

Identification of Bacterial and Eukaryotic Genes using Interpolated Markov Model

Jacob Pritt
mpritt4@jhu.edu

Guannan Ren
gren3@jhu.edu

Abstract

1 Introduction

In the field of computational biology, gene identification and location is important in various applications such as controlling the expression level of genes. In the scope of the project, we seek to investigate a highly accurate gene identification tool which distinguishes coding and non-coding sections of the genome. We will also compare the IMM with fixed length Markov chains for the classification of coding regions. GLIMMER (Gene Locator and Interpolated Markov ModelER), is a tool designed to find the genes inside a microbial genome. It uses an interpolated Markov model which is an improvement upon the Markov chain model. In the original GLIMMER publication, the authors noted that from a test batch of hundreds of bacterial genomes, the software was able to achieve over 95% accuracy on gene identification. For our project, we seek to implement the IMM algorithm used in GLIMMER and to match our classification results with that of the publication's. Next we will run our algorithm on eukaryotic genomes to examine the gene identification performances on eukaryotic species. We expect the accuracy of gene identification to drop significantly for eukaryotic genes, since eukaryotic genomes have highly complex coding sequences than that of the prokaryotes.

GLIMMER uses interpolated Markov models. IMM is more powerful than simple Markov

chains. A fixed-order Markov chain uses a fixed number of preceding bases to predict the DNA sequence. For example, a 5th order Markov chain would use five previous bases to predict the next base in the sequence. Markov chain run into problems when the training data is low such that the feature vectors created is too sparse to estimate the probability of bases after every combination of 5-mers. A k th-order Markov model would require up to $4^{(k+1)}$ probabilities to be estimated from the training data. In order to estimate the probabilities, at least few occurrences of all possible k -mers must be present to yield a valid estimate.

It is often difficult for a training genome to have sufficient 5mers to provide a valid estimate. The interpolated Markov Model works around this problem by using a combination of probabilities of different length oligomers to give a better prediction. The IMM assigns different weights to each oligomer depending on its number of occurrences within the training DNA sequence. A highly frequent k -mer occurrence receives a higher weight while a low occurrence k -mer receives a lower one. When the sample data lacks a certain occurrences for a specific length oligomer, the IMM will resort to using a combination of the shorter oligomers sharing similar prefix sequence for its decision.

1.1 Reading Frame and Coding Sequence Description

The original GLIMMER software first parts the DNA sequence into six different reading frames. Consecutive reading frames share the DNA se-

quence but differ on which nucleotide is first read. Since each nucleotide triplet specifies a codon during translation, there can be 3 different positions at which we can start reading codons at. And, three more reading frames applies to the reverse complement strand of the DNA. This gives us a total of 6 possible reading frame positions to find starting codons for coding sequences. From each reading frames, multiple open reading frames can be found. Each open reading frame is the DNA sequence that starts with the start codon and ends with the stop codon, without having any stop codon in the middle. However, there can be additional start codons in the reading frame. An ORF would generate a protein sequence once translated.

1.2 Why Identification of Coding Region Useful?

Talk about the implication of matching the coding region with different gene expression

2 Dataset Overview

In our project, we seek to duplicate the results obtained from the original publication of GLIMMER. From the specifications of the original paper, we are using *H. Influenzae* to test the IMM. The first 5000 DNA nucleotides within the complete *H. Influenzae* genome are used to train our model, and the rest are used for testing. The complete genome for the organism was retrieved from GenBank database.

3 Methods

3.1 Classification

To better convey how the interpolated Markov model algorithm works, we will first explain how a Markov chain probabilistic model is able to classify its target. In a sense, a DNA sequence can be viewed as a sequence generated by a probabilistic model of the interactions between its random variables, which are the 4 coding nucleotides in our case. Each of the random variable can take the values of A, C, G, or T. The Markov chain probabilistic model captures this probability by enumerating the probabilities as states. The probability of a DNA nucleotide to take on a certain base depends on its previous neighbor in the DNA sequence. This can be illustrated by observing the model for a simple

1-mer model as follows,

$$P(S_{x+1} | S_1, \dots, S_x) = P(S_{x+1} | S_x)$$

TODO get rid of

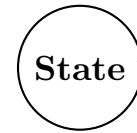


Figure 1: A directed graph

For a first order Markov chain, the occurrence probability of the nucleotide random variable X_i only depends on the probability of the preceding variable X_{i-1} . This model can be extended to higher orders, in which the random variables will depend on additional preceding variables. For example, a k -th order Markov model specifies that X_i will depend on variables X_{i-k} to X_{i-1} . It is worth noting that by increasing the order of the Markov model, the predictive power of the model will steadily increase. However, by increasing the number of preceding variable probabilities, the space requirement for storing all of the combinations increases. For example, a first order Markov model will hold 4^2 probabilities for nucleotide combinations, while a fifth order Markov model can potentially hold 4^6 probabilities.

