A Generic Criterion for Gene Recognitions in Genomic Sequences

Chungfan Kim [1] **Akihko Konagaya** [1] **Kiyoshi Asai** [2]

ckim@jaist.ac.jp kona@jaist.ac.jp asai@etl.go.jp

[1] Japan Advanced Institute of Science and Technology, 1-1 Asahidai Tatsunokuchi, Ishikawa 923-1292, Japan

[2] Electrotechnical Laboratories, 1-1-4 Umezono, Tsukuba 305-8568, Japan

Abstract

In this paper, we evaluated the complexity and accuracy of dicodon model for gene finding using Hidden Markov Model with Self-Identification Learning. We used five different models as competitors with smaller parametric space than the dicodon model. Our evaluation result shows that the dicodon model outperforms other competitors in terms of sensitivity as well as specificity. This result indicates that the dicodon model can not be represented by a combination of the pair amino-acid, the codon usage, and the G+C content.

1 Introduction

Gene-finding, a computational method to find protein coding regions from un-annotated genome sequences, has been studied extensively and many systems have been developed until now. HMM (*i.e.* Hidden Markov Model) has been widely used for the gene-finding [1, 2, 3, 4]. Because the genes have a non-deterministic and probabilistic structure like a natural language, computational linguistic methods are effective for describing genomic structures [1]. In order to understand the structures of the coding regions, it is necessary to integrate statistics and computational linguistics for DNA sequence analysis [5]. A coding region can be described as a probabilistic model, and it definitely requires understanding biological and stochastic attributes of the coding regions.

The dicodon model, along with its relatives; the hexamer model and the 5th markov model, has been recognized as one of the most effective models for gene finding. It discerns coding regions according to conditional codon usages of coding regions in a targeted genomic sequence.

There is a peculiar bias among dicodons, and the peculiarity varies among every species. Henceforth, the dicodon model need to acquire *dicodon features* of coding regions from each targeted sequence. Consequently, we encounter the difficulty to gain a “generally correct data set” before the process of gene finding. This means that the dicodon model has to perform learning without referring correct answer during its learning phase.

The self-identification learning [2, 7] is used in our gene finding system in order to realize the learning without correct answer. The learning method do not need to refer correct answers during its learning phase. The method solves the difficulty in a following *boot-strapping* way:

- It simply starts its learning with uniform learning parameters.
- The first trial finds several coding regions with uniform initial parameters.
- Re-calculate parameters (*i.e.* dicodon usages) according to the regions found.
- Iterate learning with revised parameters until it reaches plateau of learning curve.

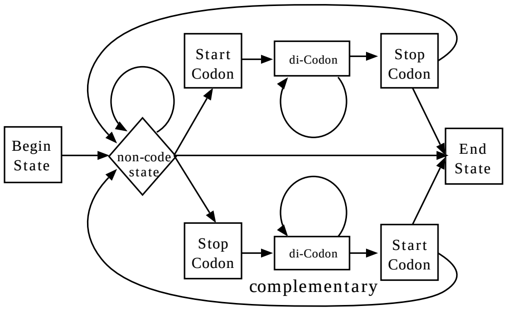


Figure 1: A network diagram of a dicodon oriented HMM. "Start Codon" state emits possible three start codons (ATG, TTG, GTG). "Stop Codon" state emits possible three stop codons (TAA, TAG, TGA). "di-Codon" state emits possible 61 dicodons iteratively.

However, efficiency of the self-identification largely depends on the number of its learning parameters as well as the size of training data. When it employs a large set of parameters, it requires a large set of training data. The model is not accurate with insufficient training data. On the other hand, the model is not accurate when the number of parameters is too large for the amount of training data. This problem can be generalized as a problem of complexity and accuracy of a model. Hence we have to consider *trade-off* between the complexity and the accuracy.

We used a dicodon oriented HMM [2] (Fig. 1) gene finding system with self-identification learning as a test-bed for evaluation of complexity/accuracy of the dicodon model. The self-identification learning has an advantage that it requires only partial training data to perform reasonable gene finding in our preliminary examination (see Fig. 3 and Table 1). The dicodon model employs 3,721 (as of possible combinations of 61 codons except stop codons) parameters to be trained. Our examination clarified that smaller parametric space than 3,721 is sufficient for the gene finding. In this paper, we evaluated six models with smaller parametric space defined by a set of biological attributes such as pair amino-acid, codon usage, and G+C content in order to find what the significances of these attributes in a dicodon model is. The models are evaluated in both approximation error and specificity/sensitivity aspects against the dicodon model.

