i$ and state durations $d_i$, $0 \le i \le n$, with each state $q_i$ emitting subsequence $S_i$ of length $d_i$, so that the concatenation of all $S_0 S_1 ... S_n$ produces the complete output sequence $S$, but note that states $q_0$ and $q_n$ are silent, producing no output, $p_e(S_i \mid q_i, d_i)$ represents the probability that state $q_i$ emits subsequence $S_i$, given duration $d_i$; $p_t(q_i \mid q_{i-1})$ is the probability that the GHMM transitions from state $q_{i-1}$ to state $q_i$; and $p_d(d_i \mid q_i)$ is the probability that state $q_i$ has duration $d_i$. The $\arg\max$ is over all parses of the DNA sequence into well-formed exon-intron structures, hence, the problem is to find the optimal parse which maximizes the product in Eq. 2.

With no consideration of frame constraints and the single or double strand question and through structural transformations of Eq. 1, we can get an oversimplified GHMM

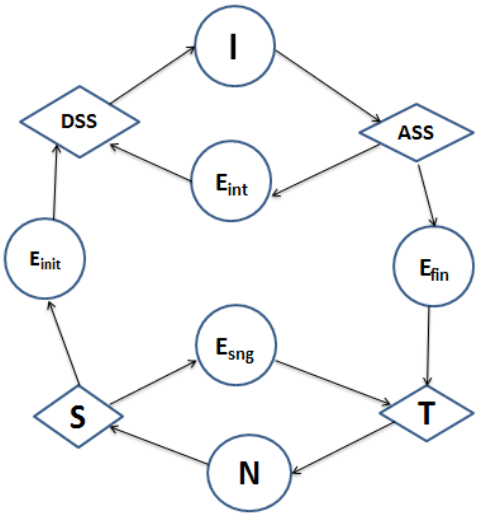


Figure 1.A typical GHMM topology

topologyof gene structure depicted in Fig. 1,which is just right beneficial for us to assign to the elements of Eq. 2 and then handle it [13]. In Fig. 1, S means start codon, usually referring specifically to *ATG*; T means stop codon, mainly referring to *TAG*, *TAA* and *TGA*; *DSS* and *ASS* respectively are donor and acceptor, separately corresponding to string *GT* and *AG* described in Eq. 1. These characters in diamonds denote the states for fixed length features, which represent signal sensors and can be simulated by the WMM [14], WAM [15], WWAM, MDD [2] tress, etc. N denotes intergenic, I denotes intron, Einit, Eint, Efin and Esng respectively denote initial exon, internal exon, final exon and single exon. They are included in circles that signify states for variable length features, associated with these circular states are variable-length models and can be simulated by the MC or IMM model [16]. Combining these related state-simulated models with the approaches based on Dynamic Programming such as the Prefix Sum Arrays (PSA) algorithm and the Dynamic Score Propagation (DSP) algorithm [11], we can confidently gain what we want from the Eq. 2. However, as for the details of those state-simulated models, we will elaborate them below.

## Some Statistical Models of Related States

### WMM

WMM is the abbreviation of the weight matrix method introduced by Staden in 1984, which is one of the earliest and most influential approaches to construct some numerous models of biological signal sequences such as donor and accepter sites, promoters, polyA_signal, etc.It only describes the distribution relationship of different characters or states in the same position, with no need to consider the associated effect produced by its adjacent and non-adjacent positions. It requires that the training dataset of signal sequences must be aligned, namely, all the length of the signal sequence must be the same. And in the WMM model, the frequency $p_j^{(i)}$ of each

nucleotide $j$ at each position $i$ of a signal of length n is derived from a collection of aligned signal sequences and the product $P\{X\} = \prod_{i=1}^{n} p_{x_i}^{(i)}$ is used to estimate the probability of generating a particular sequence, $X = x_1, x_2, ..., x_n$ . The following shows an example ofprobability model results of signal ASS trained by the WMM.

```
part of values of probability model of signal ASS trainend by WMM:
A    0.092437   0.0735294  0.0619748  0.0829832  0.27626    0.0351891  1   0
C    0.393382   0.415966   0.450105   0.382353   0.284139   0.741597   0   0
G    0.113445   0.072479   0.0535714  0.049895   0.214286   0.00367647 0   1
T    0.400735   0.438025   0.434349   0.484769   0.225315   0.219538   0   0
```

Figure 2.