For a first order Markov chain, the current state of a variable is only dependent upon the previous state. We generalize this model to higher order by using addition states prior to predict the current state. The combination of Markov chains of different orders allows us to come up with the algorithm for an interpolated Markov model where the weights from different orders of Markov chain are tallied and used to produce a final score for the sequence. By choosing the model with the highest probability, we would figure out where the true coding region starts.

As mentioned, a higher-order Markov chain would do at least as well as a lower-order chain given sufficient training data. However, when we move to a higher-order Markov chain, the probabilities for sequences are sparse and the sparsity leads

to inaccuracies. For rare k-mers not satisfying a certain occurrences threshold, the interpolated Markov model resorts to combining all probabilities from previous base sequences. There would be more data available for lower-order models since shorter sequences appear more frequently.

$$P(S | M) = \sum_{x=1}^n \text{IMM}_8(S_x) \quad (1)$$

3.2 Features

3.3 Feature extraction/selection

We can compute the probability of base s_x given the i previous bases as

3.4 Weight calculation

In this section we describe how GLIMER computes the weights for the different kth-order IMM. First, a set of known coding sequences are assembled to a training set. To be certain these are truly coding is problematic at times. Hence, we decided to only use ORFs satisfying a certain nucleotide length requirements.

From the training set of genes, the frequencies of occurrences of all possible substring patterns of length 1 to $k + 1$ are counted and stored for each of the reading frames. If we consider a single reading frame at a time, and let $f(S)$ denote the number of occurrences of the sequence $S = s_1 s_2 \dots s_n$, then, we can get an initial estimates of the probability of base s_x given the previous sequences $s_{x-i}, s_{x-i+1}, \dots, s_{x-1}$, which is denoted as $S_{x,i}$. The probability of base s_x is given as the following:

$$P_i(s_x) = P(s_x | S_{x,i}) = \frac{f(S_{x,i} s_x)}{\sum_{b \in \{acgt\}} f(S_{x,i} b)} \quad (2)$$

The weight attached to $P_i(s_x)$ can be seen as a confidence measure of how close the estimated probability matches that of the true probability. GLIMMER computes the weight factor by first checking the frequency of occurrences for the k-mer sequence. When the frequency exceeds certain threshold value, the weight for the k-mer sequence is set to 1. However, the k-mer sequence frequency count is less than the threshold number, we resort to an alternate method for calculating the weight factor. (link to figure?). For

a given context string $S_{x,i}$ of length i , we can compare the observed frequencies of all possible base pairings, $f(S_{x,i}, a)$, $f(S_{x,i}, c)$, $f(S_{x,i}, g)$, and $f(S_{x,i}, t)$ with the previously calculated IMM probabilities using the next shorter context, which is $\text{IMM}_{i-1}(S_{x,i-1}, a)$, $\text{IMM}_{i-1}(S_{x,i-1}, c)$, $\text{IMM}_{i-1}(S_{x,i-1}, g)$, $\text{IMM}_{i-1}(S_{x,i-1}, t)$. Now, using a χ^2 test, we can determine how likely it is that the four observed frequencies are consistent with the IMM values from the next shorter context. If the frequencies differ significantly from the IMM values, we would give them a higher weight since they would give us more information. The result of the χ^2 confidence c can be used to designate a weight as follows:

$$\lambda_i(S_{x-1}) = \begin{cases} 0, & \text{if } c < 0.50 \\ \frac{c}{400} \sum_{b \in \{acgt\}} f(s_1, s_2, \dots, s_i b), & \text{if } c \geq 0.50 \end{cases} \quad (3)$$

For the purposes of reproducing the results of the research publication, we set our frequency of occurrences threshold to be 400.

