Algorithm for finding coding signal using homogeneous Markov chains independently for three codon positions

Paweł Błażej, Paweł Mackiewicz, StanisławCebrat
Department of Genomics, Faculty of biotechnology
University of Wrocław
ul. Przybyszewskiego 63/77, 51-148 Wrocław, Poland
e-mail: blazej@smorfland.uni.wroc.pl

Abstract—**Many currently used algorithms for protein coding sequences require large learning sets of true genes to estimate sensible values for used parameters which are necessary to make the prediction reasonable. They also fail in recognition of short genes which usually contain weak coding signal. To avoid these problems, we worked out a new algorithm for finding protein coding potential in prokaryotic genomes. This algorithm uses homogeneous Markov chain for modeling nucleotide transition between fixed positions in codons thereby reduces order of Markov chain retaining simultaneously information on dependence between nucleotides in sequence on relatively long distances. We tested performance of this algorithm in relationship to size of the learning set with true and false positive rates for different model orders. We also made some comparisons between our algorithm and commonly used GeneMark. The presented algorithm works better especially for smaller learning sets.**

Keywords; ORF, gene finding, Markov chains

I. Introduction

Although many algorithms using different measures [7] for predicting protein coding sequences in prokaryotic genomes have been developed (see [9] and [1] for recent reviews), there is still an unsolved problem to distinguish true and false coding sequences among short open reading frames (ORFs) fewer than 300 bp. Though majority of these ORFs are spurious, some short genes are likely present in this set. They may encode peptides important for cell functioning, e.g. fulfilling regulatory functions. The number of small ORFs (smORFs) increases exponentially with decrease in their length [8], which hampers to recognize real genes among false frames. Recognition of these genes is also difficult because their coding signal is disturbed by statistical fluctuations coming out from their short sequences. As a result, gene predicting programs that achieve very high rates of detection, accept simultaneously quite a lot of false positives. Moreover, many of these algorithms rely only on large learning sets of true genes which are necessary to make reliable estimation of used parameters. Therefore, they are not optimal for small bacterial genomes which encode smaller sets of real genes. Then, to reach the proper size of learning sets, some non-coding ORFs are probably included in the training procedure. It may additionally increase the false positive rate in the stage of gene recognition. Furthermore, more general models which are assumed to be universal for a wide range of genomes are not appropriate for some, especially small genomes which are characterized by a specific nucleotide or codon bias. To avoid these problems we developed a suitable statistical model which can be useful for detection a protein coding signal. This model utilizes specific properties of protein coding sequences related to correlations in nucleotide composition in particular codon positions, which was observed both in prokaryotic [5] and eukaryotic genomes [6]. Our algorithm uses homogeneous Markov chains to analyse this coding information on long distances in particular codon positions (separately for the first, the second and the third) and does not require high chain order to work properly. The new method was compared with commonly used GeneMark also based on Markov chains [3].

II. Algorithm for finding a coding signal

The most common gene finders use Markov chain approach for modeling dependences between occurrence of nucleotides in a genome [4], [3]. Our method uses six homogeneous Markov chains of protein coding sequences to determine the positional pattern frequencies which are used to detect a coding signal in analyzed sequences. This algorithm consists of two stages: the training step and the analysis step.

A. *Training step*

The main task of this step is to compute model parameters which are calculated from a learning set of nucleotide sequences. For a given genome such a set is in fact a collection of ORFs annotated with ascribed function in GenBank database, excluding ORFs that were described as questionable or hypothetical.

1) *Construction of transition matrices*

Let us consider **S**=$\{S_{i1}, S_{i2}, ..., S_{in}\}$ a sequence of nucleotides extracted from fixed codon positions (i = 1, 2, 3) in a protein coding sequence. We construct the initial probabilities $P(S^h_i)$ of h nucleotides S_i situated in the same codon positions i (where h defines the model order) and also the probability transition matrices (i.e. between nucleotides in the same codon position). Matrices M_1, M_2, M_3 concern to direct (sense) strands of training sequences whereas matrices M_4, M_5, M_6 are based on complementary strands of these sequences (antisense). Matrices M_4, M_5, M_6 are useful for a model of ”shadow” coding regions. Obviously matrices M_1, ..., M_6 are transition matrices for homogeneous Markov chains.