In Fig. 2, the sum of probabilities of nucleotide A, C, G and T in the same column is constantly equal to 1. If there are probability distributions of individuals behaving obvious bias instead of uniform distribution, it can be thought that the WMM model has the ability to accurately describe the features in this signal sensor. The more such columns are, the more accurate these features of this signal sensor are depicted by the WMM. However, observing more carefully, we can perceive a wrong place that the probability values of some individuals in the last two are 0, which may result in a certain analytical error caused by aforementioned decoding algorithms. Therefore, in order to get rid of this situation, although it is indeed entirely correct description to signal features, we must make a slice of adjustments to ensure that all the probability values of individuals in the same column are not equal to 0.

**WAM, WWAM**

WAM, termed weight array model, is an enhanced version of the WMM. It was firstly applied by Zhang & Marr in 1993, in which dependencies between adjacent positions are considered. In this model, the probability of generating a particular sequence is:

$$P\{X\} = p_{x_1}^{(1)} \prod_{i=2}^{n} p_{(x_{i-1}, x_i)}^{(i-1,i)} \tag{3}$$

Where $p_{j,k}^{(i-1,i)}$ is the conditional probability of generating nucleotide $X_k$ at position $i$ , given nucleotide $X_j$ at position $i-1$, it is estimated from the corresponding conditional frequency in the set of aligned signal sequences.Of course, high-order WAM models capturing second-order or third-order dependencies in signal sequences could be used in principle, but it is emphasized that there must be sufficient data available to estimate the increased number of parameters in such models.

WWAM, called the windowed weight array model, is an improved version of the WAM model. It, for example the WWAM of order $k$ and window size $2r+1$, can be regarded as an inhomogeneous Markov Model of order $k$ in which the probability of

observing nucleotide at position $i$, given that the preceding $k$ nucleotides are $x_1,...,x_k$ is estimated by the relative frequency of observing $x$ after nucleotides $x_1,\ldots,x_k$ in the training data at one of the positions in the window $i-r,...,i+r$.Of course, for that purpose, the training data is also required to be aligned with respect to the biological signal modeled.

**MDD**

MDD means Maximal Dependence Decomposition, which is firstly introduced to model donor splice sites by Chris Burge & Samuel Karlin in 1997. It is designed to compensate for the lack of modeling capability of the above-mentioned model to the donor splice signal, because of the highly significant dependencies between non-adjacent as well as adjacent positions in it. However, itcan perfectly capture these most significant dependencies, essentially by replacing unconditional WMM probabilities by appropriate conditional probabilities provided that sufficient data is available to do so reliably. Given a data set $D$ consisting of $N$ aligned sequences of length $k$, our first step is to gain a consensus nucleotide or nucleotides at each position. Then, for each pair of position, calculate the $\chi^2$ statisticsof $C_i$ versus $X_j$ ($C_i$ defined that if the nucleotide at position $i$ matches the consensus at $i$, its value is 1, otherwise 0, and nucleotide indicator $X_j$ identifying the nucleotide at position $j$) for each $i$, $j$ pair with $i \neq j$. If no significant dependencies are detected, given an appropriate P-value, then the simple WMM should be sufficient. If significant dependencies exclusively or predominantly locate at adjacent positions, then a WAM model should be appropriate. If, however, there are strong dependencies both of non-adjacent and adjacent positions, then we process it as follows.

(1) Calculate, for each position $i$, the sum $S_i = \sum_{j \neq i} \chi^2(C_i, X_j)$, which is a measure of the amount of dependence between the variable $C_i$ and thenucleotides at the remaining positions of the site.

(2) Choose the value $i_m$ such that $S_{i_m}$ is maximal and partition $D$ into two subsets: $D_{i_m}$ all sequenceswhich have the consensus nucleotide(s) at position $i_m$; and $D_{i_m}$ all sequences which do not.

Then repeat steps (1) and (2) on each of the subsets, $D_{i_m}$ and $D_{i_m}$, until yield a binary subdivision tree with at most $k-1$levels. Of course, this process can also be ended when either of the following two conditions occurs. One is that no significant dependencies between positions in a subset are detected so that further subdivision is

not indicated. The other is that the number of sequences remaining in a subset becomes so small that reliable WMM frequencies could not be determined after further subdivision. Finally, we can form a composite model by proportionallycombiningthose separate WMM models, which contains the crucial information of dependencies between positions. It will be extremelybeneficial to the development of accuracy of gene prediction.