2 Models

There are 61 possible codons, possible dicodon counts up to 3,721. Hence the size of the parametric space of the dicodon model is 3,721. The size matters when we examine gene finding that uses *self-identification learning*. Self-identification learning with too many parameters usually fails because it requires too large training data while they are not sufficiently available. On the other hand, accuracy of a model hardly gets high enough when the model conveys too few parameters.

Fickett and Tung [6] evaluated many protein coding measures including diamino acid, codon usage, and dinucleotide bias. These measures never perform better than dicodon usage. However, dicodon can be represented by combinations of these well known biological attributes in certain degree.

We presumed that the product of diamino acid, codon usage, and G+C content emulates dicodon usage very well. Because, (i) there presumably are structural information of proteins embedded in coding regions that corresponds to the diamino acid information. The diamino acid information employs fairly larger amount of information ($20 \times 20 = 400$ parameters) than the information derived by a pair of dinucleotides ($16 \times 16 = 256$ parameters). (ii) codon usage determines third nucleotide which follows a couple of nucleotides determined by an amino acid. The amino acid information is derived from diamino acid information. (iii) the third nucleotide might have a modification according to G+C content.

Based on the idea (i) to (iii), we defined the models B to F. The model B is a simple product of diamino acid and codon usage and it does not use C+C content in order to evaluate how this model behave worse than those using G+C content information. The models C and D include correction term. In the model D, we supposed a certain bias among each nucleotide instead of seeing G-C and A-T are identical respectively. In this model, the codon usage is modified by a relation between its own third nucleotide and that of preceded codon. The model E uses two codon usage sets, which are used selectively regarding G+C content of the preceded codon. The model F uses four codon usage sets, which are used based on nucleotide-wise rather on G+C content-wise. The model G is more similar to the dicodon model than the other models. Because this model is a dicodon model without distinction of G-C and A-T at its third nucleotide. The model conveys smaller parameter size(1,024) than that of the dicodon, but it is the largest among the other emulator models.

When these models perform well enough in comparison with dicodon model, that would help us to clarify which attribute is the most crucial to the dicodon model.

A) the dicodon model:
$61 \times 61 = 3,721$ parameters

$$p_A(c_j|c_i) \equiv p(c_j|c_i). \tag{1}$$

B) model of pair amino-acid and codon usage:
$20 \times 20 + 61 = 461$ parameters

$$p_B(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j)). \tag{2}$$

C) model of pair amino-acid and codon usage modified by G+C content:
$20 \times 20 + 61 + 2 = 463$ parameters

$$p_C(c_j|c_i) \equiv p(A(c_j)|A(c_i))\{\lambda_B p(c_j|A(c_j)) + (1-\lambda_B)p(f_{gc}(c_j)|f_{gc}(c_i))\}. \tag{3}$$

D) model of pair amino-acid and codon usage modified by pair G+C content:
$20 \times 20 + 61 + 4 \times 4 = 478$ parameters

$$p_D(c_j|c_i) \equiv p(A(c_j)|A(c_i))\{\lambda_C p(c_j|A(c_j)) + (1-\lambda_C)p(f_{atgc}(c_j)|f_{atgc}(c_i))\}. \tag{4}$$

E) model of pair amino-acid and codon usage with G+C content dependency:
$20 \times 20 + 2 \times 61 = 522$ parameters

$$p_E(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j), f_{gc}(c_i)). \tag{5}$$

F) model of pair amino-acid and codon usage with pair G+C content dependency:
$20 \times 20 + 4 \times 61 = 644$ parameters

$$p_F(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j), f_{atgc}(c_i)). \tag{6}$$

G) model of *shrunk* dicodon usage:
$32 \times 32 = 1024$ parameters

$$p_G(c_j|c_i) \equiv p(S(c_j)|S(c_i)). \tag{7}$$

$A(c)$ stands for an amino-acid which corresponds to a codon c.

