Promoter Prediction Based on E. coli Characteristics

Li Wen

Follow this and additional works at: http://scholarworks.sjsu.edu/etd_projects

Recommended Citation
Wen, Li, "Promoter Prediction Based on E. coli Characteristics" (2012). Master's Projects. 216.
http://scholarworks.sjsu.edu/etd_projects/216

This Master’s Project is brought to you for free and open access by the Master’s Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master’s Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.
This project uses the characteristic in TATA-less regions on E. coli sequences to predict the promoter region before TSS, which indicate that the real gene has been located. It uses several well-known algorithms and methods such as the sliding window algorithm, and a clustering method to predict promoters. It also contains D2K algorithm and method to compare predicted result with other online promoter package result.
# Table of Contents

Introduction .....................................................................................................................2
Theory Testing (Approach) .................................................................................................4
The D2K Algorithm ........................................................................................................18
Implementation ...............................................................................................................21
Result ...............................................................................................................................22
Conclusion ......................................................................................................................31
Reference .......................................................................................................................31
Appendix 1—Training Data ..............................................................................................34
Appendix 2—List of NCBI Accession Number of Testing Data ........................................35
Appendix 3—D2K Algorithm Coding in EGF .................................................................36
Appendix 4—Result Comparison .....................................................................................41
Appendix 5—Web Application .........................................................................................61
Introduction

Recent promoter predictions
A promoter is a piece of DNA sequence which frequently appears before its associated gene in an E. coli sequence. Researchers predict the location of a promoter when trying to locate a gene in a given sequence. These researchers use a variety of algorithms to predict locations of promoters. These algorithms include machine learning, artificial neural network, the Markov model, the weight matrix, etc. Abeel et al apply machine learning [10][13] and Bland et al use artificial neural networks [8] to classify and output the predicted promoter location of any given unknown sequence. Burden et al apply the weight matrix algorithm [5] to identify motifs in the promoter region. Burden et al also use Markov model [5] to find the shortest path to the promoter location. Most of the researchers use the combination of artificial neural network, machine learning, and Markov model to increase the true positive prediction results [8].

Some researchers provide additional information when predict promoter location. For example, Gan et al introduced the idea of using non-CpG [6] region information as CpG region for prediction. Gan et al found the equal significance of non-CpG region in their experiment. Burden et al use the distance between TLS and TSS for promoter prediction [5]. The distance between TSS and TLS provide additional clues of promoter location which can increase the true positive of prediction. Davuluri et al are interested in finding the first exon [4], since it is the most difficult promoter location to find. First Exon Finder—FirstEF [4] uses CpG information to find the first donor site for first gene exon predict.

Data used for promoter prediction varies. Wang et al use the comparison of human and mouse genome for prediction [14] [4]. Laser et al use mammal and plant genome for prediction [26]. Most of the data used in promoter prediction is E. coli [8], especially E. coli K12 for its promoter richness [30]. In my research, I used E. coli from NCBI [30] database and plant data that I retrieved form plantDB [3]. Plant data are only used for testing and training purpose.

The researchers discussed above using their algorithms and data were able to reliably predict the location of a promoter.

The lack of using TATA-less regions in researches
TATA box is a piece of DNA sequence that usually appears in promoter region. Originally, it has been used to locate promoter locations [2]. However some researchers prefer not use TATA box as a signal of the promoter region. Burden et al state that the TATA box is not an effective resource for promoter prediction [5]. The data sequences they used in their experiment are mixed with TATA-rich, and
TATA-less promoters; their analysis shows no strong relation connects TATA-less regions with promoter regions. For example, in the paper “Improving promoter prediction”, Burden et al said that the characteristic of TATA box is limited and cannot be used to recognize when indel happens [5]. Also, the definition of TATA box says that less than 20% of human promoters have TATA box, which leads the rest of the promoter regions unsearchable by using TATA box as searching factor [21]. Burge et al show in their experiment only 70% of core promoter contains TATA box [23]. Therefore, most researchers ignored a large amount of available data—TATA-less region during promoter prediction process.

**Motive of using non-TATA region**

The idea of using non-TATA region for promoter prediction comes from one of the research papers—“A pattern-based nearest neighbor search approach for promoter prediction using DNA structural profiles”. In their research, Gan et al discovered that non-CpG-island region contains similar characteristics with CpG-island region [6]. Gan et al researched CpG-island region by compute the gravity of CpG in every promoter region in given sequences. Researchers did the same calculation for non-CpG-island region. The results show that non-CpG-island region provide equally important prediction information. For any given unknown sequence, the promoter can be predicted by using both CpG-island and non-CpG island classes to increase prediction result.

In my research I will use information on non-TATA regions for promoter prediction. TBP (TATA binding protein) is used to bind sequences with TATA box, and I am looking for the regions in TATA-less sequences that will be bind with other binding proteins in TFIID (transcription factor II D) [22]. Promoter regions usually contain TFIIB recognition element (BRE), TATA box, Inr, and downstream promoter element (DPE). Most of promoters miss one of these elements, and for promoters that do not have TATA box will have a high probability of having DPE [23].

**State of theory**

Thus far, no research has focused on using non-TATA regions for promoter prediction. TATA box is used for promoter prediction in many papers for sequences contain TATA box. For sequences without TATA box information, existing algorithms cannot do much analysis in the data mining step of the prediction. Sequences without strong TATA box information are basically ignored during data mining step.

In this paper, I studied the characteristic of non-TATA or less-TATA region. I found the characteristic in TATA location of non-TATA regions, to get clues as to predict promoter region based on both TATA and non-TATA information. The model of this project was trained and acts like a polymerase. Polymerase does not use TATA box regions, but it was based on the structure or chemical statement of TSS in order to open the double helix, and start coping gene. This initial location indicates that a real gene is many nucleuses away, and waiting to be copied. Finding the initial location
will be used as characteristic of the TATA box in this paper. These characteristics will be coded in the model to detect any given E. coli sequence from NCBI.

I hope to show that the predicted gene location will be close to real gene location in NCBI database, or get more true positive values than several popular online prediction tools. These tools include EasyGene, GenScan, Virtual Footprint, and Glimmer (the original prediction that used by NCBI).

### Theory Testing (Approach)

#### Software and algorithms used during theory testing step

For this research I used a combination of commercial and custom software. The commercial software is SAS—a well-known business analytical software. I’ll first try to find the significant of non-TATA region information by using SAS. For this research, I also created a custom Perl program—called E. coli Gene Finder (EGF), which encodes machine learning, data mining, five folds, and the artificial neural network algorithm. I used this software combination to find the most important characteristics of the non-TATA region in my training data. I also use these algorithms to find the threshold which define the decision tree. The EGF is adjusted in the theory testing step to filter out a promoter sequence out of any given E. coli sequences. The steps of using SAS for sequences are listed in the steps section below. During model training step, fivefold method will be used.

#### Data set used in theory testing step

The data set I used for training my model purpose is plant promoters that are obtained from PlantProm DB [7]. 170 of them are TATA rich plant sequences and 130 of them are TATA-less plant sequences. Once the significance of non-TATA information is discovered by using SAS and my program, I’ll use E. coli data sets for both my training and testing data in the machine learning step. The reason to choose promoter data set from PlantProm DB are: first, they provide a clear plant data for both TATA rich promoter region, and TATA-less promoter region. I choose the plant data with the intention that the theory could be applied to other species.

#### Pretest on TATA region

The basic characteristic of TATA box is it matches the expression 5’-T A T A (T/A) A (T/A) ---3’ Data in TATA rich promoter are input in an online promoter prediction service—BDGP [3]. It is a neural network promoter prediction web service that developed by Berkeley Drosophila Genome Project. It uses the combination of a neural network and weight pruning to search for consensus elements, such as TATA box, CpG Island, CAT box, etc. When testing 170 TATA rich sequences, TATA rich sequence’s characteristic is clearly displayed in the similar location of each of
submitted sequence—25 base pair upstream of transcription start site. None TATA rich sequences—130 of them, also have been tested by using the same service, but no characteristic are identified. The target of using SAS is to find some valuable information around the location where TATA usually appears; since in reality, when polymerase is walking along on the DNA double helix, it will find the initial point not the letter TATA inside promoter region.

Steps of finding characteristic
Data of plant promoter are obtained from PlantProm DB [7] to test my hypotheses.
Steps of finding TATA and non-TATA data:
1. Get both TATA promoter and TATA-less promoter data from PlantProm DB
   1.1 Base URL: http://mendel.cs.rhul.ac.uk/mendel.php?topic=plantprom
   1.2 175 TATA rich location: http://mendel.cs.rhul.ac.uk/pprom/PLPR_TATA.seq
   1.3 130 TATA less location: http://mendel.cs.rhul.ac.uk/pprom/PLPR_TATA-less.seq
2. Filter out the known promoter parts from both data sets
   2.1 Write EGF (Perl program) to filter the two original data sets
      2.1.1 Change the input and output file names of both data files
      2.1.2 Delete the unwanted original data file
   2.2 Extract the promoter parts
   2.3 Output two generated promoter data sets
      2.3.1 Match all the word character lines in the file
      2.3.2 Go to a new line for a new sequence
      2.3.3 Both TATA rich and TATA less sequences were processed
3. Generate statistics on both promoter sets
   3.1 Perform a relation test on TATA promoter data set
      3.1.1 Find the average location of TATA box for both “TATA” and “TATAA”
         3.1.1.1 Get TATA location in each sequence
         3.1.1.2 Get average location by using below formula:
            \[
            \text{average} = \frac{\sum \text{locations}}{\text{total number of sequence}}
            \]
            where locations are cutting off from 160 since less than 160 will be out of promoter region. Average = 171. This formula was coded and tested inside the main Perl program in data testing section. The results are given in Figure 1 and Figure 2.
3.1.2 Find average location of TATAAA box

3.1.2.1 Get TATAAA location in each sequence

Figure 1. Test result for TATA location for TATA rich promoters in file TATApromoter175.txt. First column is the sequence number from 1 to 175, followed by the TATA location that is detected by the program in that sequence. Location number less than 160 is ignored since it is out of promoter region. The average of remaining sequence that matches promoter region definition is calculated as 171.08333.

Figure 2. The graph shows that the TATA location (in green) is almost at the same location for each sequence. The capitalized sequence is the start of TSS.
3.1.2.2 Get average location by using below formula:

\[
\text{average} = \frac{\sum \text{locations}}{\text{total number of sequence}}
\]

where locations are cutting off from 160 since less than 160 will be out of promoter region. Average = 174. Results are given by Figure 3 and Figure 4.