**MC**

MC, termed Markov Chain, is a well-known tool for analyzing biological sequence data, as an example; modeling the information of base conditional distribution of intergenic, intron and exon. A first order Markov Chain is a sequence of random variables where the probability that $x_i$ takes a particular value only depends on the preceding variable $x_{i-1}$, and according to the natural generalization of this definition, in an $n^{th}$ order Markov Chain, the probability distribution of the random variable $x_i$ only depends on the $n$ preceding bases. When modeled a state $q_i$ with the duration length $d_i$ in a gene structure, it can be evaluated by this form:

$$\prod_{j=0}^{n-1} p(x_j \mid x_0 \ldots x_{j-1}) \prod_{j=n}^{d_i-1} p(x_j \mid x_{j-n} \ldots x_{j-1}) \tag{4}$$

Here, $x_j$ is the $j^{th}$ nucleotide in the sequence of the putative feature, $d_i$ is the length of that feature, and $p(x_j \mid x_{j-n} \ldots x_{j-1})$ is the conditional probability of nucleotide $x_j$, given its $n$ predecessor nucleotides. However, in spite of its simplicity and flexibility, there is still one essential point need to be noted that sufficient data must be ready for accurately estimating the probability of each base occurring after every possible combination of $n$ preceding bases, which requires $4^{n+1}$ probabilities to be estimated simultaneously from that training data, but it is generally tremendously difficult, e.g., 4096 probabilities for a commonly used $5^{th}$ -order model.

**IMM**

IMM, namely, interpolated Markov models, is unprecedentedly used by the team of Steven's (1997) in a system of gene prediction, called GLIMMER, which is exclusively to identify coding regions in microbial DNA. It is based on this idea that in the genome of some organisms, some low-mers will occur too infrequently to give reliable estimates of the probability of the next base, while some high-mers occurring

frequently can do so, and exploit a linear combination of probabilities obtained from several lengths of oligomers to make predictions by giving high weights to oligomer that occur frequently and low weights to those that do not. Therefore, the IMM is a model which uses a longer context to make prediction as far as possibly, taking advantage of the greater accuracy produced by high-order Markov models, and can overcome the problem of the incompleteness of base probability model trained without adequate training data set. But if the statistics on longer oligomers are insufficient to produce good estimates, it can fall back on shorter oligomers to make its predictions.

Yet as for the reasons why the high order Markov models can perform better than the low order ones, theoretically speaking, the higher the order of Markov models used is, the stronger constraints the conditional probability of each nucleotide at a certain position will gain, and then it also can be evaluated much more accurately. In addition, as for the details of the IMM model which are applied and constructed, we firstly need to know the probability, $p(S \mid M)$ that the model generated a new sequence $S$, given a trained IMM model. It can be computed as [16]

$$p(S \mid M) = \sum_{x=1}^{n} IMM_k(S_x) \tag{5}$$

Where, $S_x$ is the oligomer ending at position $x$, $n$ is the length of the sequence and $k$ is the order of the interpolated Markov model. $IMM_k(S_x)$, the $k^{th}$-orderIMM score, can be written as:

$$IMM_k(S_x) = \lambda_k(S_{x-1}) \bullet p_k(S_x) + [1 - \lambda_k(S_{x-1})] \bullet IMM_{k-1}(S_x) \tag{6}$$

Here, $\lambda_k(S_{x-1})$ is the numeric weight associated with the $k$-mer ending at position $x-1$ in the sequence $S$ and $p_k(S_x)$ is the estimate obtained from the training data of the probability of the base located at $x$ in the $k$ order model.

In this section we roughly describe how to compute the values of the $\lambda$ parameters for the $k^{th}$-order IMM in Eq. 6. In GLIMMER, the value of $\lambda_k(S_{x-1})$ that we associate with $p_k(S_x)$ is regarded as a measure of our confidence in the accuracy of thisvalue as an estimate of the true probability, which can be determined by two criteria. The first of these is simplefrequency of occurrence. If the number of occurrences of context string $S_x$ in the training data exceeds a specific threshold value, it can be straightly set to 1.0, which the AUGUSTUS only adopts and the

threshold is similarly 400 (which gives ~95% confidencethat the sample probabilities are within $\pm 0.05$ of the true probabilities from which the sample was taken. ).And when there are insufficiently many sample occurrences of a context string to estimate the probability of the next base with confidence, the other criteria is that, for a given context string $S_{x,k}$ of length $k$ ,comparing the observed frequencies of the base, $f(S_{x,k},a)$ , $f(S_{x,k},c)$ , $f(S_{x,k},g)$ and $f(S_{x,k},t)$ ,calculating IMM probabilities using the next shorter context, $IMM_{k-1}(S_{x,k-1},a), IMM_{k-1}(S_{x,k-1},c), IMM_{k-1}(S_{x,k-1},g)$ and $IMM_{k-1}(S_{x,k-1},t)$, and using a $\chi^2$ test to determine how likely it is that the four observed frequencies are consistent with the IMM values from the next shorter context. If they are significantly different, chose them as better predictors of the next base by giving them a high $\lambda$ value, while they offer little predictive value and hence are given a lower $\lambda$ value. Specifically, when calculate the $\lambda^2$ confidence $c$ that they are not consistent and set.

$$\lambda_k(S_{x-1}) = \{ \begin{matrix} 0.0 & if & c < 0.5 \\ \frac{c}{400} \sum_{b \in \{acgt\}} f(s_1 s_2 \ldots s_k b) & if & c \geq 0.5 \end{matrix} \tag{7}$$

Of course, all roads lead to Rome, as for the other methods of assigning $\lambda$ values for IMMs, or even for building nonuniform Markov models are cited in [17].

