

Improving Start Codon Prediction Accuracy in Prokaryotic Organisms Using Naïve Bayesian Classification

Sean Landman and Imad Rahal

Computer Science Department

College of St. Benedict / St. John's University

Collegeville, MN 56321

srlandman@csbsju.edu and irahal@csbsju.edu

Abstract

With an overwhelming amount of genetic data now becoming publicly available, there is a growing need to develop more effective gene prediction methods that produce reliable results. Although prediction of the stop codon location for genes in prokaryotic organisms is largely considered to be a solved problem, accurate prediction of the exact start codon location continues to lag behind because of the ambiguity for these start codons in the genetic code. This paper will detail a new approach to predicting more precise gene locations for both the start and stop codon in prokaryotic organisms. This approach uses gene prediction results from other prediction programs to find consistently predicted gene locations. It then uses these “consistent genes” as a training set for Naïve Bayesian classification to improve accuracy in the “ambiguous genes,” those in which there is some variability or inconsistency in predicted locations between the prediction programs.

1 INTRODUCTION

Once an organism's genome is obtained through a process known as sequencing, scientists and researchers must then apply gene prediction techniques to discover the exact locations of an organism's genes in this DNA data. The result of the sequencing process for an organism's genome is essentially one long sequence of DNA data. This DNA data contains all of the genes for the organism, but it is not known initially where these genes are located in this unorganized mess of DNA. Some of the DNA is coding DNA (genes), while much of it is non-coding DNA. The goal of gene prediction is to differentiate the coding DNA portions from the non-coding DNA portions. Thus, gene prediction is essentially a data mining classification problem. Gene prediction results in a list of all the locations in the DNA that are believed to be coding DNA, or in other words, the predicted locations of the genes.

Knowing the locations of genes leads to countless other discoveries such as uncovering information about the proteins they code for, the way they are activated and expressed, and the way they are transmitted through the process of evolution. In many bacteria and other prokaryotic organisms, the study of individual genes often leads to other discoveries, such as in medicine or microbiology. In order to make such discoveries, new genes must be discovered and analyzed. Not only does gene prediction reveal the locations of known genes in DNA, but it also uncovers genes that were previously not known to exist.

Just as gene prediction will help to locate previously unknown genes, it will also falsely predict genes that are actually non-coding sequences. Thus, it is also important for gene prediction to produce accurate results that predict locations for as many of the actual genes as possible while minimizing wrong predictions. Since there is no one reliable way to identify a sequence of DNA as a gene, thorough research in gene prediction techniques is a necessity for obtaining accurate and reliable gene location predictions.

In this paper, we take a critical look at the current status of gene prediction for prokaryotic organisms, and some of the existing computational methods that are commonly used for this task. We will highlight the areas in which prokaryotic gene prediction needs further improvement, and propose a new technique that addresses these issues by reconciling the differences between pre-existing techniques, improving accuracy and reliability.

2 ANALYSIS OF GENE PREDICTION METHODS

Research in gene prediction has progressed significantly over the past decade. Although there is still much work to be done in eukaryotic gene prediction research, some have begun to declare prokaryotic gene prediction as a solved problem, citing the 99% accuracy that can be achieved by a variety of modern programs (see Table 1). However, there are

still some issues that require further research and improvement before prokaryotic gene prediction can be considered “solved” by any stretch of the imagination.

Prediction Program	Stop Codon Accuracy	Start Codon Accuracy
GLIMMER 3.02	99.5%	90.6%
MED 2.0	99.0%	90.9%
ZCURVE 1.02	98.8%	88.3%

Table 1: Percentage of stop codons and start codons from the EcoGene dataset found by various prediction programs run on *E. coli*.

For biological reasons, the beginning location (or start codon) of a gene is much more difficult to predict than the end location (or stop codon). This is because of ambiguity in the genetic code for start codons. The three possible stop codons (TAA, TGA, or TAG) always signify the end of the gene. However, the three nucleotide sequences that represent start codons (ATG, TTG, or GTG in prokaryotic organisms), can either signify the start of a gene or can encode for an amino acid, depending on the context. For a given gene and its corresponding stop codon, there are many 3-nucleotide sequences upstream that could potentially represent the start codon. Some of these are part of the non-coding DNA upstream of the gene, one of these is the true start codon, and the rest are part of the gene itself and encode for their respective amino acids (Methionine for ATG, Leucine for TTG, and Valine for GTG). Once a gene is located and its stop codon location determined, there are many start codon locations that might represent the beginning of this gene. This difficulty in choosing which of these potential start codons is the true location is what makes start codon prediction generally less accurate than stop codon prediction.