$$\text{IMM}_k(S_k) = \lambda_k(S_{k-1}) \cdot P_k(S_k) + [1 - \lambda_k(S_{k-1})] \cdot \text{IMM}_{k-1}(S_k) \quad (4)$$

4 Results

As previously mentioned, we are using the complete H. influenzae genome to evaluate the performance of our IMM gene classifier. We ran the IMM algorithm under different training and test settings. We were able to evaluate the model performance at maximum k-mer length between the range 0 to 9. A maximum k-mer length of 0 indicates that we are predicting independently of the preceding nucleotide in the DNA sequence, while a maximum k-mer length of 9 indicates that the current nucleotides probability can be dependent upon at most 9 preceding DNA nucleotides.

We implemented and tested fixed-length Markov chain models of orders between 0 and 9 to compare our IMM results against.

4.1 Comparison of IMM and Markov chain on H.influenzae data

Figure 2 shows how the IMM compares against the Markov chain model at identification of H. influenzae genes at the different model complexities.

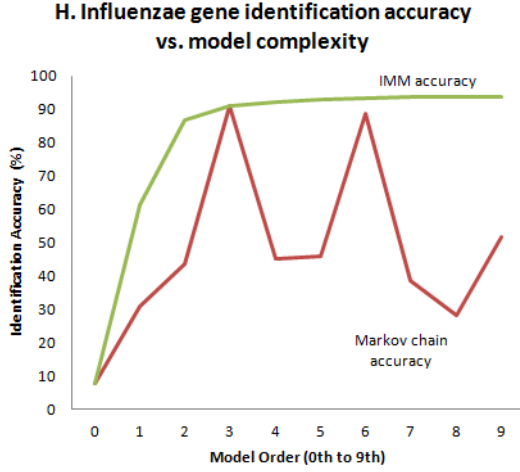


Figure 2: Accuracy comparison for IMM and MC models at different model complexity for *H. influenzae* genome

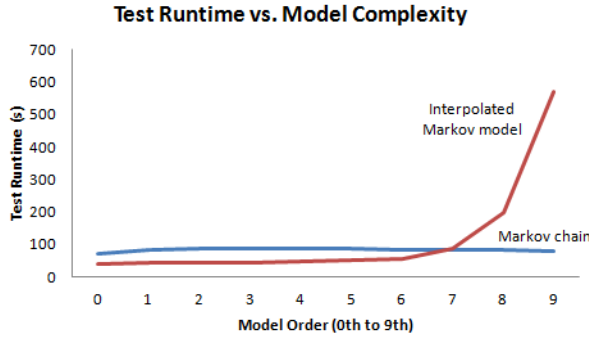


Figure 3: Runtime comparison for IMM and MC models at different model complexity

When running our IMM and Markov chain programs, we imposed the same set of selection criteria as the mentioned in the publication. First, we would rule out all ORFs that are less than 500 bases long, since longer ORF length are more likely to be coding regions. Next, we impose the constraint that the potential ORFs selected must not overlap one another.

Besides comparing the gene identification accuracies between our models, we chose to record the running time for our models as well. We ran both the IMM and the Markov chain models at different complexities on the *H. influenzae* genome to generate Figure 3.

We thought checking the number of false positives would be interesting. Figure 4 shows the

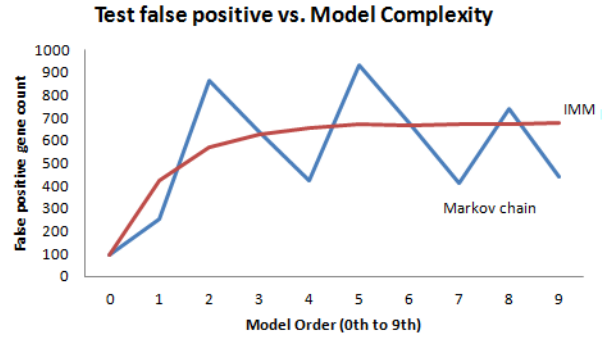


Figure 4: False positive gene identified for IMM and MC models at different model complexity

Organism	MC Acc.(%)	IMM Acc.(%)
<i>H. influenzae</i>	51.87	93.87
<i>H. pylori</i>	44.47	96.88
<i>E. coli</i>	27.92	58.79

Table 1: Generalization of Markov chain and interpolated Markov model to other prokaryotic genomes at 9th order complexity

counts of false positives selected by our models.