## Conclusion

Gene prediction, as an important open problem in bioinformatics, have been cultivated by human for many years, and with all kinds of advanced computational approaches, such as ANN, fuzzy logic, Decision Tree, etc. significant progress have taken place in this area, however, some problems along with this progress are still vividly exist, e.g., how to handle non-canonical splice sites and predict alternative splice sites, how to process a large number of false positives resulted from splice site programs and how to locate exactly short exons, especially those bordered by long introns, which concurrently make the gene prediction more permanently challenging, thus if we still expect to perfectly and thoroughly solve it, more effort and time, especially innovation, must be devoted. Here, in order to more or less help to reach the goal as soon as possible, we write this review thatmainlyinvolves in implementation points of the method GHMM and some statistical models of related statesincluded, and share it, also hoping that brilliant you can make a greater breakthrough.

**Acknowledgment**

This article is partly supported by the Shenzhen Municipal Science and Technology Innovation Council (Nos. JCYJ20140904154645958).

**References**

[1] Cairui, L., Z. Changsong, and S. Guoli. "Recent progress in gene mapping through high-throughput sequencing technology and forward genetic approaches." *Yi chuan= Hereditas/Zhongguoyichuanxuehuibianji* 37.8 (2015): 765-776.

[2] Haussler, David Kulp David, and Martin G. Reese Frank H. Eeckman. "A generalized hidden Markov model for the recognition of human genes in DNA." *Proc. Int. Conf. on Intelligent Systems for Molecular Biology, St. Louis*. 1996.

[3] Burge, Chris, and Samuel Karlin. "Prediction of complete gene structures in human genomic DNA." *Journal of molecular biology* 268.1 (1997): 78-94.

[4]Stanke, Mario, and Stephan Waack. "Gene prediction with a hidden Markov model and a new intron submodel." *Bioinformatics* 19.suppl 2 (2003): ii215-ii225.

[5] Yip, Kevin Y., Chao Cheng, and Mark Gerstein. "Machine learning and genome annotation: a match meant to be." *Genome Biol* 14.5 (2013): 205.

[6] Zhou, You, et al. "An artificial neural network method for combining gene prediction based on equitable weights." *Neurocomputing* 71.4 (2008): 538-543.

[7] Goel, Neelam, Shailendra Singh, and Trilok Chand Aseri. "A comparative analysis of soft computing techniques for gene prediction." *Analytical biochemistry* 438.1 (2013): 14-21.

[8] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77.2 (1989): 257-286.

[9] Rabiner, Lawrence R., and Biing-Hwang Juang. "An introduction to hidden Markov models." *ASSP Magazine, IEEE* 3.1 (1986): 4-16.

[10] Salzberg, Steven L., D. B. Searls, and S. Kasif. "Computational methods in molecular biology." *Computational Methods in Molecular Biology*49.2(1999):191-192.

[11] Majoros, William H, et al. "Efficient decoding algorithms for generalized hidden Markov model gene finders." *Bmc Bioinformatics* 6.2(2005):8-16.

[12] Ryan, Matthew S., and G. R. Nudd. "The Viterbi Algorithm." *Warwick Research Report Rr* 37.2(1993):160 - 163.

[13] Michael, Lynch. "The origins of eukaryotic gene structure.." *Molecular Biology & Evolution* 23.2(2006):450-468.

[14] Staden, R.,. "Computer methods to locate signals in nucleic acid sequences.." *Nucleic Acids Research* 12.1(1984):505-519.

[15] Zhang, M. Q., and T. G. Marr. "A weight array method for splicing signal

analysis.." *Computer Applications in the Biosciences Cabios*9.5(1993):499-509.

[16] Salzberg, S L,, et al. "Microbial gene identification using interpolated Markov models.." *Nucleic Acids Research* 26.2(1998):544-8.

[17] Ristad, E., and R. Thomas. "Nonuniform Markov Models." *Acoustics, Speech, and Signal Processing, IEEE International Conference on* IEEE Computer Society, 1997:791-791.