Conventionally, accuracy of prediction programs refers to its ability to predict the correct stop codon only. While most modern prokaryotic gene prediction programs are able to predict the stop codons of genes at 99% accuracy, their accuracy in predicting the correct start codons is actually roughly 90% (see Table 1), which is still unsatisfactory if these results are meant to form the foundation for future research and studies on these genes.

Another remaining issue is how to fairly measure the accuracy of a gene prediction program. Any given prokaryotic organism can have thousands of genes, and the exact location is positively known for only a few of these. The vast majority of the documented gene locations in public databases come from gene prediction programs, and thus are in actuality nothing more than predictions. This is to be expected however, since verification of gene locations using biological lab techniques is a costly and timely process, and would simply be impossible to do for every gene from every organism. This is why accurate gene prediction is so important, so that these gene locations that are being predicted can be relied

on without lab verification. However, since gene prediction is not yet at that reliable point, many of the documented gene locations in databases are still theoretical, especially for the start codon location, and thus measuring accuracy against annotations in these public databases gives a bias towards the techniques that were used to make the originally predictions that are now annotated in the databases. Thus, there exists a need to better ensure accuracy in gene predictions without having to rely on wet lab verification.

The EcoGene dataset attempts to provide in-depth information about the genes for *E. coli* [7]. Part of this project is the “Verified Set”, which is an attempt to address the problem of unreliable database annotations with which to compare gene prediction results against. The start codon and stop codon locations for each of the 886 genes in the “Verified Set” have been experimentally verified in a wet lab setting. Because of the validity of these gene locations, this dataset has been used as a comparison standard for evaluating many new gene prediction programs [3][4][5][8][10]. From this point on, the 886 “Verified Set” genes will be referred to as the “EcoGene dataset.”

We tested the results of several gene prediction programs, run on *E. coli*, against this EcoGene dataset. GLIMMER 3.02 is a gene prediction program based on interpolated Markov models. MED 2.0 is based on the EDP model and the TIS model. Finally, ZCURVE 1.02 is a gene prediction program based on modeling DNA sequences as their “Z curve representation.” Thus, these three prediction programs offer a good variability in methodology. These programs were also chosen because of their widespread popularity and usage, as well as their relatively good computational speed and space requirements. These test results are the results previously seen in Table 1.

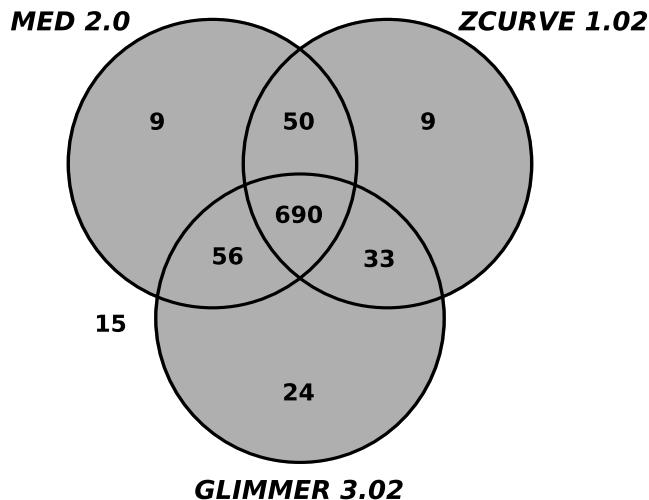


Figure 1: Venn diagram of the correct start location predictions for predictions programs run on *E. coli* using the EcoGene dataset (886 genes) as the comparison standard.