We were interested to see how our model generalize to other bacterial genomes, so we ran similar tests on *H. pylori* and *E. coli* genomes. The comparisons between an 8th order interpolated Markov model and an 8th order Markov chain for different prokaryotes are shown in Table 1.

4.2 Gene finding on Eukaryotic organisms

While not mentioned in the original GLIMMER publication, gene identification on Eukaryotic genome sequences using the IMM is something we set out to test. Using the complete genome for *Drosophila melanogaster* chromosome X, we ran our IMM and fixed-length Markov chain algorithms to yield the results as shown in Figure 5.

We noticed the poor gene identification accuracy. This result is expected, since Eukaryotic genomes are much more complex than that of Prokaryotes. For example, the extra non-coding intron sequences and the regulatory sequences such as promoters and enhancers increase the noise which our gene identifier has to go through.

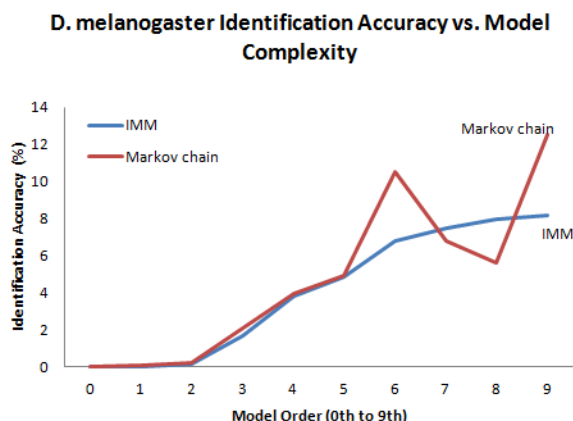


Figure 5: Accuracy comparison for IMM and MC models at different model complexity for *D. melanogaster* genome

4.3 Gene finding on with interpolated Hidden Markov Models

TODO

5 Discussion

When comparing our 8th order IMM results against the same model as used in the publication, we notice that our accuracy value, of 93.78 %, differs from the accuracy of 97.8 % reported. There are several reasons that might have caused this difference.

We noticed there were 1717 annotated genes for the publication while our annotated gene file only contained 1174 genes. We suspect that they had additional experimental data to work with. This difference in the number of annotated genes only affected the number of false positive ORF turn outs in our results. Since we are mainly focused on replicating the results for true positives, we decided to ignore this difference.

From the fixed-length Markov chain results, we noticed that the identification accuracy peaks at intervals of three bases. We think that this might be due to the codon length.

Overall, we are satisfied with the sensitivity of our IMM on the identification of true bacterial genes.

5.1 Comparison to Project Proposal

Judging by our original project proposal, we have accomplished the must-achieve milestones by im-

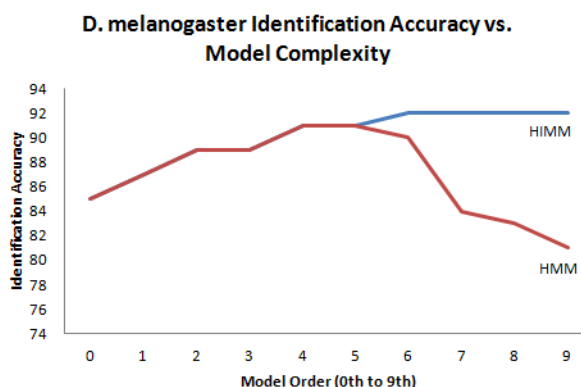


Figure 6: Accuracy comparison for HIMM and HMM models at different model complexity for *D. melanogaster* genome

plementing an IMM classification program closely following the program specifications of the GLIMMER publication. We have also managed to finish the features mentioned in the expected to achieve section. We have written a Markov model to compare the results obtained to that of the IMMs. If we had additional time, we would likely to have accomplished the reach-goals for improving upon the IMMs accuracy by training additional model organisms and to implement an interpolated context model to decide which positions to use for prediction.

References

- Salzberg, S.L., Delcher, A.L., Kasif, S. and White, O. 1998. *Microbial gene identification using interpolated Markov models* Nucleic Acids Research: Oxford University Press.