![Figure 3. Test result for TATAAA location for TATA rich promoters in file TATApromoter175.txt. First column is the sequence number from 1 to 175, followed by the starting point of TATA location in that sequence. Location number less than 160 is ignored since it is out of promoter region as before. The average TATA start point is calculated as 173.8556](image1)

![Figure 4. The graph—from program notepad, shows that the TATAAA location (in green) is almost at the same location for each sequence with several nuclides length difference. The capitalized sequence is the start of TSS.](image2)

3.1.3 Result: the location of TATA box in these plant sequences is around 170, since the data starts from -200 of TSS, the relative location of TATA to TSS will be around -30 (-30=-200-170). This confirms that promoters containing TATA box are right before TSS, around -30
3.2 Perform the same test on TATA-less promoter data set (TATAlessPromoter130.txt)

3.2.1 Theory: Get sub-sequence from -35 (get from -30-5 above) with length 20 out of each sequence. The idea of getting the information out of TATA-less sequences on the same location where TATA box appear in TATA-rich sequence is the motivation of polymerases. The polymerase has to be initialed before face the real gene; and that initialization must happen on the similar location of each sequence, and it doesn’t matter whether it contains TATA or not. Therefore, this section will get other characteristics in addiction to TATA box out of TATA-less sequence. The finding will be used to predict gene in TATA-less sequences.

3.2.2 Find characteristic

3.2.2.1 It cannot be done by using the same approach since it is TATA less sequence. TATA are all over the sequences. It is shown in Figure 5.

3.2.2.2 Get the promoter region (-35, -1) with TSS region (+1, 40), and find characteristics from the same data by using SAS, where all characters are converted to numbers, [a,c,t,g,A,C,T,G]/[1,2,3,4,5,6,7,8]. SAS is the data analysis software, it can tell information out of given data. Next section will try to get characteristic by using SAS.

3.2.3 Relational test by using SAS to get characteristics

3.2.3.1 Pearson correlation coefficient is tested between promoter region (X1-X35) and TSS region (X36-X86). Pearson tests are done between every pair of column. Some of the columns are related with correlation value bigger than 0.9, such as the correlation between X17 and X37 is 0.9783.
3.2.3.2 Given X17, what is the value for X37? In other words, how to predict X37 of TSS region by using X17 of promoter region? We need to find a line that best fits the regression, so that the regression testing is done by using dependent variable X37 and independent variable X17 to minimize the sum of squared vertical distance from each data point to the line (residual). The formula we used is:

\[ X_{37} = a + b \times X_{17} \]

where “a” is intercept and b is slope. The dependent variable—b is not significant enough to predict the independent variable X37.

3.2.3.3 Similar test for codon—3 nucleotides a pair, instead of single nucleotide are also finished. No significant results are found between codons.

3.2.3.4 Similar test for first TSS codon of each single nucleotide contains a percentage of the promoter regions which means testing a nucleotide in the first TSS codon that appears in the promoter region. Half of the total length of the promoter region (33/2=17) is used to prune noisy data. Noisy data is defined as outstanding data—the first 3 nucleotides of TSS is less than half of the total promoter nucleotide. Steps to calculate first 3 TSS single nucleotides are list as below.

a) Get number of each of the first 3 nucleotides of TSS in promoter region. Such as $aa =$ number of first nucleotide used in promoter region.

b) Sum them for each sequence. $dd =$ sum of all three nucleotides usage.

c) Get percentage of above sum = $dd /$ promoter region length

d) Sum the percentage for all sequences with total usage bigger or equal to half of the promoter length. $total = \sum \text{for every } dd \geq 17.$

e) Take the average = $total /$ number of sequences with $dd \geq 17.$

3.2.3.5 Test Result.

3.2.3.5.1 The result shows that most of the promoter region contains more nucleotides that belong to the first 3 TSS. In this case 130 TATA-less sequences are tested, 97 of them contains high usage of the first 3 TSS with average usage equal to 0.7085.
3.2.4 Conclusion: From this experiment we can see that even though the sequences don’t have TATA box as characteristic to signal promoter region, the nucleotide usage in promoter region can be used as a clue of TSS region. From Figure 6, sequence 128, it clearly shows that the ‘tga’ usage is 31 out of 33, there is only one ‘c’ used before TSS. This finding can be used as a secondary characteristic to indicate a real gene in TATA-rich sequence. And for the same reason, it can be used as the main characteristic to predict real gene from TATA-less sequence.

4. Analyze the results and get characteristic for both data set
   4.1 For TATA rich sequences, use TATA as consensus sequence to identify the promoter region.
   4.2 For TATA less sequences, use first 3 TSS nucleotides to signal the TSS position. Since the finding depends on individual nucleotides, it is not very significant, therefore it cannot be used as a characteristic to locate promoter region in general.

5. Working with the data on protein level
   5.1 Translate both TATA rich and TATA-less sequences into amino acids equivalents and see if any significance appears up in the next attempt. The idea behind this transition is when polymerase bind to the TATA boxes, it is not attracted by the nucleotides of TATA or TATAAA, and instead it is attracted by the product of the nucleotides. Therefore, if polymerase can bind to the promoter region without TATA box as common notation, then it needs to find similar protein to bind. The TATA-less region needs to have such a protein to attract polymerase for initiation of TSS.

   5.2 Test the TATA rich sequences without using TATA as characteristic
      5.2.1 Translate each sequence to its amino acid equivalent by using hash table [1]. In order to find the relationship between the first amino acid with a promoter region, use the number to represent amino acids instead of the real protein name[15][16].
      5.2.2 Clean the data by filtering all sequences containing at least a quarter
of the first amino acids of the TSS region; and the resulting data after this step looks like Figure 7 below. Only 142 out of 175 met the requirement.

Figure 7. TATA rich file numerical protein product representation

5.2.3 Find the location that is most closely associated with the first amino acid in TSS by using the sliding window algorithm [1]. For each 10 amino acid (because the length of TATA box) in each sequence, the maximum appearance of the last number in this sequence will be counted; and the middle location of that maximum number will be calculated by using below formula:

Peak of the first aa in TSS in promoter region = Max (appearance in each 10 amino acid)

Location of the max = the peak location + 5; 5 means set the location to the middle of the window, since the sliding window size is 10.

5.2.4 Thresholds: data will be cut if it does not meet the thresholds in two conditions. If the maximum total appearance is less than 6 (half of the sliding window) and the location is not close to TSS region. In this case the threshold for location is 33, which is 32 amino acids long to TSS; otherwise, the location of such peak cannot be characterized. Below is the running result with above thresholds.
Figure 8: Test result of the max appearance and the average location in TATA rich file. The average peak in one sequence contains about 72% of the first amino acid of TSS. The peak appears around the 50th amino acid, which is very close to the TATA box location examined before.

5.3 Search for similar characteristic in TATA-less file, so this can be used to detect promoter region for any given unknown sequence.

5.3.1 Translate the TATA-less file into its protein equivalent [15] [16].

5.3.2 Clean the data by filtering all sequences containing at least a quarter of the first amino acid of TSS region; and the result data after this step is represented in Figure 9 below. Only 107 out of 130 meet the requirement.

Figure 9. TATA-less file numerical protein product representation

5.3.3 Find the location that is most associated with the first amino acid in
TSS by using the same algorithm and formula as in 5.2.3. Test result on TATA-less file is showed in figure 10 below.

![Figure 10. Test result of max appearance and average location in TATA-less file. The average peak in one sequence contains about 73% of the first amino acid of TSS. The peak appears around the 51th amino acid, which is very close to the results in TATA rich file.](image)

5.4 Discussion

Further testing needs to be done by using the five folds method [2]. We randomly select 20% of sequences from TATA-less file, and test if the finding will give the right TSS position.

5.4.1 Test TATA rich sequence by using five folds

5.4.1.1 Five Fold run results

![Figure 11. Randomly choose 80% of data out of 175 TATA rich sequences, and run the same program. The maximum of first aa is about 7.15, and average location is around 50.](image)
Figure 12. Random choose 80% data out of TATA rich sequences again, and run it again. The maximum of first aa is about 7.22, and average location is around 51.

5.4.1.2 Discussion
Based on the above two observations by using the five fold method, we can see that the results are similar to what we get by testing all 175 sequences. It tells that the maximum of first aa is about 7, and the peak location is around 50 in TATA rich sequences.

5.4.2 Test TATA less sequence by using five folds
5.4.2.1 Five Fold run results

Figure 13. First run against TATA less sequences by using five fold methods. The maximum number of first aa is about 7.3, and the average location is about 50.6.
5.4.2.2 Discussion

Based on above two tests against TATA less sequences by using Five Fold method, we can see that it shows the similar maximum number of first aa and average location with the results getting from all 130 sequences.

5.4.3 Conclusion

If we use these findings to detect the promoter region in either TATA rich or TATA-less sequences, then the first step is to partition the data in two classes\[2\] \[17\]. Sequences with the clear TATA box characteristic will go to the TATA rich class. And sequences without such characteristics will go to the TATA less class. In TATA rich class, use TATA box to find the location for promoter and TSS. In TATA less class, use the finding to detect the promoter region.

Since the finding in both TATA rich and TATA-less file is very close, then no classification will be needed, which means use the finding directly.

Which method will provide the most accurate result? Compare the two results; also compare the results with online tools. For E. coli data that needs to be used later, ORF needs to be found first.

6. Use the finding in section 5 to test the prediction accuracy on both TATA rich and TATA-less files by using five-fold method \[2\]. Using 4/5 as training data, and the rest 1/5 as test data to test the prediction accuracy.

6.1 Test the finding in TATA rich file

6.1.1 Random select 1/5 to be testing data, and 4/5 as training data among
TATA rich file.

6.1.2 Input 4/5 data to get the location of the peak value of the first TSS codon. This was covered in section 5.4.1.

6.1.3 Use the location to predict the 1/5 data’s TSS and compare it to the real TSS of each and get the accuracy. See the test result below by using 20% of given TATA rich data.

![Figure 15. Test result of 1/5 of 175 TATA rich data. It is similar with 4/5 and all TATA rich data. Therefore the maximum number and the average location can be decided as 7 and 50.](image)

6.2 Test the finding in TATA-less file

6.2.1 Random select 4/5 of TATA less sequences as training data. Please see the results 5.4.2.

6.2.2 Test 1/5 out of 130 TATA less data.

![Figure 15. Test result of 1/5 of 175 TATA rich data. It is similar with 4/5 and all TATA rich data. Therefore the maximum number and the average location can be decided as 7 and 50.](image)

6.2.3 Discussion
The results are close to what we expected, which means we can use the characteristic in real E. coli data from NCBI.

6.3 Conclusion

The characteristic found in both TATA-rich and TATA-less sequence provide similar results as show in this section. The results indicate that gene can be predicted by using first “aa” to detect the start of a gene in about 50 nuclides away from the peak. The algorithm will be discussed in the Algorithm section.