All programs performed at relatively similar accuracy levels, roughly 99% for stop codon prediction and 90% for start codon prediction. What is most striking about these results

is how each program predicted a noticeably different set of gene locations, which is likely explained by their variations in methodology. Figure 1 shows a Venn diagram of which prediction programs were correct for each of the 866 true start codon locations in the EcoGene dataset. Although many of the start codon locations were predicted accurately by all three prediction programs, there was significant overlap in the areas of the Venn diagram for which only two programs overlapped. Thus, there are specific strengths and weaknesses of each prediction program. Between all three, 100% of the stop codon locations and 98.8% of the start codon locations are accurately predicted at least once. This is a substantial improvement over the 99% / 90% average accuracies obtained by these prediction programs on their own. Therefore, there is definitely room for improvement in the area of prokaryotic gene prediction.

3 APPROACH

The goal of our approach is to exploit these interesting overlaps noticed in the Venn diagram of the prediction program results. We seek to take full advantage of the specific strengths and weaknesses related to various pre-existing gene prediction programs, and to combine the results of different prediction programs with the goal of improving overall prediction accuracy. The essence of our approach is to use consistently predicted genes to build a model to help fine-tune the location predictions for other genes in the result sets, specifically by choosing a better start codon location for these “inconsistent” genes. YACOP was an attempt at post-processing results of other gene prediction programs which demonstrated that improvement can certainly be achieved [9]. However, it was simple in design, merely a series of mathematical set operations on the various prediction result sets, and offered no reasonable approach for finding statistical patterns or trends in the predictions themselves. Other post-processors such as RBSfinder and GS-Finder attempted to post-process result sets with the goal of improving their start location prediction accuracy [5][8]. Upon their release, they provided drastic improvement in start location accuracy, but they have since become irrelevant due to the advances in start location accuracy in the prediction programs themselves.

Taking the union of results from other prediction programs can generally improve accuracy, as was demonstrated in the YACOP approach [9]. This is certainly the case for the three prediction programs described previously, where the union of results will find 100% of the EcoGene dataset. However, the problem is in knowing which of the start codons to chose for each gene, since different prediction programs will choose different start codons for the same gene. Thus, improvement can be made by fine-tuning these start codons locations to predict the most accurate one. Our goal is achieved through a combination of majority voting and Naïve Bayesian classification. Genes in which there is some consensus among the prediction programs for both stop codon and start codon locations of the gene are used as a training set to predict a more accurate start codon for the other genes, in which there is some level of disagreement among the prediction programs regarding the true start codon

of the gene. Although in this paper we focus exclusively on GLIMMER 3.02, MED 2.0, and ZCURVE 1.02 for reasons described previously, this approach can easily be applied to combinations of other prediction program results.

To begin, the desired prediction programs are all run on the same genome, each producing their own set of gene location predictions. The resulting set of all these genes are parsed and the votes for individual stop codons and start codons are tallied. Each stop codon receives a vote for every prediction program that predicted the end of a gene at this location. Each start codon receives a vote for every prediction program that predicted the beginning of a gene at this location. Note that every predicted start codon “belongs” to a specific stop codon. Thus, $V_x = \sum_{t \in T_x} V_t$ where V_i is the number of votes for codon i , x is the stop codon, and T_x is the set of start codons “belonging” to the stop codon x . In other words, the votes for all potential start codons for a gene sum up to the votes for its stop codon.

Genes are then grouped into two sets based on the result of these tallies. The first set is the training set, which will later be used to build a model using Naïve Bayesian classification. All genes in which the stop codon has a majority vote tally and in which there also exists a start codon with a majority vote tally are grouped into the training set. Thus, a gene is grouped into the training set if, for its stop codon x , if we have $V_x > \frac{N}{2}$ as well as $V_t > \frac{N}{2}$ for some $t \in T_x$, where N is the number of prediction programs we are using as input. The second set is the classification set. All remaining genes are grouped into this set.

For each gene in both of these sets, we generate a list of records which correspond to every possible start codon for that gene. Each record is classified by a set of attributes describing the potential start codon. These attributes are described in detail in the following subsection. For genes in the training set, each record also includes a class label of “true” or “false” depending on the validity of the start codon. Recall that the training set is the set of genes for which there was a start codon with a majority vote tally, thus this is the start codon that receives the class label of “true.” All other potential start codons for this gene receive a class label of “false.”