2) Determination of positional pattern frequencies

The obtained matrices are used to determine vectors of positional pattern frequencies in the learning set. The positional pattern is a vector of indices of matrices that give the highest value of total probability for a given codon position. In sum, there are 216 such potential patterns i.e. 111, 112, 113, etc. It is easy to notice that in this case we actually used a maximum likelihood approach. The frequencies of these vectors are obtained as follows:

1. Each sequence in every reading frame is analyzed by moving windows with a fixed length (e.g. 96 nt) and a fixed window shift (e.g. 12 nt);
2. For each window a vector of digits (d_1, d_2, d_3) (called the positional pattern) is determined in the following way:
    a. For each of three codon positions probabilities P_{M1}, P_{M2}, P_{M3}, P_{M4}, P_{M5}, P_{M6} are calculated by using trained matrices M_1, M_2, M_3, M_4, M_5, M_6, respectively;
    b. if P_{Mj} =max(P_{M1}, P_{M2}, P_{M3}, P_{M4}, P_{M5}, P_{M6}) (for fixed codon position i), then d_i = j and finally a positional pattern (d_1, d_2, d_3) is obtained;
3. The frequency for each positional pattern are calculated from all analyzed windows which are made of the learning set for each reading frame (Fig. 1).

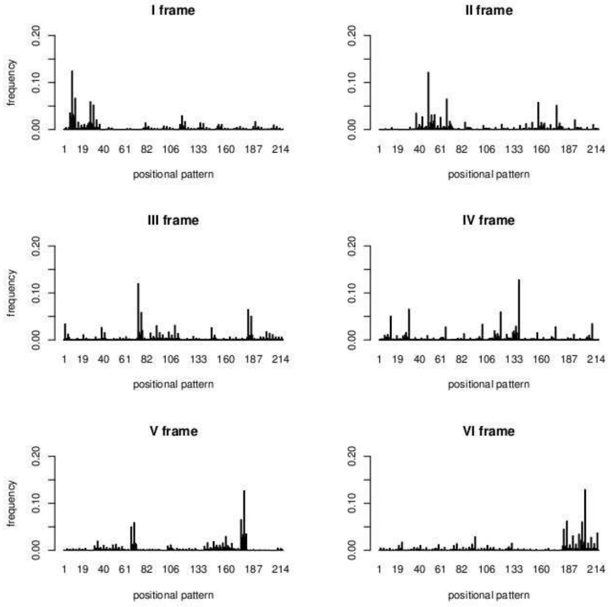

Fig. 1. Barplots of positional pattern frequencies computed for the training set from *Escherichia coli* genome for six reading frames.

B. Test or analysis step

The aim of this step is to detect the correct reading frame for an analyzed DNA sequence.

The first two steps are the same as in determination of positional pattern frequencies (subsection II.A):

1. As 1 in II.A.2;
2. As 2 in II.A.2;
3. For a positional pattern (d_1, d_2, d_3) found for every window and every reading frame, we ascribe a respective frequency P_1, P_2, P_3, P_4, P_5, P_6 which were determined previously for the learning set;
4. As an additional non-coding reference we assume uniform distribution of positional pattern frequencies and introduce $P_7 = 1/216$;
5. For every window we obtain a coding signal vector of frequencies for six reading frames plus the non-coding reference:
$$\left(\frac{P_1}{\sum_{i=1}^{7} P_i}, \frac{P_2}{\sum_{i=1}^{7} P_i}, \ldots, \frac{P_7}{\sum_{i=1}^{7} P_i}\right)$$
6. Finally, the respective elements of the coding signal vector are averaged over all windows for a given sequence. The sequence is coding in frame i if the i position in coding signal vectors has the highest value.

The idea of the presented algorithm is similar to the algorithm which was introduced in the paper [2]. The main difference is the extension of the set of possible positional pattern frequencies from 27 to 216. The new approach takes into account all possible frequencies obtained by using matrices M_1, M_2, ..., M_6 at once. This approach gives better results especially in genomes with strong coding signal in the complementary (antisense) strand (e.g. in *E. coli* genome).