7. Signals or characteristics that can be used for promoter detection.

7.1 Ideas that been using in this project

7.1.1 Inr: finding Inr without DPE will provide the same result with finding Inr with DPE, since TBP will bind Inr with DPE for the lack of TATA box. The consensus sequence of Inr is PyPyAN(T/A)PyPy, where Py is pyrimidine (C or T), N is any base (A, C, G, T), the underline A is TSS [23].

7.1.2 DPE: downstream promoter elements located about 30bp downstream of TSS with consensus sequence G(A/T)CG in Drosophila when there are no TATA box.

7.1.3 Start Codon: in 5-10% of cases, the initiator will pass the first start codon, and use the next one [23 page 539]. A hair-pin before AUG will make this AUG a start codon. This information can be used to justify the start codon location in the late of the process. When AUG at the beginning of mRNA, it is start codon; if AUG in the middle of mRNA, it codes for methionine.

7.1.4 T here are only 15 TFIIBs, each will bind to different sequence in promoter region. If I can find what they bind in TATA less region, then I’ll be able to locate promoter region. The chemical reaction of TBP with TATA box is explained in [25].

7.1.5 C C box: upstream of TATA box are GC box with GGGCGG and CCGCCC in -47 to -61 and -80 to -105 region[23].

7.2 Ideas that can be used in for other researchers

7.2.1 Use RNA secondary structure to find correlations in sequences, need more test to support this idea

7.2.2 TBP (TATA binding protein) binds to the minor groove of TATA box, and then other element of TFIIID may bind to region without TATA box [23]. Compares to other steps like A-U, or G-C, T-A is much easier to distort to initial the transcription. TBP works on both TATA rich and TATA less promoters; TBP is not TATA sensitive, but temperature sensitive [23]. Substituting C for T and I for A in the sequence will get the same result since the minor groove of C and I is the same as with T and A. “What about the promoter that lack a TATA box?” [23]. TBP will bind to initiators, DPEs, or GC box to secure TFIIID’s functionality with the help from TAF (TBP associated factors) 150 and TAF 250. According to figure 11.13[23], TBP will find either TATA box, Inr with DPE, or GC box to bind on the sequence; therefore, there will be three clusters with
7.2.3 Polymerase melting DNA based on Figure 11.5 can also be used to detect promoter region. Polymerase trying to find the weak connection to melt, for any T-A pair it passed, it will provide 25% damage (from TATA, and ATAT it hits four same base pair, and it melts the double strand). Without TATA box, in any location, if the damage adds up to 1, then it will melt the DNA. For example, in Inr, if the sequence is TTAGTT, then the pairs for Inr will be AATCAA, so the calculation will be 25%+25%+25%-25%+25%+25% = 1; then the Inr is melt by polymerase. AT rich region that located before TSS in the promoter region is important as CG contents; it acts as enhancer [26]. So AT-rich can be used to locate the promoter without TATA box; or TATA box is just part of AT-rich. In progress, program is partially done.

7.2.4 Shine-Dalgarno sequence AGGAGGU [23]: after TSS and before start codon, it will attract ribosomes to the nearby AUG to start translation. Eukaryotes do not have a SD sequence, but use a cap called eIF4E at the 5’ end, that help attract ribosomes [23].

7.2.5 Testing E. albertii: genes are overlapping, for example ealbertii1.txt with gi number 169405087 contains many genes with end location mixed with the start location of next gene. 687-2534, 2518-2724, 2721-4313, etc. And genes predicted are in different reading frames. Try to use start codon + 3n+ end codon, and the next start codon is not +3 but any.

7.2.6 Use AT rich as melting point, and CG as looking location. For example, for any given sequence, first search for the melting point by using sliding window algorithm find rich AT region; and then combine the result with CpG island profile in the same sequence to decide if the region is most likely to be the promoter region before TSS.

The D2K Algorithm

Software and algorithms used
I use EGF (E. coli Gene Finder, Perl program) as detection tool in this step. The purpose of this research is to explore the use of TATA-less regions for promoter prediction and based on existing algorithms and methods to find an algorithm which performs as well as or better than the existing approaches. As a result of my research and testing of existing algorithms and methods, I discovered an algorithm D2K (double k-mean with k=2) which performs better than most of online promoter predictors.

D2K Algorithm depends on the findings of TATA-less regions in Theory Testing section, and implemented in clustering step in this section. Improved promoter prediction
will use the characteristic from both TATA-rich regions and TATA-less regions to increase true positive prediction result. During data mining step, k-mean with k=2 will be used since in this case there are only two classes during data classification. K-mean is a well known clustering algorithm, it partitioning all observation into r clusters where each observation belongs to the nearest mean. D2K indicates that k-mean will be use twice as explained below.

1. First clustering will partition the given sequences into 2 clusters, based on the characteristic of TATA box. Sequences with clear TATA box information are grouped as resolved sequence. It will be discussed in detail in section 3 below.
2. Second clustering process the reminder unresolved sequence in order to detect more promoter regions that do not contain TATA box. By combine both results, EGF use k-mean one more time to get the probabilities of six reading frames. The probability of the highest reading frame will be recorded as detected promoter region. It will be discussed in detail in section 5 below.

**Data set used in algorithm step**

For Eukaryotes, since promoter contains core promoter (TATA -35, TFIIB—upstream of TATA, Inr, and downstream promoter element—DPE) and upstream promoter element [23], all four elements will be considered. For TATA less promoters, GC box or DPE will appear. Combine both CG content and TATA box information to locate the promoter region in Eukaryotes is the next step of this project.

Based on the clustering algorithm introduced in chapter 16 [2], a sub solution—E. coli Gene Finder (EGF) of promoter prediction is finished on E. coli sequences. For E. coli uses E. coli Gene Finder from cs123b, where -10 (TATAAT) and -35(TTGAC) can be easily found. For any short E. coli sequence from NCBI, the program will predict the location of possible genes.

**Steps of detecting promoter region**

1. Data preparation. One method that can make the DNA data independent in the prepare data level is PCA (principal component analysis). There are dependencies in DNA analysis, such as the properties of some data may not be truly independent. Another example is some genes are co-expressed. Samples of such data are against the principle of data mining, in which each pieces of data must be independent [17]. One way to minimize the dependency effect is to use PCA—principal component analysis. PCA will transform those data into components, which are independent of each other. New variables will become linear combination of its raw data [18]. PCA usually reduces the raw data principle to two or three components, which contain the most of the variations and ignore others, so those components can be used to classify sample experiments. For example, M sequences with N genes in each will create a matrix $X = N \times M$. After passing through PCA, the formula will be changed to $X = U \epsilon V^T$.

   Where $U$ is the expression level of every gene, $\epsilon$ is the $\text{A}\text{t}h$ eigengene that is
expressed in the Ath eigensample, and $V^T$ is the expression level in each sample. Then the data can be plot for each gene/protein pair. The PCA step used in this project will test only one sequence on six frames each run, so $M = 6$ each time a new sequence entered in the program. N is the number of genes that will be predicted in each frame, and $N$ varies in each frame. The matrix will become $X = 6N$. A group of results will be chosen out of six in the clustering step.

2. Distance definitions. There are three distances calculated in our program: the length of gene, the distance between TATA box and TSS, and the distance between -35 element and TSS. In this step we do not use any of Euclidean, Pearson, or Mahalanobis; instead we use the direct distance by finding the difference between two locations.

3. Clustering. Since there are only TATA rich and TATA less two classes in the data set, I am using k-means to classify our data with $K=2$. We partition the data set into two clusters with the number of TATA boxes as classifier. K is the number of clusters and it is fixed when use k-means clustering method. The centroids of clusters are random assigned and then relocated during each cycle, and then finalized when the centroids stop change. To make k-means more accurate, it must be run several times. A similar way to cluster TATA rich and TATA less data will use SVM (supervised clustering with support vector machines). SVM can be used to classify data in one of two classes [17]. After SVM has all the training data, the unknown data will be classified into one group among the training data. Therefore, that unknown data will have the characteristic of the group to which it was assigned. In our program we treat k-mean and SVM in the same way since there are only two final classes.

4. Significance of differential expression. We use this step to evaluate our test results. I compare my test results with several online gene predictors, and provide statistic comparison between the findings. Statistical testing measures to measure true positive (TP) [18], and false discovery rate (FDR) [18]. It will be explained in detail in the discussion section.

5. Improvement after getting first result—the second K-mean clustering process

5.1 The original design can only get less than 80% of gene compare to real gene in NCBI data base. In order to detect more genes from given sequence, the cluster algorithm is refreshed with a second K-mean clustering process to classify the remaining TATA-less group.

5.2 The original cluster decision was made by using 175 TATA-rich plan sequences, and 135 TATA-less plan sequences as training data.

5.3 Both k-mean and DBSCAN [17] are used to predict more location of gene during this step, since only testing can tell which algorithm will give a better result. However, they both do not predict more gene in this step since the number of object is too small for cluster.

5.4 There are six reading frames need to be tested for each given sequence. Using k-mean will get the probability of each reading frame. The reading frame with the highest probability is recorded as final predict result.
Implementation

Steps to research non-TATA:
1. Get E. coli sequences from NCBI
2. Design: Structuring the data requires translating data to its RNA form. This gene predictor follows the tradition gene finding process with additional clustering and statistical methods from chapter 16 [1]. It first gets the RNA forms of a given data, finds all orfs from six reading frames by using PCA, then EGF decides which orfs are real genes, depending on some of the consensus sequence characters. The last step is to choose one frame out of six frames by using the cluster algorithm to pick one with the highest probability. To show how accurate the result is, we compare the detected genes with NCBI, EasyGene, and GenScan in two categories: true positive and false discovery rate. Result and discussion depends on the genes find by all four packages.
3. Bioinformatics Analysis: To choose the best frame out of six is a statistic process based on the characteristic of a real gene. For example, one way to say the finding is a real gene is to find a TATA box in its promoter region. The sum of the number of resulting gene with TATA box before TSS will be calculated to decide the probability of this frame is the best among others.

Produce for testing D2K
1. Construct the table of findings with the probability on the same data promoter prediction data.
2. Choose several findings with the highest value to build a profile.
3. Put the findings in step 2 in the program.
4. Test the same data in EGF and some online popular tools, such as EasyGene, Genscan, etc.
5. Compare the results with discussion.

Integrating D2K into a web application
1. Choose the right platform
   As recommended by Dr. Tseng, this project will use LAMP (Linux, Apache, MySQL, PHP) platform to perform the result of detection. In addition, the Zend framework will be used to minimize the amount of code need to be implemented. I will process data using EGF and excel then extract the data using PHP to get real data from EGF, and partial data from excel.
2. Design UI: user can select several available accession numbers from left menu to see what has been predicted, and also have the chance to see the prediction from other online prediction, such as Virtual footprint, EasyGen, etc. It will show each gene's start and stop location in NCBI, Glimmer and other above online tool. It also displays the start and stop gene location that predicted by using EGF. User can easily compare the predicted results. TP and FDR will show the standing of EGF among other promoter predictors.
3. Coding: PHP, Perl, HTML, XHTML, JavaScript, JQuery, YahooSiteBuilder, etc.
3.1 Combine LAMP with YahooSiteBuilder

The original design is to try to get result from other online web gene finder and put the result on one web page for any given E. coli sequence. However, I have spent some time on each of the online gene predictor; I recognized that it is hard to accomplish this goal (within one semester). For example, Virtual footprint is PHP based software suite, it does not support automatic promoter analysis. The user has to go through several steps to get the result of a given sequence. For EasyGen, the predicted result may come from email instead of instance result. Therefore, it is not effective to implement an automatic web result compare tool. To fully support the main origin of this paper, it is redesigned as a web representation tool, to support the main goal of this research.