The training set, containing records for every potential start codon for every gene in the set, is then used to train a Naïve Bayesian classifier (NBC). Naïve Bayesian classification was chosen for this task because it performed best in a preliminary test of start codon classification, and because, being based on probability theory, it applies well to a biological application, where attributes do not always deterministically determine classes. A trained NBC, given a test record, will output an estimate of the probability of that record belonging to a certain class. This is often expressed as $P(\text{class} = k|A)$, or the posterior probability that a record with attribute set A belongs to class k . In this case, we can use our trained NBC to calculate $P(\text{class} = \text{true}|A_t)$ for every record in the classification set, where A_t is the attribute set for the record of start codon t .

Records for which $P(\text{class} = \text{true}|A_t) > P(\text{class} = \text{false}|A_t)$ do not necessarily signify a true start codon. Note that there may be some genes for which $P(\text{class} = \text{false}|A_t) > P(\text{class} = \text{true}|A_t)$ for every record. Likewise, there may be some genes for which

"Classification Set"	
	Gene 1
Gene 2	Start Codon 1.1 Start Codon 1.2 Start Codon 1.3 Start Codon 1.4 Start Codon 1.5 Start Codon 1.6 Start Codon 1.7 Start Codon 1.8
	Classify one record as "true," all others as "false."
	Gene 2
Gene 2	Start Codon 2.1 Start Codon 2.2 Start Codon 2.3 Start Codon 2.4 Start Codon 2.5 Start Codon 2.6 Start Codon 2.7 Start Codon 2.8
	Classify one record as "true," all others as "false."
	Start Codon 3.1 Start Codon 3.2 ...
	...
	...

Figure 2: Visualization of how the records in classification set are organized by genes

$P(\text{class} = \text{true}|A_t) > P(\text{class} = \text{false}|A_t)$ for more than one record. To address this, records are evaluated (and classified) within the context of their gene. Each gene must have one record that is classified as “true” while all others are classified as “false.” For each gene in the classification set, the single most probable start codon is chosen based on which record for a gene has the highest value for $P(\text{class} = \text{true}|A_t)$. Figure 2 shows a visualization of this organization of the classification set.

All genes in the classification set now have one record classified as “true” and the rest as “false,” just like in the training set. The final set gene of predictions is the union of these two sets. This reports, for each gene, its stop codon location as well the start codon location corresponding to the record classified as “true.”

3.1 Attributes

For each potential start codon, we seek to create a set of simple yet biologically meaningful attributes. A total of 44 attributes were selected, which fit into one of the three categories listed below.

Input: Results of N prediction programs, each with a list of predicted gene locations (1 start codon and 1 stop codon)

Output: Improved list of start codons and stop codons for every predicted gene

```
foreach prediction program result set  $P$  do
    foreach gene  $G$  in result set  $P$  do
        Add 1 to tally  $V_t$  for start codon  $t$  of  $G$ 
        Add 1 to tally  $V_x$  for stop codon  $x$  of  $G$ 
    end
end
foreach gene  $G$  do
    if  $V_t > N/2$  and  $V_x > N/2$  for start codon  $t$  and stop codon  $x$  of  $G$  then
        Generate list of potential start codons for  $G$ , with corresponding attributes and
        class label
        Add  $G$  to classification set
    else
        Generate list of potential start codons for  $G$ , with corresponding attributes
        Add  $G$  to testing set
    end
end
Train NBC with classification set
foreach gene  $G$  in training set do
    foreach start codon record  $t$  for  $G$  do
        Use NBC to calculate  $P_t = P(class = true | A_t)$  where  $A_t$  is the attribute set for  $t$ 
    end
    Classify  $t$  with highest  $P_t$  as “true”, all others as “false”
end
Output the union of classification set and training set, reporting the stop codon and
“true” start codon for each gene
```