III. Results

We have tested our algorithm on *Escherichia coli* 536 genome and also have analyzed several small *Mycoplasma* genomes. To evaluate efficiency of our algorithm we measured true positive rate (sensitivity) and false positive rate. For fixed model orders (h = 2, 4) we also compared our results with the results obtained by GeneMark 2.5 (h = 2, 5) software using *E. coli* as a reference genome.

A. Analysis of Escherichia coli genome

1) Estimation of true positive rate

The whole set of annotated ORFs as protein coding sequences (2773 ORFs) was divided into two parts:

1. training set (1000 ORFs);
2. test set (the rest 1773 ORFs).

Furthermore, from the training set we chose randomly subsets containing increasing number of ORFs, i.e. 100, 200, ..., 1000 ORFs which we used as training sets. Our aim was to find dependences between true positive rate in the test set and the size of the learning set for fixed model order h = 1, 2, 3, 4. These results averaged on 20 simulations are presented in Fig. 2. The fraction of correctly recognized genes increases rapidly with the learning set size and stabilizes from the set of 300 or 400 ORFs. Interestingly, lower order models perform much better for smaller learning sets than the most complex one (h=4) which slightly surpasses the simpler models for larger learning sets.

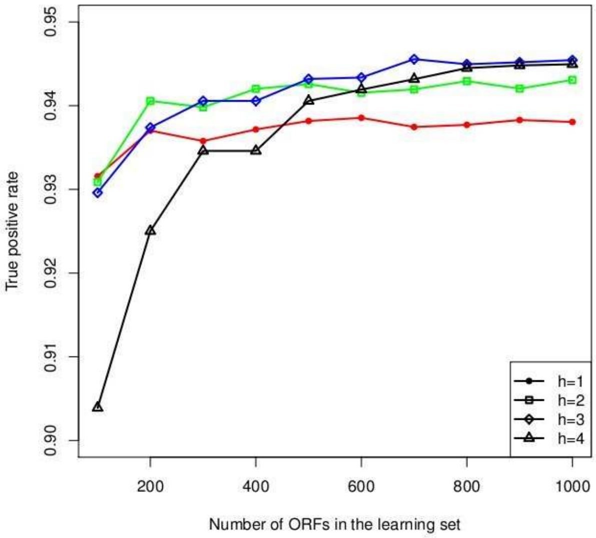

Fig 2. Relationship between true positive rate and the size of training set.

2) Estimation of false positive rate

We estimated false positive rate using two test sets:

1. protein coding sequences in incorrect reading frame;
2. random sequences generated according to the genome nucleotide composition and the length distribution of real genes.

The results averaged on 20 simulations are shown in Fig. 3. The relationships between false positive rate and the learning set size is differs for the two test sets. When ORFs in the incorrect reading frame are used as a test set, false positive rate of the h=4 model is higher than the rate of the simpler models for the smaller learning sets but is lower for the larger learning sets. The rate increases with the size for generated sequences and decreases for ORFs read in incorrect frame. In the case of the generated sequences, the high order model (h=4) receives the lowest false positive rate for all learning sets in comparison to the simpler models.

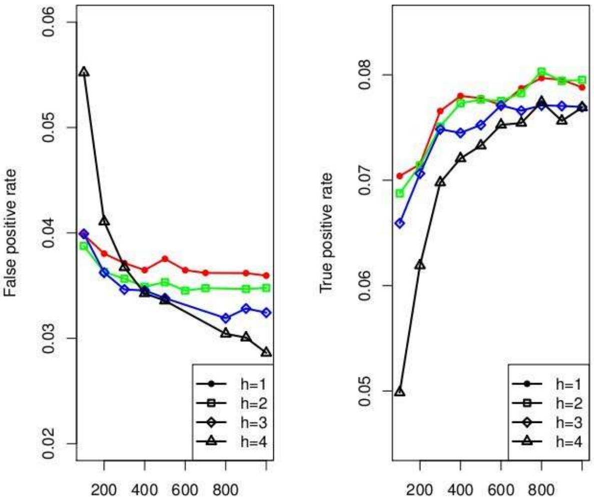

Fig 3. Relationship between false positive rate and the size of the training set for: sequences in incorrect reading frame (in the left) and randomly generated sequences (in the right).