3.2 YahooSiteBuilder with Excel

Since most result were processed by using excel, displaying data from excel to a web browser made it more user friendly. To achieve this display, I insert an iframe inside YahooSiteBuilder page. This iframe can upload an excel and display it on a web browser.

4. Testing: using accession number from section “Result and Conclusion”

5. Improving:

5.1 Adding more data from NCBI to test
5.2 Choose a nice layout for each sequence
5.3 Make an index page with
   5.3.1 Links to each sequence result page
   5.3.2 Short summary of this paper
   5.3.3 Purpose of this web implementation
   5.3.4 How to use, etc.

Result

Twenty different E. coli Data are tested, and genes locations are listed. Start and Stop is the result of E. coli Gene Finder. The last row is the number of genes found in each. Please see appendix 4 for detail running result of each promoter predictor.

Result

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>4363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4383</td>
<td>5129</td>
<td>4383</td>
<td>5129</td>
<td>86</td>
<td>4363</td>
<td>89</td>
<td>6890</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1: Result of gene search for 300901746

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>278</td>
<td>1372</td>
<td>278</td>
<td>1372</td>
<td>278</td>
<td>1372</td>
<td>431</td>
<td>1372</td>
<td>309</td>
<td>5512</td>
<td>66</td>
<td>79</td>
</tr>
<tr>
<td>1396</td>
<td>1785</td>
<td>1396</td>
<td>1785</td>
<td></td>
<td></td>
<td>1396</td>
<td>1785</td>
<td></td>
<td></td>
<td>975</td>
<td>988</td>
</tr>
<tr>
<td>1904</td>
<td>4777</td>
<td>1904</td>
<td>4777</td>
<td>1904</td>
<td>4777</td>
<td>1988</td>
<td>4777</td>
<td></td>
<td></td>
<td>4202</td>
<td>4215</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Result of gene search for 403342

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>138</td>
<td>1580</td>
<td>138</td>
<td>1580</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>254</td>
<td>2669</td>
<td>1197</td>
<td>1210</td>
</tr>
<tr>
<td>1592</td>
<td>2515</td>
<td>1592</td>
<td>2515</td>
<td>1592</td>
<td>2515</td>
<td></td>
<td></td>
<td>2672</td>
<td>4851</td>
<td>1600</td>
<td>1613</td>
</tr>
<tr>
<td>2564</td>
<td>3646</td>
<td>2624</td>
<td>3646</td>
<td>2564</td>
<td>3646</td>
<td></td>
<td></td>
<td>3730</td>
<td>4644</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4823</td>
<td>5935</td>
<td>4751</td>
<td>5935</td>
<td>4751</td>
<td>5935</td>
<td></td>
<td></td>
<td>4868</td>
<td>6091</td>
<td>5390</td>
<td>5403</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Result of gene search for 325965637

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>138</td>
<td>1580</td>
<td>138</td>
<td>1580</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>254</td>
<td>2669</td>
<td>1197</td>
<td>1210</td>
</tr>
<tr>
<td>1592</td>
<td>2515</td>
<td>1592</td>
<td>2515</td>
<td>1592</td>
<td>2515</td>
<td></td>
<td></td>
<td>2672</td>
<td>4851</td>
<td>1600</td>
<td>1613</td>
</tr>
<tr>
<td>2564</td>
<td>3646</td>
<td>2624</td>
<td>3646</td>
<td>2564</td>
<td>3646</td>
<td></td>
<td></td>
<td>3730</td>
<td>4644</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4823</td>
<td>5935</td>
<td>4751</td>
<td>5935</td>
<td>4751</td>
<td>5935</td>
<td></td>
<td></td>
<td>4868</td>
<td>6091</td>
<td>5390</td>
<td>5403</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Result of gene search for 346421495

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>546</td>
<td>88</td>
<td>210</td>
<td>81</td>
<td>323</td>
<td></td>
<td></td>
<td>69</td>
<td>516</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Result of gene search for 346421495

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1026</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1026</td>
<td>232</td>
<td>1026</td>
<td>66</td>
<td>1079</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5: Result of gene search for 260765442

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>519</td>
<td>128</td>
<td>498</td>
<td>174</td>
<td>392</td>
<td>50</td>
<td>1557</td>
<td>272</td>
<td>285</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6: Result of gene search for 167509193

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>253</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>214</td>
<td>92</td>
<td>17</td>
<td>26</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7: Result of gene search for 354515243

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>225</td>
<td>796</td>
<td>141</td>
<td>788</td>
<td>225</td>
<td>788</td>
<td>332</td>
<td>855</td>
<td>679</td>
<td>692</td>
<td>692</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8: Result of gene search for 354515242

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>104</td>
<td>808</td>
<td>35</td>
<td>277</td>
<td></td>
<td>204</td>
<td>874</td>
<td>257</td>
<td>266</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 9: Result of gene search for 354515240

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>381</td>
<td>390</td>
<td>418</td>
<td>427</td>
<td>251</td>
<td>260</td>
</tr>
<tr>
<td>589</td>
<td>818</td>
<td>589</td>
<td>34</td>
<td>322</td>
<td>573</td>
<td>145467</td>
<td>145467</td>
<td>397</td>
<td>406</td>
<td>451</td>
<td>460</td>
</tr>
</tbody>
</table>

Table 10: Result of gene search for 354515237

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>458</td>
<td>67</td>
<td>264</td>
<td>9</td>
<td>458</td>
<td>705</td>
<td>920</td>
<td>251</td>
<td>260</td>
<td>397</td>
<td>406</td>
</tr>
<tr>
<td>589</td>
<td>818</td>
<td>589</td>
<td>34</td>
<td>322</td>
<td>573</td>
<td>145467</td>
<td>145467</td>
<td>397</td>
<td>406</td>
<td>451</td>
<td>460</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>389</td>
<td>458</td>
<td>489</td>
<td>528</td>
<td>489</td>
<td>528</td>
</tr>
</tbody>
</table>

Table 11: Result of gene search for 145467

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>1688</td>
<td>489</td>
<td>1688</td>
<td>450</td>
<td>1688</td>
<td>9</td>
<td>386</td>
<td>123</td>
<td>395</td>
<td>267</td>
<td>280</td>
</tr>
<tr>
<td>2</td>
<td>191</td>
<td>2024</td>
<td>1682</td>
<td>1695</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 12: Result of gene search for 342315677

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>674</td>
<td>322</td>
<td>621</td>
<td>145</td>
<td>154</td>
<td>224</td>
<td>233</td>
<td>351</td>
<td>360</td>
</tr>
</tbody>
</table>

Table 13: Result of gene search for 341941295

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>674</td>
<td>322</td>
<td>621</td>
<td>145</td>
<td>154</td>
<td>224</td>
<td>233</td>
<td>351</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1299</td>
<td>1</td>
<td>1299</td>
<td>132</td>
<td>1361</td>
<td>572</td>
<td>585</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 13**: Result of gene search for 341941295

### 14. Accession number 41745

<table>
<thead>
<tr>
<th></th>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>227</td>
<td>1621</td>
<td>227</td>
<td>1621</td>
<td>227</td>
<td>1621</td>
<td>280</td>
<td>1645</td>
<td>900</td>
<td>909</td>
<td>180</td>
<td>200</td>
<td>1999</td>
</tr>
</tbody>
</table>

**Table 14**: Result of gene search for 41745

### 15. Accession number 41727

<table>
<thead>
<tr>
<th></th>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1971</td>
<td>2459</td>
<td>1971</td>
<td>2459</td>
<td>1971</td>
<td>2459</td>
<td>2064</td>
<td>2459</td>
<td>1998</td>
<td>2037</td>
<td>1908</td>
<td>1508</td>
<td>1571</td>
</tr>
</tbody>
</table>

**Table 15**: Result of gene search for 41727

### 16. Accession number 41592

<table>
<thead>
<tr>
<th></th>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>1505</td>
<td>300</td>
<td>1505</td>
<td>300</td>
<td>1505</td>
<td>432</td>
<td>1505</td>
<td>319</td>
<td>1524</td>
<td>1575</td>
<td>1524</td>
<td>1588</td>
</tr>
</tbody>
</table>

**Table 16**: Result of gene search for 41592

### 17. Accession number 41580

<table>
<thead>
<tr>
<th></th>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>1107</td>
<td>31</td>
<td>1107</td>
<td>31</td>
<td>1107</td>
<td>1181</td>
<td>1107</td>
<td>31</td>
<td>959</td>
<td>1181</td>
<td>959</td>
<td>972</td>
</tr>
</tbody>
</table>

**Table 17**: Result of gene search for 41580

### 18. Accession number 414745

<table>
<thead>
<tr>
<th></th>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
</table>

26
Table 18: Result of gene search for 414745

<table>
<thead>
<tr>
<th>Accession number</th>
<th>NCBI start</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>474</td>
<td>31</td>
<td>474</td>
<td>31</td>
<td>474</td>
<td>178</td>
<td>187</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>1884</td>
<td>502</td>
<td>1884</td>
<td>742</td>
<td>1884</td>
<td>1354</td>
<td>1363</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 19: Result of gene search for 312761

<table>
<thead>
<tr>
<th>Accession number</th>
<th>NCBI start</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>300901746</td>
<td>1</td>
<td>58.77778</td>
<td></td>
<td></td>
<td></td>
<td>0.8</td>
<td>888</td>
<td>0.44444</td>
<td>1397.75</td>
<td>0.33333</td>
<td>2785.66</td>
</tr>
<tr>
<td>325965637</td>
<td>1</td>
<td>66.2</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>666</td>
<td>0.33333</td>
<td>4171</td>
<td>1</td>
<td>1580.3</td>
</tr>
</tbody>
</table>

Table 20: Result of gene search for 297393

<table>
<thead>
<tr>
<th>Accession number</th>
<th>NCBI start</th>
<th>Glimmer Start</th>
<th>Glimmer Stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyGene start</th>
<th>EasyGene stop</th>
<th>Genscan start</th>
<th>Genscan stop</th>
<th>Virtual Footprint Start</th>
<th>Virtual Footprint Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Discussion:
TP is true positive which means the gene is predicted when there is a gene, the bigger value the better result. And the formula is

\[
TP = \frac{\text{number of genes predicted}}{\text{total genes in NCBI of this sequence}}
\]

FDR is false discovery rate which provides the rate of false location in prediction, the smaller the value the better. And the formula is

\[
FDR = \frac{\text{total location shift predicted}}{\text{number of genes predicted}}
\]
From the above table we can see that EGF predicts most of the real genes with a lower false discovery rate among the other predictors.