The function $f_{gc}(c)$ returns “GC” if the third nucleotide in a codon c is “G” or “C”. Otherwise it returns “AT”. Henceforth, the probability $p(GC|AT)$ stands for a probability to have a codon looks like “XXG” or “XXC” right after a codon “XXA” or “XXT”. Another function $f_{atgc}(c)$ returns the third nucleotide of a codon c. $p(c_j|A(c_j), f_{gc}(c_i))$ represents two codon usages. One is a codon usage observed right after a codon which has “G” or “C”. The another is a codon usage observed right after a codon which has “A” or “T”. $p(c_j|A(c_j), f_{atgc}(c_i))$ represents four codon usages that correspond to a third nucleotide of a preceded codon c_i. λ is a *weight* coefficient. It is calculated so that the square error between the dicodon model become minimal. For model G, $S(c)$ represents *shrunk* codon. *Shrinked* codon does not distinguish G-C, and A-T. For instance, $S(XXG) = S(XXC)$ and $S(XXA) = S(XXT)$.

3 Evaluation of models

In order to evaluate these six models(B to G) against the dicodon model, We used 17 microbial genomic sequences [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24] and *C. elegance* [25] genome sequences obtained from GenBank and took following procedure:

1. So-called *Jack knife strategy* is applied here.
2. Several size of *Learning* sets and *Testing* sets are prepared in order to evaluate performance and robustness of each model.
3. When an examined genomic sequence has N genes, we take N/n genes out of the sequence randomly($n = 1.3, 1.7, 2, 4$).
4. The extracted genes are used for the *Learning* sets.
5. Rest of the genes and the non-coding regions are used as the *Testing* set.
6. Train six models including the dicodon model using the *Learning* set.
7. Accumulate coding potentials cod_x of every coding region in the *Testing* set based on the six models.
8. Train the dicodon model using the *Learning* set and accumulate a coding potential cod_o.
9. Obtain profiles of coding potentials for coding regions and non-coding regions.
10. Evaluate every models in two ways: Approximation error and Learning/Testing evaluation.

A coding potential of a coding region $\mathbf{C} = (c_1, c_2, \ldots, c_n)$ which consists of n codons can be computed as follows:

$$cod(\mathbf{C}) = \frac{1}{n} \log p(c_1, c_2, \ldots, c_n) = \frac{1}{n} \log\{p(c_2|c_1) \ldots p(c_n|c_{n-1})\} = \frac{1}{n} \sum_{i=2}^{n} \log p(c_i|c_{i-1}). \tag{8}$$

3.1 Approximation error

The models B to F are approximations of the dicodon model. Therefore, we can evaluate these models in terms of approximation error of each models against the dicodon model.

- We split a sequence into the learning sequence and the testing sequence.
- M_{DT} is the dicodon model that was trained with testing sequence.
- M_{DL} is the dicodon model that was trained with learning sequence.
- Other models M_B to M_G are all trained with learning sequence.
- Compute coding potentials of coding/non-coding regions in the testing sequence for every model.
- We calculated square errors between coding potentials of M_{DT} and coding potentials of the other models.
- This evaluation shows how these models accurately approximate the dicodon model.

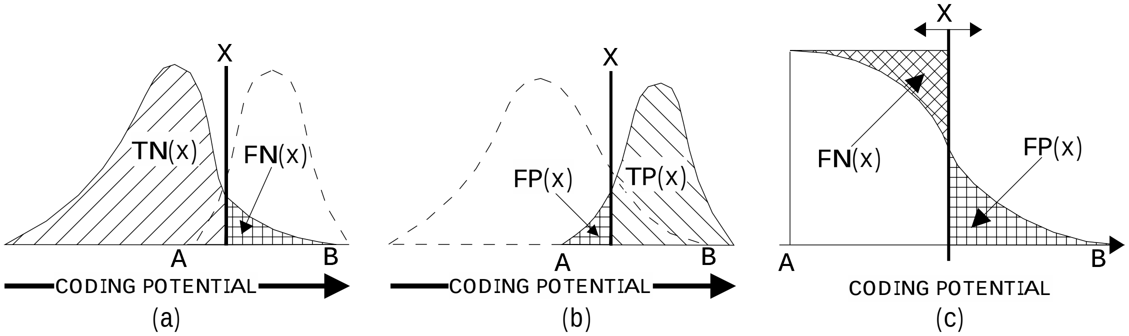


Figure 2: Profile of coding potentials for coding(right heap) and non-coding(left heap) regions. The two heaps have overlapped area $[A, B]$. We set a threshold coding potential x within $[A, B]$. (a)For coding potentials over x are taken to be coding regions. So cross-hatched area become *false negatives.* (b)For coding potentials under x are taken to be non-coding regions. The cross-hatched area become *false positives.* We take x so that the sensitivity and specificity become equivalent.