3) Comparison of the new algorithm with GeneMark

We used the same learning and test sets for both algorithms and compared the new model of the order h = 2, 4 with GeneMark of the order h = 2, 5. We chose the GeneMark order of h=5 because it is the most common used order in the current GeneMark version 2.5. Performance of these two algorithms according to true positive rate in relationship to the size of learning set is shown in Fig. 4 and Fig. 5. All algorithms achieve true positive rate higher than 0.93. For low order models (Fig. 4) the new algorithm receives higher true positive rate than GeneMark, more than 0.94 for all learning sets with exception to the smallest one. When more complex model are used (Fig. 5) the new algorithm still works better for all learning sets but the difference between two algorithms diminishes with the learning set size and two algorithms converge for the set consisting of 1000 ORFs achieving true positive rate about 0.945.

Comparison of two methods regarding relationship between false positive rate and the size of learning set is presented in Fig. 6 and Fig. 7. The relationship is weaker than for true positive rate. Interestingly, performance of two algorithms depends on the test set. The new algorithm has lower false positive rate for incorrect reading frames with the order of h=2 and for random sequences with the order of h=4 while GeneMark performs better in the case of incorrect reading frames with the order of h=5 and for random sequences with the order of h=2. By average the two algorithms show similar 0.055 false positive rate. GeneMark achieves both the lowest and the highest false positive rate values.

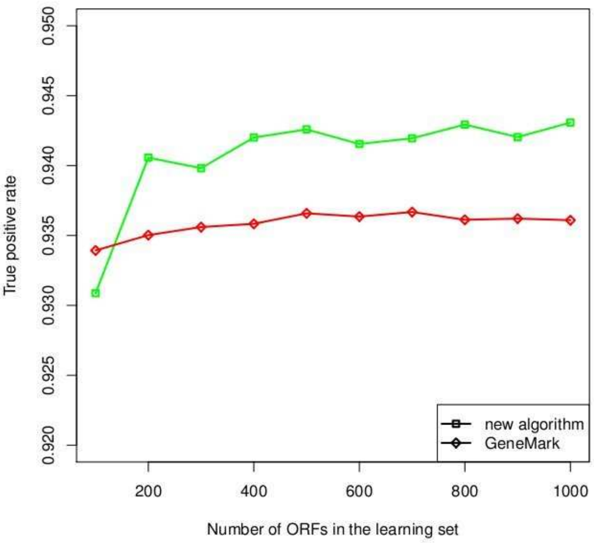

Fig 4. Comparison of true positive rates between our algorithm (green) and GeneMark 2.5 (red) for model order h=2.

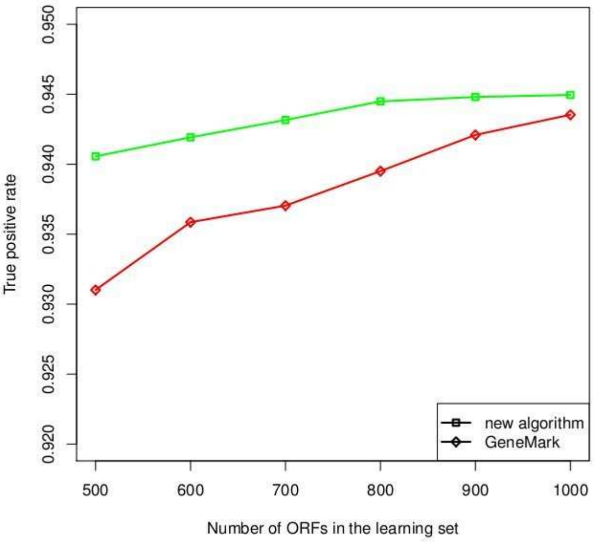

Fig 5. Comparison of true positive rates between our algorithm (green) and GeneMark 2.5 (red) for model order h=4 and h=5, respectively.