Charts of above data

Table 21: Result for TP and FDR comparison

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>346421495</td>
<td>1</td>
<td>136</td>
<td>1</td>
<td>110</td>
<td>0</td>
<td>1103</td>
<td>1</td>
</tr>
<tr>
<td>260765442</td>
<td>0</td>
<td>1027</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>231</td>
<td>1</td>
</tr>
<tr>
<td>167509193</td>
<td>1</td>
<td>106</td>
<td>1</td>
<td>46</td>
<td>0</td>
<td>520</td>
<td>1</td>
</tr>
<tr>
<td>354515243</td>
<td>0</td>
<td>254</td>
<td>0</td>
<td>254</td>
<td>1</td>
<td>254</td>
<td>1</td>
</tr>
<tr>
<td>354515242</td>
<td>0</td>
<td>1021</td>
<td>1</td>
<td>92</td>
<td>0</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>354515240</td>
<td>0</td>
<td>912</td>
<td>2</td>
<td>678</td>
<td>0</td>
<td>912</td>
<td>1</td>
</tr>
<tr>
<td>354515237</td>
<td>0.5</td>
<td>1243</td>
<td>1</td>
<td>640</td>
<td>0.5</td>
<td>1399</td>
<td>0.5</td>
</tr>
<tr>
<td>145467</td>
<td>1</td>
<td>909</td>
<td>2</td>
<td>126</td>
<td>0</td>
<td>1268</td>
<td>3</td>
</tr>
<tr>
<td>342315677</td>
<td>0</td>
<td>675</td>
<td>1</td>
<td>268</td>
<td>0</td>
<td>675</td>
<td>1</td>
</tr>
<tr>
<td>341941295</td>
<td>0</td>
<td>1300</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1300</td>
<td>1</td>
</tr>
<tr>
<td>41745</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1848</td>
<td>1</td>
</tr>
<tr>
<td>41727</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>93</td>
<td>1</td>
</tr>
<tr>
<td>41592</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>132</td>
<td>1</td>
</tr>
<tr>
<td>41580</td>
<td>0</td>
<td>1138</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>175</td>
<td>1</td>
</tr>
<tr>
<td>414745</td>
<td>1</td>
<td>5652</td>
<td>2</td>
<td>245</td>
<td>1</td>
<td>5652</td>
<td>2</td>
</tr>
<tr>
<td>312761</td>
<td>0</td>
<td>2891</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>297393</td>
<td>1</td>
<td>1406</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 1: TP result. X axis is the 20 different genes, and y axis is the number of genes predicted/ gene in NCBI database. NCBI results are all 1s, since they are the number of genes in the database. For other predictors, some predicted more than expected, and some are less.
Figure 2: TP result average. TP Average = TP result /number of experiments (20). To see the performance of each predictor, it is better to use an averaged data for comparison. Series 1 is the default genes existed in NCBI database, Series 2 is the percentage of genes found by Glimmer, Series 3 is the percentage of genes found by using E. coli Gene Finder, Series 4 is the percentage of genes found by EasyGene, and Series 5 is the percentage of genes found by GenScan, Series 6 is the percentage of genes found by Vertual Foot Print. In this case, the closer the better; therefore EGF and GenScan are more closer to NCBI database.

Figure 3: FDR Result. FDR = sum of number of position shift / number of gene predicted. X axis is the 20 genes used in this project; Y axis is the number of false gene locations. The data represent the difference between the predicted gene location with the gene location in the database, therefore the smaller the better.
Figure 4. FDR average. FDR Average = sum of FDR/ total number of predicted gene (20). Series 1 is the default false gene location in NCBI which is 0, Series 2 is the miss calculated gene location by Glimmer, Series 3 is the false predicted gene location by EGF, Series 4 is the false gene location that was predicted by EasyGene, Series 5 shows the wrong gene locations as predicted by Genscan, and Series 6 is the false prediction by Virtual Foot Print. EGF predicts the less FDR among other predictors.

Final results can be improved by checking the start codon condition. For example, if there is a hairpin structure before AUG; then this AUG will be most likely the start of a gene. Or, if there is another AUG just few codon after a start codon; then the start codon will be passed and this AUG will become the start of a gene [24]. This step can be done to either locate promoter phase or improve the result phase.

Related topics:
Cancer: the cancer is caused not only by a single mutation, but several mutations on the chromosome. It can be explained by the exponential growth of some cancer cause death growth with age. One example is the colon cancer death raise in “One Renegade Cell” page 47 [27]. If colon cancer is caused by 3 mutations, then the formula will be $2^3$. If it takes about 2 years for one mutation to happen, then after 16 ($2^3\times2$) years, someone who has all the mutations will develop colon cancer. If it caused by 4 mutations, then the time needed are 32. Off cause, mutations that happened not related to colon cancer will result in no colon cancer even when more than 4 mutations are detected in one sequence. To prevent colon cancer or any other cancer, couples of analysis need to be done. First, all mutations required by colon cancer need to be defined. Second, use examples (someone who has colon cancer family history, does not have colon cancer yet), and find out the difference. Third, prevent the last (one or more) mutation from happening by providing some treatment or medicine.
Conclusion

The prediction made by this program works only for some E. coli sequences as we can see from the data from NCBI. Compare each predicted gene with NCBI, EGF predicts more real gene than other online predictor, and it also gives less wrong prediction than other predictor. Even though the results look better than some of the online packages prediction, it still has the limitation on predictions. For example, it can’t predict short gene (length less than 60 nucleotides) and overlapping gene (gene inside gene). Those genes do exist in NCBI database which include the E. coli gene I used in this paper. It is just the start point of this project by using the basic characteristic that found in E. coli; further study on other related organisms needed to broader the search power of the system. After all the ideas have being finalized in section “steps to find characteristics 7”, then the prediction will be enhanced by adding more idea in the original program.

Reference

10. Thomas Abeel, Yvan Saëys, Eric Bonnet, Pierre Rouzé, and Yves Van de Peer:
Generic eukaryotic core promoter prediction using structural features of DNA. Genome Res. 2008. 18: 310-323. http://genome.cshlp.org/content/18/2/310.full
14. Xiaowo Wang, Zhenyu Xuan, Xiaoyue Zhao, Yanda Li1 and Michael Q. Zhang: High-resolution human core-promoter prediction with CoreBoost_HM. Genome Res. 2009. 19: 266-275. http://genome.cshlp.org/content/19/2/266.full
16. RNA codon chart http://www.ccs.k12.in.us/chsteachers/BYost/Biology%20Notes/translationnoteguide.htm
17. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Introduction to Data Mining, 2006 by Pearson Education, Inc.
27. Robert A. Weinberg, One Renegade Cell
29. Marc Wandschneider, Core Web Application Development With PHP and MySQL, 2006 by Pearson
34. Virtual Footprint:  http://www.prodoric.de/vfp/vfp_promoter.php
Appendix 1—Training Data

1. 175 TATA rich sequences
   http://mendel.cs.rhul.ac.uk/pprom/PLPR_TATA.seq

2. 130 TATA less sequences
   http://mendel.cs.rhul.ac.uk/pprom/PLPR_TATA-less.seq
Appendix 2—List of NCBI Accession Number of Testing Data

1. ecoli41580
2. ecoli41592
3. ecoli41727
4. ecoli41745
5. ecoli145467
6. ecoli297393
7. ecoli312761
8. ecoli403342
9. ecoli414745
10. ecoli167509193
11. ecoli260765442
12. ecoli300901746
13. ecoli325965637
14. ecoli341941295
15. ecoli342315677
16. ecoli346421495
17. ecoli354515237
18. ecoli354515240
19. ecoli354515242
20. ecoli354515243
Appendix 3—D2K Algorithm Coding in EGF

# This program take any E.coli sequence from NCBI and find genes from it

# if(!open(infile, 'ecoli356875267.txt')){
  if(!open(infile, 'ecoli297393.txt')){
    print "error opening input file\n";
    exit;
  }
  if(!open(outfile, '>out.txt')){
    print "error opening output file\n";
    exit;
  }
  if(!open(outfile1, '>out1.txt')){
    print "error opening output file\n";
    exit;
  }
}$data = <infile>; #ignore FASTA comment
while ($data = <infile>){
  chomp $data;
  $seq = $seq . $data;
}

# $seq is the nontemplate strand from the 5' end
# first three reading frames come from $seq

# other three reading frames come from the reverse complement
$complement = $seq;
$complement =~ tr/ACGTacgt/TGCAtgca/; # complement of strand
$reversecomplement = reverse($complement); # reverse of compement
$reversecomplement =~ s/T/U/g; # convert to RNA

$seq =~ s/T/U/g; # convert to RNA

# find ORF in original sequence
$foundorf = 0;

# find ORF in reversecomplement
$stop = 0;

print "seq strand\n";
$seq_result = findORF($seq);
print "\n\n reversecomplement strand\n";
$reversecomplement_result = findORF($reversecomplement);