3.2 Evaluation of Learning/Testing

Here we define a measure to evaluate an accuracy to distinguish coding regions and non-coding regions for each model. Then compute "distances", based on the measure, between profiles of coding/non-coding regions, and evaluate specificity/sensitivity of six models based on the distance(defined below) of each model.

We obtained profiles of coding/non-coding regions look like Fig. 2. Two heaps of coding/non-coding regions are overlapped each other in certain degree. When we have a coding potential x for a predicted coding region, and the potential goes a midst of two heap, it has a probability to belong to a coding region and another probability for a non-coding region simultaneously.

When the overlap, based on a model, is wider than that of other model, we need to do a stochastic decision for every predicted coding region whether it belongs to coding or non-coding regions more frequently than other model. This means we have to make one more *guess* after prediction of coding region.

On the other hand, if a model has narrower overlap, most predicted coding regions are easily distinguished without *guess.*

This can be a measure for relative accuracy of a model against other models.

Then, we defined a *distance* d using the measure described above(see Fig. 2).

$$\begin{aligned} d = \mathit{sensitivity}(x_0) + \mathit{specificity}(x_0) \quad &, \quad \mathit{sensitivity}(x_0) = \mathit{specificity}(x_0) \\ \mathit{sensitivity}(x) = \frac{TP(x)}{TP(x)+FN(x)} \quad &, \quad \mathit{specificity}(x) = \frac{TP(x)}{TP(x)+FP(x)} \\ TN(x) = \sum_{i=x_{min}}^{x} h_{nc}(i) \quad &, \quad FN(x) = \sum_{i=x}^{x_{max}} h_{nc}(i) \\ TP(x) = \sum_{i=x}^{x_{max}} h_{cd}(i) \quad &, \quad FP(x) = \sum_{i=x_{min}}^{x} h_{cd}(i) \end{aligned} \tag{9}$$

As shown above, we take d of the equilibrium where sensitivity and specificity become equivalent.

Table 1: Transition of Learned Parameter Size and Recognition rate for *A.fulgidus*. Notice that the dicodon model keeps 93.6 % of recognition rate even though its learned parameter size dropped to 1,278(40% of maximum size of parameters). This indicates that the dicodon model has a certain redundancy for recognition of coding regions.

Seq. Len.	Parm. Size	Recog.%
2,178,400	3,189	93.8
1,000,000	2,822	93.8
500,000	2,345	93.8
300,000	1,924	93.9
150,000	1,278	93.6
75,000	873	93.1
32,500	520	90.4
15,000	286	87.0
7,500	176	74.8

4 Result

Fig. 3 shows a correlation between the length of learned sequence and consequent recognition score for 17 microbial genomes. Interestingly, the score stays atop while the length of learned sequence decreases. We found a correlation between the recognition score and the number of parameters which have non-zero values after learning (see Table 1). The above result indicates that the dicodon model tends to be redundant.

Table 4 shows a comparison of specificity and sensitivity (*distance*) d (see equation 10) for each models.

Fig. 5 shows comparisons of average square errors for coding region and non-coding region. The square errors are calculated against coding potential of dicodon model for each six models(B to G).

5 Discussion

Our evaluation shows that the dicodon model outperforms other six models(B to G) in terms of specificity and sensitivity (Fig. 4). Besides, none of the emulation models (B to F) get closer than the model G in terms of approximation error (Fig. 5).

The models B to F apparently failed emulating the dicodon model. This means that the information among a pair of codon conveys richer feature of coding regions than a mere combination of the diamino, codon usage, and G+C content, and the diamino acid simply drops some crucial information in the coding region.

Performance of the model B, which is the simplest, is constantly low among the other models. This corresponds to an evidence of the significance of G+C content.

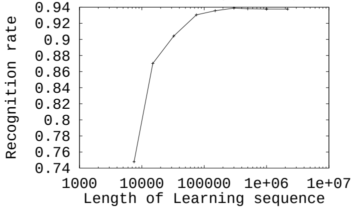
The model C performs slightly better than the B but it is not so apparent. While the model C has information of G+C content, linear interpolation of codon usage and G+C content did not work so much in this case.