Algorithm 1: Algorithmic description of our approach

Location - The *positionRating* attribute is a score that represents how far upstream the start codon is located at, relative to the other potential start codons for the gene. This attribute is based on the length variable in [5] which is defined as $length\ variable = e^{-\frac{l}{l_0}}$, where l_0 is the nucleotide length between the stop codon and the most upstream potential start codon for the gene, and l is the nucleotide length between this potential start codon and the most upstream start codon. The length variable described in [5] varies on a scale between $1/e$ and 1. For our *positionRating* attribute, we modified this to scale between 0 and 1, to remain consistent with our other attributes. To do this scaling, the expression becomes $position\ Rating = e^{-\frac{l}{l_0}} - \frac{1-e^{-\frac{l}{l_0}}}{e-1}$. Thus, the length variable is higher for potential start codons that are further upstream, with a length variable of 1 for the most upstream start codon. The theoretical minimum *positionRating* value would then be 0. The exponential curve used allows differences in upstream position to carry more significance than differences in downstream position, which makes it biologically meaningful since it has been shown that true start codons are often the most upstream of the potential start codons for the gene [1][6].

Codon usage - It has been shown that different organisms have different tendencies in start codon usage. For example, *E. coli* strongly favors the ATG start codon, but also has significant usage of the GTG and TTG start codons [2][6]. *B. subtilis* also favors ATG, but has a relatively stronger usage rate of GTG and TTG than *E. coli* does [6]. Because of the variability between organisms and start codon usage, we choose to include binary attributes for each of these possible start codons to allow for maximum flexibility in model building. Since we deal exclusively with prokaryotes, there will be three binary attributes, one for each of ATG, GTG, and TTG.

Nucleotide distribution - Rather than include attributes related to some sort of promoter region detection or ribosomal binding site (RBS) detection, we chose to include attributes for the nucleotide and nucleotide-pair distribution both downstream and upstream of the potential start codon. Like start codon usage, specific patterns for promoter regions and RBSs can vary between organisms. We hypothesize that these nucleotide distribution attributes will provide a more flexible way of detecting relevant patterns, both upstream and downstream of the start codon location. This brings with it the added advantage of not needing to know information about the promoter region or RBS for the specific organism beforehand, which helps to further automate the gene prediction process. For our attributes, we examined the region of 30 nucleotides both upstream and downstream of the potential start codon, and calculated the percentage nucleotide distribution for each of the four single nucleotides, and the percentage nucleotide distribution for each of the sixteen possible nucleotide-pairs. These attributes will be referred to as *upA* or *downCA* for example, which refer to the percentage of nucleotides upstream that are Adenine, and the percentage of nucleotide pairs downstream that are Adenine-Cytosine, respectively. Some initial data exploration showed that our hypothesis was indeed correct, and that these attributes provided significant information about the validity of a potential start codon. For example, for genes in the EcoGene dataset the average value of the *upC* attribute was 0.188 and the *upA* attribute average value was 0.332 for true start codons, but were 0.253 and 0.244 respectively

for false start codons. In a random sequence of DNA, these attribute would be expected to about 0.250, as was observed for the false start codon upstream regions. Thus, this shows that the upstream promoter region and/or RBS seem to have higher concentrations of adenine and lower concentrations of cytosine, which our nucleotide distribution attributes will notice.

4 EXPERIMENTAL RESULTS

The implementation of our approach used the gene prediction programs discussed earlier for the input prediction set: GLIMMER 3.02, ZCURVE 1.02, and MED 2.0. Again, these prediction programs were chosen because of their widespread popularity, favorable execution speed, and variability in methodology as well as prediction results. All results were analyzed by comparing to the EcoGene dataset. In order to observe how our Naïve Bayesian classifier is performing, we evaluated the results in two parts. First, we observe the results of the first part of our approach, which was to partition the predicted genes into the training set and the classification set. The percentage of the EcoGene genes in the training set is shown in Table 2. The training set included 97.3% of the EcoGene genes, with the other 2.7% being partitioned into the classification set. Note that this means 100% of the genes from the EcoGene dataset are found when we take the union of these the prediction programs. In other words, every gene is found by at least one of these prediction programs, and thus every gene will show up in the final prediction results of our approach, which is an improvement over the approximately 99% of genes found by each program individually.

Stop Codons	Start Codons
97.3%	93.6%

Table 2: Percentage of the stop codons and start codons from the Eco-Gene dataset that is included as part of our training set.