# print "reverse complement result: $reversecomplement_result";
# print "seq result: $seq_result";

if (($reversecomplement_result < 0) && ($seq_result < 0)){
    print outfile "ORF not found\n";
}
# find orf
sub findORF{
    my($seq) = @_;
    $found = 0;
    # set the starting position of the reading frame
    for($frame = 0; $frame < 3; $frame++){#one promoter per sequence
        $start = $frame;
        $tataCount = 0;
        #$missingStart = substr($seq, 4383, 3);
        #$missingStop = substr($seq, 5129-3, 3);
        #print "missing gene: $missingStart -- $missingStop";
        print "\n frame: ";
        print $frame;
        print "\n";
        $findPromoter = 0;
        while ($start < length($seq)){
            # find start codon in reading frame
            $start = findStartStop($seq, $start,0);
            if ($start == -1) {last;}
            #print "start: $start ";
            #print "\n";
            # look for stop codon at least 60 codons out or 180nt
            if (($start != -1) && ($start+180<=length($seq)-3)){
                $stop = findStartStop($seq, $start,1);
                # print " stop: $stop";
                #print "\n";
                if ($stop >= $start+180){ # length of the gene > 60 codon
                    #$totalLength = length($seq);
                    $realStart = ($start +1);
                    $realStop = ($stop +3);
                    #$startString = substr ($seq, $realStart, 3);
```perl
#$stopString = substr ($seq, $realStop-3, 3);
print "start-stop $realStart--$realStop \n";  # $startString---$stopString \n";
print outfile1 "$frame start-stop $realStart---$realStop \n";
$tataLocation = findTata($seq, $realStart);
if ($tataLocation != -1) {$tataCount = $tataCount +1;}
    $found = 1;
    if ($foundorf == 1){
        print outfile "\n ---Next--\n";
    }
    $foundorf = 1;
# printed assuming first position is 1
print outfile "ORF found in reading frame ",$frame+1, " Start Loc: ",
    $start+1, " Stop Loc: ", $stop+1, " and Shine-Dalgarno is
found or not (-1): ", findShine($frame, $start, $seq), ". \n";
    print outfile substr($seq, $start, $stop+3-$start);
    #print promoter info
    if (findPromoters($seq, $start)==1){
        print outfile "\n ORF supported by promoters\n";
        $findPromoter = 1;
    }
else{
    if ($findPromoter = 0){
        print outfile "\n ORF not supported by promoters\n";}
else {
    print outfile "\n ORF is in the operon\n";}
}
if ($stop != -1) {$start = $stop;}
}
$start = $start + 3; # use $start + 3 in E. coli, and $start -15 in E.
albertii
}
$tataCount = $tataCount/6;
print "\n Probability of using frame $frame is $tataCount.\n";
print outfile1 "\n Probability of using frame $frame is $tataCount.\n";
}
if ($found == 0){return(-1);}
elsif ($found == 1){return(1);}
```

sub findStartStop { #combine find start and stop of orf
    my($seq, $start, $choice) = @_; #combine find start and stop of orf
    for ($i=$start; $i<=length($seq)-3; $i+=3){
        if ($choice){  #start
            if ((substr($seq, $i, 3) eq "UAA")
                || (substr($seq, $i, 3) eq "UAG")
                || (substr($seq, $i, 3) eq "UGA")){
                return($i);
            }
        } else {  #stop
            if ((substr($seq, $i, 3) eq "AUG")
                || (substr($seq, $i, 3) eq "GUG")
                || (substr($seq, $i, 3) eq "UGA")
                || (substr($seq, $i, 3) eq "CUG")
                || (substr($seq, $i, 3) eq "UUG")){
                return($i);
            }
        }
    }
    return(-1);
} #start
#stop

sub findShine {  #find shine-dalgarno sequence
    my($frame, $start, $seq) = @_; #find shine-dalgarno sequence
    #print "frame ",$frame+1," \n";
    $position = ($start -5)*3 + 2;
    $string = substr($seq, $position, 7);
    #print "\n String is >>> $string <<< \n";
    #print "\n String is >>> $seq <<< \n";
    return index($string, "AGGAGG");
}

sub findPromoters { #find promoter
    my($seq, $orfstart) = @_; #find promoter
    # modified to work with exercise 1 program
    $element35 = "UUGACA";
    $element10 = "UAUAAU";
    # initialize the search position for the -35 element
    $search35 = 0;
}
# check distance from translational start site
while ($search35 < $orfstart - 85){
    if (index(substr($seq, $search35, length($element35) + 10), $element35) > -1){
        # element -35 found, initialize search position for -10 element
        $elementdist = 15;
        $search10 = $search35 + length($element35) + $elementdist;
        while ($search10 < $orfstart - 60 && $elementdist <= 19){
            if (index(substr($seq, $search10, length($element10) + 10), $element10) > -1){
                # valid -10 element found, return success
                return (1);
            }
            # continue searching for -10 element
            $search10++;
            $elementdist++;
        }
        # -35 element not found, continue searching
        $search35 = $search35 + length($element35);
    } else{
        # -35 element not found, continue searching
        $search35++;
    }
}
# valid promoters not found - return 0
return(0);

sub findTata{ #find TATA box
    my($seq, $start) = @_;
    $tataRegion = $start - 20;
    $string = substr($seq, $tataRegion, 20);
    #print " tata-->$string $tataRegion ";
    #print index($string, "UAUU"); #find TATA box location
    return index($string, "UAUU");
}

close (infile);
close (outfile);
close (outfile1);
Appendix 4—Result Comparison

1. ecoli41580

Submit new Data

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
</table>

```
frame: 0
start-stop 31--1107
Probability of using frame 0 is 0.
frame: 1
Probability of using frame 1 is 0.
frame: 2
Probability of using frame 2 is 0.
```

Gn. Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

1.01 Sing + 132 1181 1050 2 0 74 35 1128 0.230 103.26

Suboptimal exons with probability > 1.000

---

<table>
<thead>
<tr>
<th>gene</th>
<th>start</th>
<th>end</th>
<th>strands</th>
<th>score</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>282</td>
<td>291</td>
<td>+</td>
<td>0.30</td>
<td>DISTRICT</td>
</tr>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>769</td>
<td>788</td>
<td>+</td>
<td>0.23</td>
<td>DISTRICT</td>
</tr>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>1110</td>
<td>1129</td>
<td>+</td>
<td>0.09</td>
<td>DISTRICT</td>
</tr>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>979</td>
<td>972</td>
<td>+</td>
<td>0.93</td>
<td>DISTRICT</td>
</tr>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>209</td>
<td>273</td>
<td>-</td>
<td>0.45</td>
<td>DISTRICT</td>
</tr>
<tr>
<td>ArgI (Escherichia coli [strain K12])</td>
<td>117</td>
<td>133</td>
<td>-</td>
<td>0.35</td>
<td>DISTRICT</td>
</tr>
</tbody>
</table>
2. ecoli41592

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;gi</td>
<td>41592</td>
<td>emb</td>
<td>X17499.1</td>
<td>E. coli gltS gene</td>
</tr>
<tr>
<td>orf00001</td>
<td>308</td>
<td>123</td>
<td>-3</td>
<td>0.65</td>
</tr>
<tr>
<td>orf00002</td>
<td>300</td>
<td>1505</td>
<td>+3</td>
<td>3.46</td>
</tr>
<tr>
<td>orf00005</td>
<td>44</td>
<td>1674</td>
<td>-3</td>
<td>2.21</td>
</tr>
</tbody>
</table>

---

Command Prompt

Frame: 1
start-stop 1187--1384
Probability of using frame 1 is 0.

Frame: 2
start-stop 300--1505
Probability of using frame 2 is 0.

reverse complement strand

---

# gff-version 2
# source-version easygene-1.2b
# date 2011-12-06
# type DNA
# model: AP02 Aeropyrum pernix
# transcripts model feature start end score +/- ? startsc ode

qi_41592_emb_X17499.1_AP02 CDS 432 1505 1.25902e-09 + 0 #OTG 36.9953

---

Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

----- --- - ------ ----- --- - -- --- --- ----- ----- -----

1.01 Sngl + 319 1524 1206 0 0 41 41 1188 0.450 103.76

Suboptimal exons with probability > 1.000

---

<table>
<thead>
<tr>
<th>ORF</th>
<th>Start Position</th>
<th>End Position</th>
<th>Strand</th>
<th>Score</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>174</td>
<td>193</td>
<td>+</td>
<td>7.33</td>
</tr>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>129</td>
<td>130</td>
<td>-</td>
<td>8.56</td>
</tr>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>172</td>
<td>191</td>
<td>+</td>
<td>6.51</td>
</tr>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>1975</td>
<td>1986</td>
<td>-</td>
<td>8.43</td>
</tr>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>1975</td>
<td>1986</td>
<td>+</td>
<td>8.43</td>
</tr>
<tr>
<td>AP2X</td>
<td>E. coli cell (strain K12)</td>
<td>1982</td>
<td>1996</td>
<td>-</td>
<td>8.08</td>
</tr>
</tbody>
</table>
3. ecoli41727

GLIMMER (ver. 3.02, iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>orf00001</td>
<td>536</td>
<td>513</td>
<td>-1</td>
<td>7.08</td>
</tr>
<tr>
<td>orf00002</td>
<td>1804</td>
<td>620</td>
<td>-2</td>
<td>3.37</td>
</tr>
<tr>
<td>orf00003</td>
<td>1971</td>
<td>2359</td>
<td>+3</td>
<td>6.04</td>
</tr>
<tr>
<td>orf00005</td>
<td>2966</td>
<td>2364</td>
<td>-3</td>
<td>3.07</td>
</tr>
<tr>
<td>orf00006</td>
<td>3252</td>
<td>2960</td>
<td>-1</td>
<td>5.11</td>
</tr>
<tr>
<td>orf00008</td>
<td>3572</td>
<td>3382</td>
<td>-3</td>
<td>11.32</td>
</tr>
<tr>
<td>orf00009</td>
<td>156</td>
<td>3550</td>
<td>-1</td>
<td>3.84</td>
</tr>
</tbody>
</table>

Probability of using frame 1 is 0.

frame: 2

Probability of using frame 2 is 0.

Suboptimal exons with probability > 1.000
4. *coli41745*

GLYMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>orf00001</td>
<td>2351</td>
<td>230</td>
<td>+3</td>
<td>4.29</td>
</tr>
<tr>
<td>orf00002</td>
<td>227</td>
<td>1621</td>
<td>+2</td>
<td>3.50</td>
</tr>
<tr>
<td>orf00004</td>
<td>1814</td>
<td>1753</td>
<td>-1</td>
<td>2.58</td>
</tr>
<tr>
<td>orf00005</td>
<td>1983</td>
<td>2139</td>
<td>+3</td>
<td>1.26</td>
</tr>
</tbody>
</table>

>gi|61745|emb|V00288.1| B. coli specificity gene of EcoK restriction enzyme (hesS)

**Command Prompt**

```plaintext
seq strand
frame: 0
start-step 1273--1488
start-step 1792--2022
Probability of using frame 0 is 0.16666666666666667.
frame: 1
start-step 227--1621
Probability of using frame 1 is 0.
```

#sff-plugin 2
#SOURCE-VERSION asygene-1.2b
#state 2011-12-06
#Type DNA
# model: ApO2 Aeropyrum pernix
# seqname model feature start end score +/- ? starte  odds

**Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRs P.... Tscr..**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>Init+</td>
<td>280</td>
<td>1645</td>
<td>1366</td>
<td>0</td>
</tr>
<tr>
<td>1.02</td>
<td>Intr+</td>
<td>1897</td>
<td>2031</td>
<td>135</td>
<td>2</td>
</tr>
<tr>
<td>1.03</td>
<td>Intr+</td>
<td>2076</td>
<td>2222</td>
<td>147</td>
<td>1</td>
</tr>
</tbody>
</table>

Suboptimal exons with probability > 1.000

**Exnum Type S .Begin ...End .Len Fr Ph B/Ac Do/T CodRs P.... Tscr..**
5. ecoli145467

GLINHER (ver. 3.02; iterated) predictions:
offID  start  end  frame  score
------  -----  ---  ----  ----
gecoli145467|gb|M12788.1|KCCCA E.coli com gene encoding tRNA nucleotidytransferase, complete cds
torf00001  2068  386  +3   3.13
torf00006  489   688  +3   2.91
torf00008  2815  1668 -3   1.63
torf00009  2236  2033 -2   4.26

Probability of using frame 0 is 0.
frame: 1
start-stop 1685--1906
Probability of using frame 1 is 0.
frame: 2
start-stop 9--386
start-stop 450--1688
Probability of using frame 2 is 0.