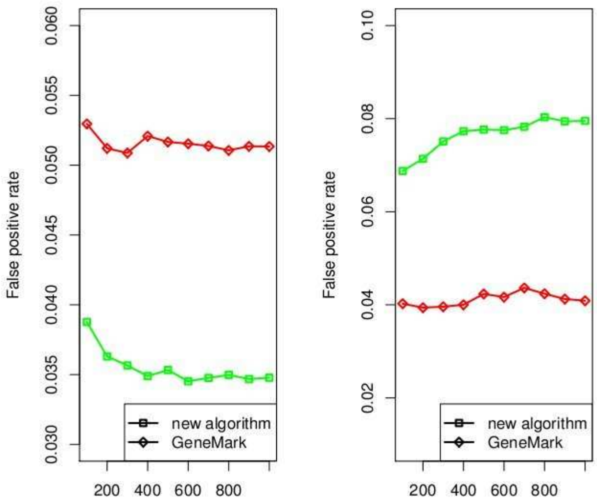

Fig 6. Comparison of false positive rates between our algorithm (green) with h = 2 and GeneMark (red) with h = 2 for: sequences in incorrect reading frame (in the left) and random sequences (in the right).

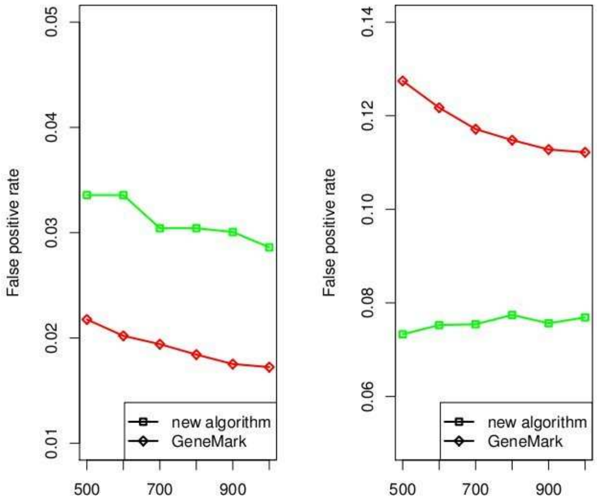

Fig 7. Comparison of false positive rates between our algorithm (green) with h = 4 and GeneMark (red) with h = 5 for: sequences in incorrect reading frame (in the left) and random sequences (in the right).

4) Comparison of coding signal

The main task of our algorithm is to find a sequence with coding signal in a proper reading frame. In Fig. 8 we compared the strength of the coding signal for model order of h=2 in different group of sequences: protein coding sequences, sequences in incorrect reading frame and random sequences. The strength was described by empirical tail distribution functions (i.e. 1-F(x) = P(X > x)), where X is a random variable of the value of strongest coding signal. The distribution for protein coding sequences is clearly shifted towards higher values of coding signal. Protein coding sequences with coding signal higher than 0.3 are over 91% while there are only 13% of incorrect ORFs and almost no random sequences (0.9%) above this value.

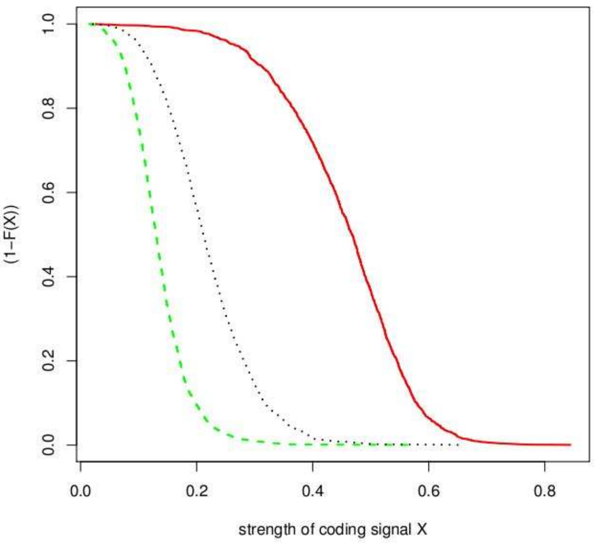

Fig 8. Comparison empirical tail distribution functions (1-F(X)) for: protein coding sequences (red solid line), sequences in incorrect reading frame (black dotted line), random sequences (green dashed line).