The model D performs better than the B and C. Although the differences of its performance between this model and the B, C are clearer than that of B and C, its performance improvement is poor. However, we should notice that nucleotide-wise bias at the third nucleotide is more significant than G+C content.

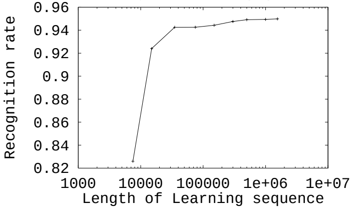
The performance of the model E shows clearer improvements. This result indicates that the second codon usage depends on the G+C content of the first codon.

The result of the model F is the best among the models B to F. With this result, there apparently is dependency of the second codon usage on the third nucleotide of the first codon rather on the G+C content. This indicates that a bias at the third nucleotide is not so uniform among G-C and A-T, and

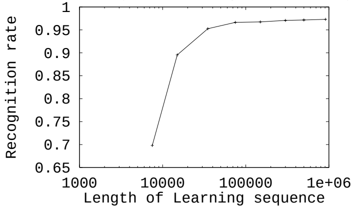
Base lavel Recognition rate(Archaeoglobus fulgidus)



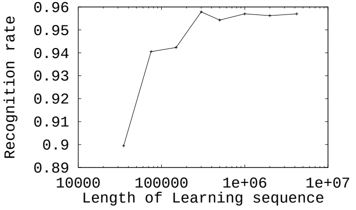
Base lavel Recognition rate(Aquifex aeolicus)



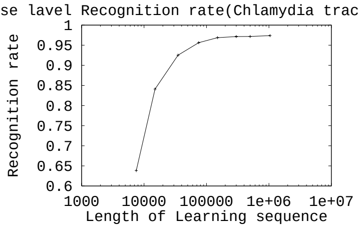
Base lavel Recognition rate(Borrelia burgdorferi)



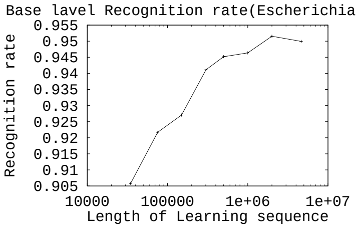
Base lavel Recognition rate(Bacillus subtilis)



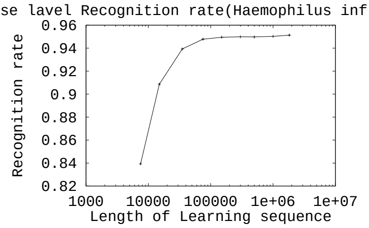
Base lavel Recognition rate(Chlamydia trachomatis)



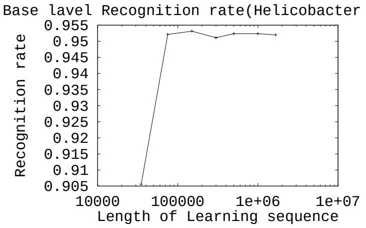
Base lavel Recognition rate(Escherichia coli)



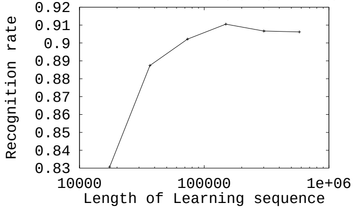
Base lavel Recognition rate(Haemophilus influenzae)



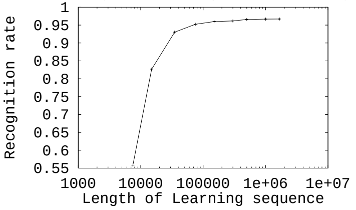
Base lavel Recognition rate(Helicobacter pylori)



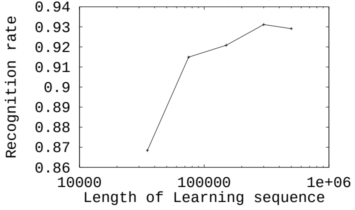
Base lavel Recognition rate(Mycoplasma genitalium)



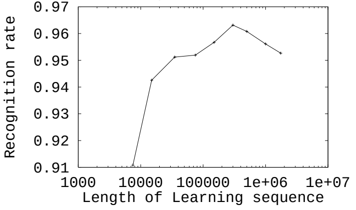
Base lavel Recognition rate(Methanococcus jannaschii)