Table 2 also shows that 93.6% of the EcoGene genes are in the training set with their true start codon. Since the criteria for being partitioned into the training set is that a stop codon as well as a start codon have majority vote tally between all the prediction programs, this implies that there are several genes from the EcoGene dataset (4.1%) for which there is majority agreement between the prediction programs on a stop codon but also majority agreement on a start codon that is not actually the true start codon. Thus, our approach will be unable to “fix” the start codon for these genes. Thus, our Naïve Bayesian classifier is working to improve the accuracy for the 2.7% of true genes that are in the classification set. This set of genes will be referred to as the set of “ambiguous genes” because these are the genes for which there was variation in predicted start codon location between the prediction programs. Our results will be evaluated as the percentage of this set of “ambiguous genes” that we were able to determine the correct start codon for.

	All Attributes	No Double-Nucleotide Distribution Attributes	No Linearly Dependent Distribution Attributes
All Start Codons	41.6%	45.8%	45.8%
Only Selected Start Codons	41.6%	45.8%	58.3%

Table 3: Percentage of the “ambiguous genes” for which our classifier was able to detect the correct start codon under varying parameters. No “Double-Nucleotide Distribution Attributes” removes the attributes of the form *downNN* and *upNN* where *N* is any nucleotide. “No Linearly Dependent Distribution Attributes” removes the same attributes, plus the *upT* and *downT* attributes. “Only Selected Start Codons” only considers the start codons for which at least one prediction program selected.

In our tests, we varied the attributes used in defining start codon records, as well as the start codons that would be considered in the classification process. We ran our approach using all attributes discussed above, and again with only the *positionRating* attribute, the codon usage binary attributes, and the single nucleotide distribution attributes. This was to see how meaningful the nucleotide-pair distribution attributes were. We also ran our approach without the nucleotide-pair distribution attributes and the *upT* and *downT* single nucleotide distribution attributes. This is because these distribution attributes were linearly dependent (i.e. $upT = 1 - upA - upC - upG$). Theoretically, no new information can be gained by including all four as opposed to all three. The thymine nucleotide was chosen at random as the attribute to remove for this purpose. We also varied the number of start codons to choose between in the classification process. In one set of runs, all of the potential start codons for a particular gene were examined. In the other set of runs, only those start codons that at least one of the original prediction programs chose were considered.

The results of these runs can be seen in Table 3. Our best result was able to correctly identify the true start codon for 58.3% of the “ambiguous genes.” Results were fairly consistent between the runs that used all potential start codons, and the runs that used only the start codons that were predicted by at least one prediction program. This demonstrates how our classifier reliably identifies the start codons that we should be most strongly considering.

The accuracy of our overall approach is shown in Table 4, with comparisons to the accuracy of the original prediction programs. Note that since we are comparing a larger set of predicted genes against EcoGene, which is only a subset of all the true *E. coli* genes, the performance measure reported is simply the percentage of EcoGene stop codons and start codons that are found.

Prediction Program	Stop Codon Accuracy	Start Codon Accuracy
GLIMMER 3.02	99.5%	90.6%
MED 2.0	99.0%	90.9%
ZCURVE 1.02	98.8%	88.3%
Our Approach	100.0%	95.1%

Table 4: Percentage of stop codons and start codons from the EcoGene dataset found by the original three prediction programs, run on *E. coli*, as well as the results of our approach.

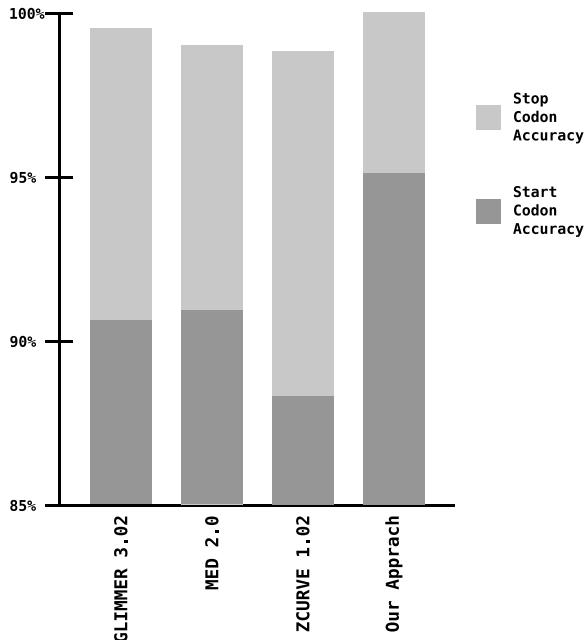


Figure 3: Bar graph of the percentage of stop codons and start codons from the EcoGene dataset found by the original three prediction programs, run on *E. coli*, as well as the results of our approach.