5) *Small genomes*

TABLE I. TRUE POSITIVE RATE (TPR) FOR SMALL MYCOPLASMA GENOMES.

Genome	TPR
M. agalatiae (0.88 Mbp)	0.97
M. arthihritidis 158L3 1 (0.82 Mbp)	0.96
M. mobile 163K (0.78 Mbp)	0.91
M. mobile 163K (0.78 Mbp)	0.97
M. pulmonis UAB CTIP (0.96 Mbp)	0.94
M. synoviae 53 (0.8 Mbp)	0.97

As was mentioned in Introduction, one of the most important problems in recognition of protein coding sequences is difficulty in obtaining a large enough training set for small genomes. Here, we tested the new algorithm in the case of small genomes assuming tiny learning sets (Table I). For every genome we chose randomly 200 annotated ORFs the training set and the rest of ORFs was used to build the test set. Sets for calculation false positive rate were prepared similarly but were based on ORFs read in incorrect frames. The algorithm achieved true positive rate higher than 0.90 and false positive rate below 0.1.

IV. CONCLUSION

The presented algorithm describes nucleotide transition in three codon positions independently. Therefore it reduces order of Markov chain retaining the same coding information that is contained in higher order chains analyzing dependence between nucleotides in subsequent positions of a sequence. This algorithm achieved good performance both for small and large learning sets. In our test we obtained average true positive rate over 0.90 and false positive rate below 0.1. Models of lower order worked usually better for smaller learning sets but the most complex ones were better for larger ones. However, the difference both in true positive rate and false positive rate between models of different order was bigger for the small learning sets than for larger ones. Models with higher order showed stronger relationship with the size of learning set than simpler ones. Our results indicate that our algorithm is comparable with GeneMark algorithm according to false positive rate but achieves higher true positive rate. Since the new algorithm work well under low order models.

REFERENCES

[1] R.K. Azad, 2008, “Genes in prokaryotic genomes and their computational prediction“, in: Computational methods for understanding bacterial and archeal genomes, Series of advances in Bioinformatics and Computational Biology, vol. VII, Y. Xu, J.P. Gogarten (Eds.), College Press, pp. 39-74.

[2] P. Błażej, P. Mackiewicz, S. Cebrat, 2010, “Using genetic coding wisdom for recognizing protein coding dequences”, Procedings of the 2010 International Conference on Bioinformatics & Computational Biology, BIOCOMP 2010, Las Vegas Nevada, USA, vol. 1, pp. 302-305.

[3] M. Borodovsky, J. Mcinch, 1993, “Genemark: pararell gene recognition for both dna strands”, Comput. Chem., 17, pp. 123-133.

[4] M. Yu. Borodovsky, Y.A. Sprizhitskii, E.I. Golovanov, A.A. Aleksandrow, 1986, “Statistical Patterns in Primary Structures of the functional Regions of the Genome in *Escherichia Coli*”, Molecular Biology, 20, pp. 826-833, 833-840, 1144-1150.

[5] S. Cebrat, M.R. Dudek, P. Mackiewicz, M. Kowalczuk, M. Fita, 1997, “Asymetry of coding versus non-coding strand sequences of different genomes”, Microbial and Comparative Genomics, 2 (4), pp. 259-268.

[6] S. Cebrat, M.R. Dudek, P. Mackiewicz, 1998, ”Sequence asymmetry as a parameter indicating coding sequence in *Saccheromyces cerevisiae* genome”, Theory in Biosciences, 117, pp. 78-89.

[7] J. Fickett, C. Tung, 1992, “Assesment of protein coding measures”, Nucleic Acid Research, 20 (24), pp. 6441-6450.

[8] A. Gierlik, P. Mackiewicz, M. Kowalczuk, M.R. Dudek, S. Cebrat, 1999, „Some hints on Open Reading frame statistics – how ORF length depends on selection”, Int. J. Modern Phys. C, 10 (4), pp. 645-643.

[9] W.H. Majoros, 2007, “Methods for computational gene prediction”, Cambridge University Press