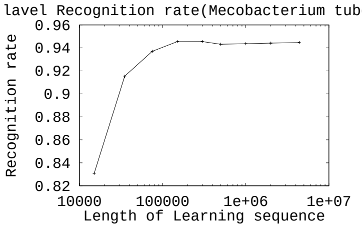
Base lavel Recognition rate(Mycoplasma pneumoniae)



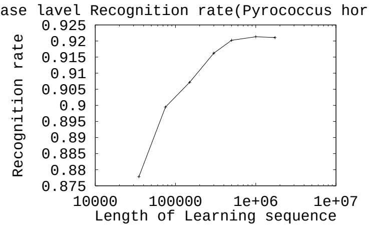
Base lavel Recognition rate(Methanobacterium thermoautotrophicum)



Base lavel Recognition rate(Mecobacterium tuberculosis)



Base lavel Recognition rate(Pyrococcus horikoshii)



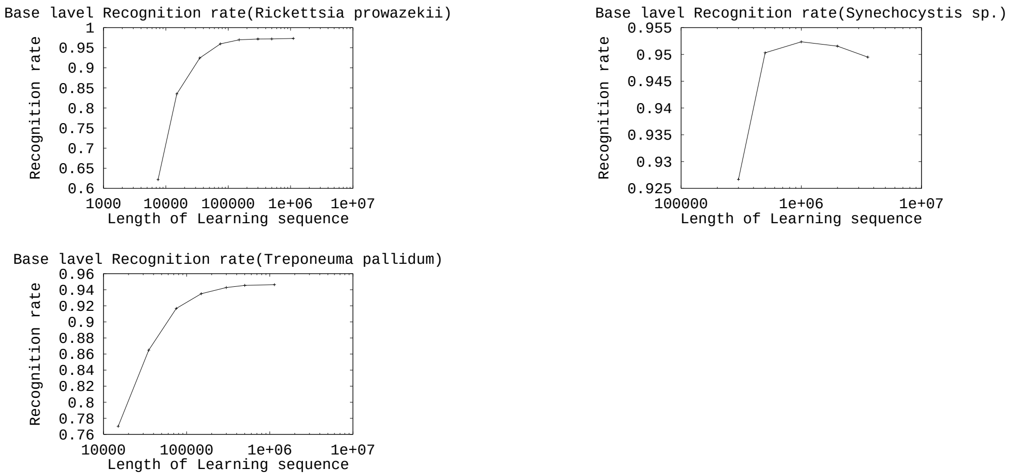


Figure 3: Transition of Recognition Rate for 17 microbial genomes using dicodon-oriented HMM with Self-Identification Learning. We used 1000,000, 500,000, 300,000, 150,000, 75,000, 32,500, 15,000, and 7,500bp of fragments out of complete genomic sequences for training data of the dicodon HMM. We observed that the recognition rate remains a top until the length of training data drops less than 150,000bp.

G+C content model is not sufficient for describing this bias. Therefore we should consider A, T, C, G individually.

The model G scores the nearest performance to the dicodon model. Let us take a look at this result not from performance improvement but from performance decline. Only difference between this model and the dicodon model is that this model does not distinguish G-C and A-T at the third nucleotide. Again, this shows that the peculiar bias at third nucleotide that is indicated by the result of the model D and F.

Considering the difference between diamino and dicodon, dependency of the third nucleotide of second codon on the first codon is important for describing superiority of the dicodon. Although the diamino acid and codon usage are undoubtedly important attributes of dicodon, our result shows that G+C content is not enough for describing peculiar bias which is found at the third nucleotide.

Acknowledgments

This work has supported in part by the Grant-in-Aid (08283101:“Genome Science”) for Scientific Research on Priority Areas from The Ministry of Education, Science, Sports and Culture of Japan, and in part by “Genome Informatics” project of Agency of Industrial Science and Technology, The Ministry of International Trade and Industry of Japan. The authors thank members of Genetic Knowledge Laboratory of Japan Advanced Institute of Science and Technology and members of Genome Informatics Group of Electrotechnical Laboratories for the support and the discussions.

References

[1] Asai, K., Itou, K., and Ueno, Y., Recognition of human genes by stochastic parsing, *Pacific Symposium on Biocomputing '98*, 228–239, 1998.

[2] Asai, K., Ueno, Y., Itou, K., and Yada, T., Automatic gene recognition without using training data, *Genome Informatics 1997*, 15–24, 1997.