5 CONCLUSION

Using our Naïve Bayesian classifier, we were able to correctly identify over half (58.3%) of the true start codons for the “ambiguous genes,” those genes that did not have consistent location predictions between the original prediction programs. Thus, our classifier was able to successfully improve reliability in start codon predictions and help discern which

program predicted the true start codon location. Our approach as a whole was able to post-process a set of gene location predictions and improve accuracy, from the roughly 99% (stop codons) and 90% (start codons) accuracies of the original predictions, to the 100% (stop codons) and 95.1% (start codons) accuracy of our final gene location predictions.

Note that the results discussed here are for a specific organism (*E. coli*) and with using specific original prediction programs (GLIMMER 3.02, ZCURVE 1.02, and MED 2.0). However, in keeping the concepts of our approach as general as possible, we hope that the same ideas can be applied to other domains to improve accuracy of gene prediction, especially in detecting the true start codons. We recommend this as an area for future study.

In keeping our attribute set as general as possible, our goal was to make this approach easy to apply to other organisms. However, more improvements can likely be made in choosing a better set of attributes to use. Since accuracy for Naïve Bayesian classification can be hurt by irrelevant or correlated attributes, further study should be done to chose a non-correlated, highly meaningful set of attributes that would likely lead to better results.

Overall, the approach discussed here shows that post-processing techniques using Naïve Bayesian classification can be effectively used to improve the accuracy of gene location predictions, especially for start codons.

References

- [1] J. Besemer, A. Lomsadze, and M. Borodovsky, “GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions,” *Nucleic Acids Research*, vol. 29, pp. 2607-2618, 2001.
- [2] F.R. Blattner, G. Plunkett III, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, et al., “The Complete Genome Sequence of *Escherichia coli* K-12,” *Science*, vol. 277, pp. 1453-1462, 1997.
- [3] A.L. Delcher, K.A. Bratke, E.C. Powers, and S.L. Salzberg, “Identifying bacterial genes and endosymbiont DNA with Glimmer,” *Bioinformatics*, vol. 23, pp. 673-679, 2007.
- [4] F.-B. Guo, H.-Y. Ou, and C.-T. Zhang, “ZCURVE: a new system for recognizing protein-coding genes in bacterial and archaeal genomes,” *Nucleic Acids Research*, vol. 31, pp. 1780-1789, 2003.
- [5] H.-Y. Ou, F.-B. Guo, and C.-T. Zhang, “GS-Finder: A program to find bacterial gene start sites with a self-training method,” *The International Journal of Biochemistry & Cell Biology*, vol. 36, pp. 535-544, 2004.

- [6] S.S. Hannenhalli, W.S. Hayes, A.G. Hatzigeorgiou, and J.W. Fickett, “Bacterial start site prediction,” *Nucleic Acids Research*, vol. 27, pp. 3577-3582, 1999.
- [7] K.E. Rudd, “EcoGene: a genome sequence database for Escherichia coli K-12,” *Nucleic Acid Research*, vol. 28, pp. 60-64, 2000.
- [8] B.E. Suzek, M.D. Ermolaeva, M. Schreiber, and S.L. Salzberg, “A probabilistic method for identifying start codons in bacterial genomes,” *Bioinformatics*, vol. 17, pp. 1123-1130, 2001.
- [9] M. Tech and R. Merkl, “YACOP: Enhanced gene prediction obtained by a combination of existing methods,” *Silico Biol*, vol. 3, pp. 441-451, 2003.
- [10] H. Zhu, G.-Q. Hu, Y.-F. Yang, J. Wang, and Z.-S. She, “MED: a new non-supervised gene prediction algorithm for bacterial and archaeal genomes,” *BMC Bioinformatics*, vol. 8, p. 97, 2007.