Gn. Ex Type  S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..
------  ----  -----  -----  ----  ----  ----  ----  ----  ----  ----  ----
1.01 Intr +  123   395  273  1  0  40  10  454  0.293  30.85
1.02 Intr +  564  1714 1151  1  2  -9  10  1305  0.060  102.55
1.03 Term +  2024  2191  168  1  0  60  43  233  0.209  19.19

Suboptimal exons with probability > 1.000

Exnum Type  S .Begin ...End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr..
6. ecoli297393

GLIMPNER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>orf00002</td>
<td>344</td>
<td>171</td>
<td>-3</td>
<td>1.15</td>
</tr>
<tr>
<td>orf00003</td>
<td>492</td>
<td>728</td>
<td>+3</td>
<td>4.99</td>
</tr>
<tr>
<td>orf00006</td>
<td>165</td>
<td>706</td>
<td>-1</td>
<td>2.95</td>
</tr>
</tbody>
</table>

---

Gn.Ex Type S .Begin ...End .Len Pr Ph I/Ac Da/T CodRg P.... Tscr...

1.02 PlyA -  696  691  6

1.01 Sngl -  1997  759 1239 2 0 60 47 1768 0.932 166.26

Suboptimal exons with probability > 1.000
7. ecoli312761

GLIMMER (ver. 3.02, iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
</table>

```
seq strand
frame: 0
start-stop 31--474
start-stop 502--1884

Probability of using frame 0 is 0.166666666666667.
frame: 1
start-stop 41--274
start-stop 971--1219

Probability of using frame 1 is 0.
```

```
#gff-version 2
#source-version easygene-1.2b
#date 2011-12-16
#Type DNA
# model: AP02 Aeropyrum pernix
# expname model feature start end score +/ - ? startc odds
#
 gi_312761_emb_X72477.1 AP02 CDS 31 474 0.729413 + 0 #ATG -0.9931
 gi_312761_emb_X72477.1 AP02 CDS 742 1884 7.06691e-09 + 0 #ATG 15.5812
```

Gnu.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr...

<table>
<thead>
<tr>
<th>1.01 Term +</th>
<th>520</th>
<th>1911</th>
<th>1392</th>
<th>0</th>
<th>0</th>
<th>90</th>
<th>42</th>
<th>1120</th>
<th>0.454</th>
<th>98.81</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02 PlyA +</td>
<td>1953</td>
<td>1983</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PMM (species)</th>
<th>Start Position</th>
<th>End Position</th>
<th>Strand</th>
<th>Score</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack</td>
<td>Escherichia coli (strain K12)</td>
<td>2067</td>
<td>2076</td>
<td>+</td>
<td>6.91</td>
</tr>
<tr>
<td>Ack</td>
<td>Escherichia coli (strain K12)</td>
<td>1364</td>
<td>1363</td>
<td>-</td>
<td>6.55</td>
</tr>
<tr>
<td>Ack</td>
<td>Escherichia coli (strain K12)</td>
<td>176</td>
<td>197</td>
<td>+</td>
<td>6.50</td>
</tr>
<tr>
<td>ApR</td>
<td>Escherichia coli (strain K12)</td>
<td>1340</td>
<td>1353</td>
<td>-</td>
<td>0.11</td>
</tr>
<tr>
<td>ApR</td>
<td>Escherichia coli (strain K12)</td>
<td>387</td>
<td>400</td>
<td>+</td>
<td>8.90</td>
</tr>
<tr>
<td>ApR</td>
<td>Escherichia coli (strain K12)</td>
<td>1217</td>
<td>1230</td>
<td>-</td>
<td>8.42</td>
</tr>
</tbody>
</table>
8. ecoli403342

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>gi</td>
<td>354515237</td>
<td>gb</td>
<td>JP918432.1</td>
<td>Escherichia coli class I integron dihydrofolate reductase ORF0001</td>
</tr>
</tbody>
</table>

---

Frame: 0
start-stop: 1393-1785
start-stop: 4201-4476

Probability of using frame 0 is 0.

Frame: 1
start-stop: 278-1372
start-stop: 1904-4777

Probability of using frame 1 is 0.16666666666667.

Frame: 2
start-stop: 2497-2303
start-stop: 2698-2975

Probability of using frame 2 is 0.

---

**AP03 Aeropyrum pernix**

<table>
<thead>
<tr>
<th>segment</th>
<th>model</th>
<th>feature</th>
<th>start</th>
<th>end</th>
<th>score</th>
<th>+/-</th>
<th>?</th>
<th>odds</th>
</tr>
</thead>
<tbody>
<tr>
<td>gi</td>
<td>354141469</td>
<td>gb</td>
<td>2497</td>
<td>2303</td>
<td>2975</td>
<td>1</td>
<td>1</td>
<td>385</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Feature</th>
<th>Start Position</th>
<th>End Position</th>
<th>Strand</th>
<th>Score</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACP</td>
<td>307</td>
<td>493</td>
<td>-</td>
<td>0.05</td>
<td>TTTTTAAA</td>
</tr>
<tr>
<td>ACP</td>
<td>291</td>
<td>293</td>
<td>+</td>
<td>0.59</td>
<td>AATTGAA</td>
</tr>
<tr>
<td>ACP</td>
<td>401</td>
<td>492</td>
<td>-</td>
<td>0.53</td>
<td>CAGCAGC</td>
</tr>
</tbody>
</table>

---

48
9. *ecoli*414745

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

>gi|414745|emb|X71063.1| E.coli feoA and feoB genes
orf00002  821 2899  +2   3.00

Gn.Ex Type S :Begin ...End .Len Fr Ph I/Ac Do/T CodRs P.... Tscr.

1.01 Init +  605 2381 2277  1  0  64  1  111 2992  0.567  287.09
1.02  Intr + 2944 3146 203  0  2  44  40 209  0.510  10.31

Suboptimal exons with probability > 1.000
10. ecoli167509193

GLIMMER (ver. 3.02, iterated) predictions:

<table>
<thead>
<tr>
<th>OrfID</th>
<th>Start</th>
<th>End</th>
<th>Frame</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>128</td>
<td>498</td>
<td>-3</td>
<td>6.38</td>
</tr>
</tbody>
</table>

```
frame: 1
start-stop 95-316
Probability of using frame 1 is 0.
frame: 2
start-stop 174-392
Probability of using frame 2 is 0.166666666666667.
```

**Explain the output. Go back.**

**Gran Ex Type S.** Begin ... End ... Len Fr Ph I/Ac Do/T CodRg P .... Tscr...

```
1.01 Init + 50 557 508 1 1 19 -11 604 0.578 36.77
```

Suboptimal exons with probability > 1.000
11. ecoli260765442

GLIMMER (ver. 3.02; iterated) predictions:

crifID start end frame score

----------- ----- ----- -- ------

Command Prompt

G:\Users\Jerry\Desktop\li wen 2018\cs123\Term Project\perl PromoterFinder.perl

seq

strand

frame: 0
start-stop 1--1026

Probability of using frame 0 is 0.

frame: 1
start-stop 110--310

Probability of using frame 1 is 0.

#gff-version 2
#source-version easygene-1.2b
#date 2011-12-05
#Type DNA
#model: AP02 Aeropyrum pernix
#seqname
#model feature start end score +/- ? static odds
qi 260765442_gb_GQ006550.1 AP02 CDS 292 1026 0.00048760 4 6 #GTC -5.9751

Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg F.... Tscr..

----------- ----- -- ----------- ----- -- -- ----------- ----- -----------

1.01 Sngl + 68 1079 1014 2 0 45 41 371 0.874 75.60

Suboptimal exons with probability > 1.000
12. ecoli300901746

GLIMMER Over. 3.02, iterated Predictions; start and frame score

<table>
<thead>
<tr>
<th>Gene</th>
<th>Start</th>
<th>Frame</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORF0002</td>
<td>33</td>
<td>30</td>
<td>6.94</td>
</tr>
<tr>
<td>ORF0005</td>
<td>4353</td>
<td>5129</td>
<td>4.31</td>
</tr>
<tr>
<td>ORF0009</td>
<td>5114</td>
<td>6048</td>
<td>4.49</td>
</tr>
<tr>
<td>ORF0011</td>
<td>6151</td>
<td>5711</td>
<td>11.15</td>
</tr>
<tr>
<td>ORF0014</td>
<td>7176</td>
<td>6925</td>
<td>5.61</td>
</tr>
<tr>
<td>ORF0018</td>
<td>7202</td>
<td>7028</td>
<td>7.70</td>
</tr>
<tr>
<td>ORF0017</td>
<td>7504</td>
<td>5015</td>
<td>5.71</td>
</tr>
<tr>
<td>ORF0019</td>
<td>8051</td>
<td>8641</td>
<td>10.51</td>
</tr>
<tr>
<td>ORF0020</td>
<td>8881</td>
<td>9141</td>
<td>8.17</td>
</tr>
<tr>
<td>ORF0021</td>
<td>9429</td>
<td>10296</td>
<td>3.12</td>
</tr>
<tr>
<td>ORF0023</td>
<td>10469</td>
<td>10230</td>
<td>3.96</td>
</tr>
<tr>
<td>ORF0026</td>
<td>23</td>
<td>10717</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Command Prompt

frame: 0

start-stop 5188-6048
start-stop 6151-6925
start-stop 6856-7798
start-stop 7222-7798
start-stop 7776-8013
start-stop 8212-8489
start-stop 8881-9141
start-stop 9594-9814

Probability of using frame 0 is 0.33333333333333333.

frame: 1

start-stop 38-4263
start-stop 4442-4654
start-stop 5282-5557
start-stop 5864-6061

Probability of using frame 0 is 0.33333333333333333.