[3] Krogh, A., Mian, I.S., and Haussler, D., A hidden Markov model that finds genes in *E. coli* DNA, *Nucleic Acids Res.*, 22(22):4768–4778, 1994.

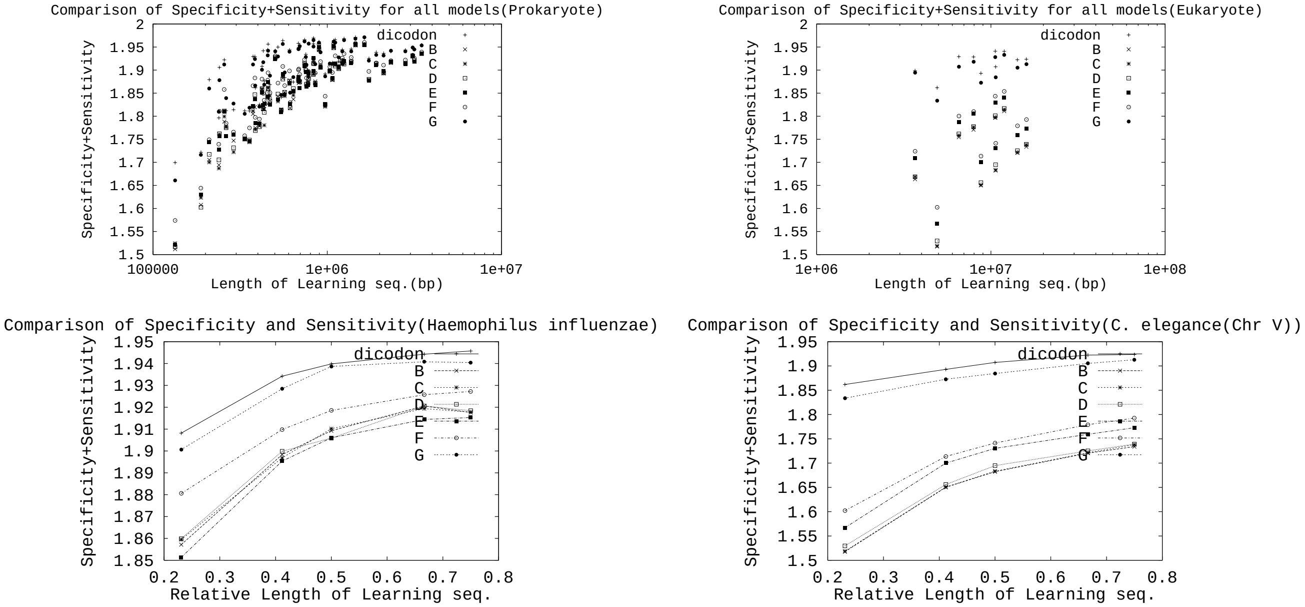


Figure 4: Comparison of Specificity+Sensitivity for seven models over 13 microbial genomes(upper left) and for 2 eukaryotic(*C. elegance*) genomes(upper right). A typical result of a microbial genome(*H. influenzae*) is shown bottom left. And a result of a eukaryotic genome(*C. elegance(chr V)*) is shown bottom right.

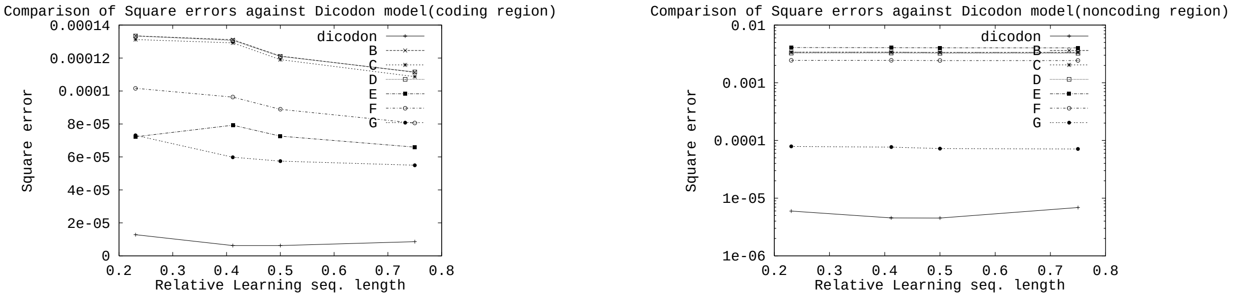


Figure 5: The left figure shows square errors of the coding potentials of testing sequences(left: coding regions, right:noncoding regions) for each models against the coding potential of dicodon model that was trained with testing sequences. The square errors are average values over 13 microbial and 2 eukaryotic genomes.