EasyGene 1.2 Server - prediction results

Technical University of Denmark

1. 01 Intrc 89 4384 4296 1 0 44 22 5745 0.999 0.652 37
1. 02 Intrc 4515 5090 576 2 0 30 59 5056 0.923 36.62
1. 03 Intrc 5104 5195 92 0 2 99 04 20 0.522 -0.01
1. 04 Intrc 5364 5078 712 0 0 41 -27 546 0.054 30.48
1. 05 Termc 6220 6774 573 0 0 16 42 774 0.085 60.46
1. 06 Plya 6885 6800 6

2. 00 Promc 7172 7211 40
2. 01 Intrc 7360 7762 403 0 1 62 16 267 0.423 12.88
2. 02 Intrc 7856 8072 217 0 1 48 85 125 0.472 6.29
2. 03 Termc 5152 5704 523 1 1 35 48 436 0.533 28.45
2. 04 Plya 8885 8595 6

2. 00 Promc 8305 8042 40
2. 01 Intrc 8944 9115 172 0 1 84 50 112 0.890 6.57
2. 02 Termc 9423 10861 869 2 2 20 48 923 0.973 75.81

52
Glimmer (ver. 3.02, iterated) predictions:
orfID  start  end  frame  score
>gi|354515237|gb|JF913432.1| Escherichia coli class I integron dihydrofolate reductase  orf00001  593  34  +2  4.50

#gff-version 1
gene-version ekgene-1.2e
gencode-

gene inhibit 1 2 4

1.01 Init +  264  1537  1694  1  1  73  64  440  0.215  34.70
1.02 Term +  1700  2631  932  0  2  63  38  508  0.444  34.61
1.03 PlyA +  2884  2669  6  6
2.00 Prom +  2972  2711  40  6  45  -29  845  0.887  62.20
2.01 Init +  2740  3746  1006  0  1  45  -29  845  0.887  62.20
2.02 Term +  3763  4760  998  0  1  9  38  598  0.759  38.18
2.03 PlyA +  4846  4861  6  6
3.00 Prom +  4868  4907  40  6
3.01 Sgl +  4939  5061  1113  0  0  49  40  425  0.915  30.36
3.02 PlyA +  6086  6091  6  6
4.00 PlyA +  6270  5265  6  6
4.02 Term +  7814  7992  123  0  0  46  46  127  0.352  1.70
4.01 Init +  8992  7877  1116  1  0  4  11  1520  0.594  130.22

Predicted original names:
MDF (dsRNA helicase)  MDF
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
ARSD   (ARSD)      ARSD   (ARSD)      ARSD   (ARSD)
14. ecoli341941295

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Command Prompt screenshot]

Gn. Ex Type S .Begin ... End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

1.01 Term +  132  1361 1230  2  0  56  41  652  0.787  49.47

Suboptimal exons with probability > 1.000

Exnum Type S .Begin ... End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr..

<table>
<thead>
<tr>
<th>Pseudomonas (species)</th>
<th>Start Position</th>
<th>End Position</th>
<th>Strand</th>
<th>Score</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arch/Escherichia coli (strain K2)</td>
<td>830</td>
<td>839</td>
<td>-</td>
<td>6.95</td>
<td>ORTHOSINIA</td>
</tr>
<tr>
<td>Arch/Escherichia coli (strain K2)</td>
<td>839</td>
<td>837</td>
<td>-</td>
<td>6.94</td>
<td>DIOTCINIA</td>
</tr>
<tr>
<td>Arch/Escherichia coli (strain K2)</td>
<td>634</td>
<td>643</td>
<td>-</td>
<td>6.70</td>
<td>OTHANNIA</td>
</tr>
<tr>
<td>Arch/Garindracta (strain K2)</td>
<td>572</td>
<td>585</td>
<td>-</td>
<td>6.66</td>
<td>DOCTHIATRIO</td>
</tr>
</tbody>
</table>

54
GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
</table>

Command Prompt output:

```
#ff-version 2
#source-version easygene-1.2b
#date 2011-12-05
#Type DNA
#model: AP02 Aeropyrum pernix
#exon model feature start end score +/- ? startc odds

Gn_Ex Type S .Begin ... End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr...

----- ---- - ------ ------ ---- -- -- ---- ---- ------ ------ ------

1.01 Intr + 400 685 286 0 1 89 80 355 0.721 31.71

Suboptimal exons with probability > 1.000

Exnum Type S .Begin ... End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr...

<table>
<thead>
<tr>
<th>PWID (species)</th>
<th>Start Position</th>
<th>End Position</th>
<th>Strand</th>
<th>Score</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AroL Escherichia coli (strain K12)</td>
<td>145</td>
<td>154</td>
<td>+</td>
<td>8.69</td>
<td>DEEDITOR</td>
</tr>
<tr>
<td>AroL Escherichia coli (strain K12)</td>
<td>224</td>
<td>223</td>
<td>+</td>
<td>6.45</td>
<td>DEEDITOR</td>
</tr>
<tr>
<td>AroL Escherichia coli (strain K12)</td>
<td>351</td>
<td>358</td>
<td>-</td>
<td>8.02</td>
<td>DEEDITOR</td>
</tr>
</tbody>
</table>
```
16. **ecoli346421495**

**GLIMMER** (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>geneID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>gi</td>
<td>346421495</td>
<td>**gi</td>
<td>35519122.1</td>
<td>Escherichia coli strain C2 beta-lactamase (blaCTX-M) gene, partial cds</td>
</tr>
<tr>
<td>crf40002</td>
<td>88</td>
<td>210</td>
<td>+1</td>
<td>11.94</td>
</tr>
</tbody>
</table>

---

**EasyGene 1.2 Server - prediction results**

**Technical University of Denmark**

```
#gene-version 3
#source-version easygene-1.2b
#date 2011-11-29
#Type DNA
# gene AP02 Aeropyrum pernix
# feature model start end score +/- ? start/odds
#
```

Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>——--</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>1.01</td>
<td>Init +</td>
<td>69</td>
<td>516</td>
<td>443</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Suboptimal exons with probability > 1.000

Exnum Type S .Begin ...End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr..
17. ecoli354515237

GLIMMER (ver. 3.02; iterated) predictions:
orfID  start  end  frame  score
-------  ------  ----  ----  ----- 
>gi|354515237|gb|JF918432.1| Escherichia coli class I integron dihydrofolate
orfo0001   583   34   +2   4.50

Gm. Ex Type S .Begin ... End .Len Fr Ph 1/Ac Do/T CodRg P .... Tscr...

1.01 Init +  705  920  216  2  0  96  -47  229  0.588  8.97

Suboptimal exons with probability > 1.000

Exnum Type S .Begin ... End .Len Fr Ph B/Ac Do/T CodRg P .... Tscr...

57
GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Probability of using frame 1 is 0.1666666666666667.

Probability of using frame 2 is 0.

Suboptimal exons with probability > 1.000
GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
</table>

Probability of using frame 0 is 0.

frame: 1

Probability of using frame 1 is 0.

frame: 2

Probability of using frame 2 is 0.

Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

---------- ----- ------- ---- ------ ------- ----- ------- ---- ------- ---- ------- ---- -------

1.01 Term + 332 855 524 0 2 53 44 546 0.465 41.14

Suboptimal exons with probability > 1.000

Bxnum Type S .Begin ...End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr..
20. ecoli354515243

GLIMMER (ver. 3.02; iterated) predictions:

<table>
<thead>
<tr>
<th>orfID</th>
<th>start</th>
<th>end</th>
<th>frame</th>
<th>score</th>
</tr>
</thead>
</table>

```
Command Prompt

Probability of using frame 0 is 0.
frame: 1
Probability of using frame 1 is 0.
frame: 2
Probability of using frame 2 is 0.
```

```
#gff-version 2
#source-version ecoli_1.2b
#date 2011-12-05
#Type DNA
# model: AP02 Aeropyrum pernix
# feature: start end score +/- ? start0 odds
```

```
Gn. Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..

----- ---- - ------ ------ ---- -- ------ ---- ---- ------ ------ ------ ------

1.01 Intr = 214 92 123 2 0 39 29 178 0.646 8.73

Suboptimal exons with probability > 1.000

Exnum Type S .Begin ...End .Len Fr Ph B/Ac Do/T CodRg P.... Tscr..
```

```
| Primer Sequence: |
|------------------|-----------------|-----------------|-----------------|-----------------|
| 1957 augmented   | 1957 augmented  | 1957 augmented  | 1957 augmented  | 1957 augmented  |
| 1957 augmented   | 1957 augmented  | 1957 augmented  | 1957 augmented  | 1957 augmented  |
| 1957 augmented   | 1957 augmented  | 1957 augmented  | 1957 augmented  | 1957 augmented  |

```

```
Pfam (access): Start Position End Position Strand Score Sequence
```

```
AeAL Esherichia coli (strain K12) 17 26 - 0.10 TTTTTTTTT
```
Appendix 5—Web Application

1. Home page

   ![E. coli Gene Finder]

   **About E. coli Gene Finder**

   **Summary:**

   This project uses the characteristics in TATA-less regions on E. coli sequences to predict the promoter region before TSS, which indicates that the real gene has been located. It uses several well-known algorithms and methods such as the sliding window algorithm and a clustering method to predict promoters. It also contains E2K algorithm and method to compare predicted result with other online promoter package result.

   **Cover:**
   - Start and Stop location
   - TP
   - FDR

   **Project Overview**

   The prediction made by this program is only for some E. coli sequences as we can see from the data from NCBI. Even though the results look better than two of the online packages prediction, it still has the limitation on predictions.

   **Algorithm E2K**

   Algorithm depends on the finding of TATA-less region, the steps during classification and statistics. Improved promoter prediction will use the characteristics from both TATA-rich regions and TATA-less regions to increase both positive prediction result.

2. One example

   ![E. coli 300901746]

<table>
<thead>
<tr>
<th>NCBI start</th>
<th>NCBI stop</th>
<th>Glimmer start</th>
<th>Glimmer stop</th>
<th>Start</th>
<th>Stop</th>
<th>EasyC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A988</td>
<td>5129</td>
<td>4988</td>
<td>5129</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5118</td>
<td>9648</td>
<td>5118</td>
<td>5118</td>
<td>6048</td>
<td>5118</td>
<td>6048</td>
</tr>
<tr>
<td>6151</td>
<td>9711</td>
<td>6151</td>
<td>6151</td>
<td>6711</td>
<td>6151</td>
<td>6711</td>
</tr>
<tr>
<td>6849</td>
<td>7002</td>
<td>6849</td>
<td>6849</td>
<td>7176</td>
<td>6849</td>
<td>7176</td>
</tr>
<tr>
<td>7268</td>
<td>7758</td>
<td>7268</td>
<td>7268</td>
<td>7758</td>
<td>7268</td>
<td>7758</td>
</tr>
<tr>
<td>7804</td>
<td>8013</td>
<td>7804</td>
<td>7804</td>
<td>8013</td>
<td>7804</td>
<td>8013</td>
</tr>
<tr>
<td>8013</td>
<td>8641</td>
<td>8013</td>
<td>8013</td>
<td>8841</td>
<td>8212</td>
<td>8499</td>
</tr>
<tr>
<td>8881</td>
<td>9141</td>
<td>8881</td>
<td>8881</td>
<td>9141</td>
<td>8881</td>
<td>9141</td>
</tr>
<tr>
<td>9429</td>
<td>10298</td>
<td>9429</td>
<td>10298</td>
<td>9574</td>
<td>9204</td>
<td></td>
</tr>
</tbody>
</table>

   - TP
   - FDR

   0.5877777778 147.25