[4] Yada, T. and Hirosawa, M., Gene recognition in cyanobacterium genomic sequence data using the hidden Markov model, *Proc. ISMB*, AAAI Press, 4:252-260, 1996.

[5] Dong, S. and Searls, D.B., Gene structure prediction by linguistic methods, *Genomics*, 23:540–0551, 1994.

[6] Fickett, J.W. and Tung, C.S., Assessment of protein coding measures, *Neucleic Acid Research*, 20(24):6441–6450, 1992.

[7] Audic, S. and Claverie, J.M., Self-identification of protein-coding regions in microbial genomes, *Proc. Natl. Acad. Sci. USA*, 95(17):10026–10031, 1998.

[8] Klenk, H.P., *et al.*, The complete genome sequence of the hyperthermophilic, sulphate-reducing archaeon *Archaeoglobus fulgidus*, *Nature*, 390(6658):364–370, 1997.

[9] Deckert, G., *et al.*, The complete genome of the hyperthermophilic bacterium *Aquifex aeolicus*, *Nature*, 392(6674):353–358, 1998.

[10] Fraser, C.M., *et al.*, Genomic sequence of a Lyme disease spirochaete, *Borrelia burgdorferi*, *Nature*, 390(6660):580–586, 1997.

[11] Kunst, F., *et al.*, The complete genome sequence of the gram-positive bacterium *Bacillus subtilis*, *Nature*, 390(6657):249–256, 1997.

[12] Stephens, R.S., *et al.*, Genome sequence of an obligate intracellular pathogen of humans: *Chlamydia trachomatis*, *Science*, 282(5389):754–759, 1998.

[13] Blattner, F.R., *et al.*, The complete genome sequence of *Escherichia coli* K-12, *Science*, 277(5331):1453–1474, 1997.

[14] Fleischmann, R.D., *et al.*, Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd, *Science*, 269(5223):496-512, 1995.

[15] Tomb, J.F., *et al.*, The complete genome sequence of the gastric pathogen *Helicobacter pylori*, *Nature*, 388(6642):539–547, 1997.

[16] Fraser, C.M., *et al.*, The minimal gene complement of *Mycoplasma genitalium*, *Science*, 270(5235):397–403, 1995.

[17] Bult, C.J., *et al.*, Complete genome sequence of the methanogenic archaeon, *Methanococcus jannaschii*, *Science*, 273(5278):1058–1073, 1996.

[18] Himmelreich, R., Hilbert, H., Plagens, H., Pirkl, E., Li, B.C., and Herrmann, R., Complete sequence analysis of the genome of the bacterium *Mycoplasma pneumoniae*, *Nucleic Acids Res.*, 24(22):4420–4449, 1996.

[19] Smith, D.R., *et al.*, Complete genome sequence of *Methanobacterium thermoautotrophicum* deltaH: functional analysis and comparative genomics, *J. Bacteriol.*, 179(22):7135-7155, 1997.

[20] Cole, S.T., *et al.*, Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence, *Nature*, 393(6685):537–544, 1998.

[21] Kawarabayasi, Y., *et al.*, Complete sequence and gene organization of the genome of a hyperthermophilic archaebacterium, *Pyrococcus horikoshii* OT3, *DNA Res.*, 5(2):55–76, 1998.

[22] Kaneko, T., *et al.*, Sequence analysis of the genome of the unicellular cyanobacterium *Synechocystis* sp. strain PCC6803. II. Sequence determination of the entire genome and assignment of potential protein-coding regions, *DNA Res.*, 3(3):109–136, 1996.

[23] Fraser, C.M., *et al.*, Complete genome sequence of *Treponema pallidum*, the syphilis spirochete, *Science*, 281(5375):375–388, 1998.

[24] Andersson, S.G., *et al.*, The genome sequence of *Rickettsia prowazekii* and the origin of mitochondria, *Nature*, 396(6707):133–140, 1998.

[25] The *C. elegans* Sequencing Consortium, Genome sequence of the nematode *C. elegans*: a platform for investigating biology, *Science*, 282(5396):2012–2018, 1998.