A Study on Dicodon-oriented Gene Finding using Self-Identification Learning

By Juang Kim

A thesis submitted to
School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Knowledge Science
Graduate Program in Information Science

Written under the direction of
Professor Akihiko Konagaya

February 15, 2000

Abstract

An evaluation of a probabilistic measure of the gene structural attributes for computational prediction of protein coding region in prokaryotic genomic sequence data is presented in this paper. The measure is known as dicodon usage measure and has been known as the best measure to discriminate protein coding/non-coding regions but its biological semantics have been remaining unknown. Besides, there has been no objective and quantitative investigation carried out. Our preliminary analysis for the dicodon usage measure using "dicodon-oriented Hidden Markov Model", which gives a simple but powerful probabilistic description of gene structure with relatively small size of parameters, with "Self-identification learning method", which offers novel learning scheme that does not require prepared training data, indicated that the dicodon usage measure is redundant as a gene finding measure. This result induces a hypothesis; there is another measure that has smaller size of parameters and performs as good as the dicodon usage measure. Such measure would make the gene finding more self-identification learning friendly. Hence it will facilitate complete automation of gene finding that is desperately demanded by next generation of genome sequencing projects.

In order to evaluate the redundancy and accuracy of dicodon usage measure, we took "divide and conquer" approach. Based on the compositions of the dicodon usage measure such as codon usage, diamino-acid, and C+G content, we used six different model to emulate the dicodon usage measure by the compositions with smaller parameter size than the dicodon usage measure. We evaluated the six models and the dicodon usage measure in aspect of sensitivity, specificity, and approximation error. Our evaluation result shows that the dicodon model outperforms the six emulators 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. Our hypothesis is not fully evaluated but this negative result has reasonable impact on "common sense" of Bioinformatics.

Contents

1 Introduction **5**
- 1.1 Background . . . 5
- 1.2 Overview of Gene Finding . . . 6
- 1.3 The Dicodon Usage Measure . . . 8
- 1.4 Hidden Markov Model . . . 8
- 1.5 Self-identification Learning Method . . . 13

2 Evaluation of Self-identification Learning **15**
- 2.1 Method . . . 15
  - 2.1.1 System Overview . . . 15
  - 2.1.2 Dicodon Oriented HMM . . . 17
  - 2.1.3 Self-identification Learning . . . 17
- 2.2 Result and Discussion . . . 20
- 2.3 Conclusion for the preliminary examination . . . 21

3 Evaluation of Dicodon Usage Measure **31**
- 3.1 Models . . . 31
- 3.2 Evaluation of models . . . 34
  - 3.2.1 Approximation error . . . 35
  - 3.2.2 Evaluation of Learning/Testing . . . 37
- 3.3 Result . . . 38
- 3.4 Discussion . . . 46

4 Conclusion **47**

5 Appendix **48**

List of Figures

1.1 The elongation phase of protein synthesis on a ribosome 9
1.2 Derivability of coding measures 12
1.3 Comparison between generic learning scheme and self-identification learning 13

2.1 Overview of the gene finding examination 16
2.2 Dicodon oriented HMM diagram 18
2.3 Measures of prediction accuracy at the nucleotide level 20
2.4 Results of gene finding (a) 23
2.5 Results of gene finding (b) 24
2.6 Results of gene finding (c) 25
2.7 The number of HMM parameters (a) 26
2.8 The number of HMM parameters (b) 27
2.9 The number of HMM parameters (c) 28
2.10 Detailed prediction results 29

3.1 Attributes in a hexamer 32
3.2 Measuring distance of the overlap 36
3.3 Actual histogram of coding/non-coding potential for *E.coli* 37
3.4 Sensitivity+Specificity and training data size (a) 40
3.5 Sensitivity+Specificity and training data size (b) 41
3.6 Sensitivity+Specificity and training data size (c) 42
3.7 Sensitivity+Specificity and training data size (d) 43
3.8 Sensitivity+Specificity and training data size (e) 44
3.9 Square error distance 45

List of Tables

1.1 Codon Usage differences between coding and non-coding region in *E.coli* (a) 10
1.2 Codon Usage differences between coding and non-coding region in *E.coli* (b). 11
1.3 Percentage accuracy (average of specificity and sensitivity) of the coding measures in predicting phase-specific coding (*excerpt from* [11] *Table 3*). . 11

2.1 Example output format of H2Vite 19
2.2 Recognition result for 17 microbial genomic sequence data. CC stands for correlation coefficient. R stands for recognition result. And R^* shows another self-identification gene finding result by Audic and Claverie [3]. . 22

3.1 Maximum sensitivity+specificity of every model for 14 microbial genomic sequence data and 14 eukaryotic genomic sequence data. Mean sensitivity+specificity is shown in bottom of the table. 39

5.1 17 microbial genomic sequence data 48

Chapter 1

Introduction

1.1 Background

It is well known that every biological life, including tiny viruses, has one or more long chain of molecule known as Deoxyribose Neucleic Acid(DNA) and the DNA carries massively rich information that responsible for forming, developing, and reproducing life forms no matter the life is a noble saint or a tiny bacteria in a mud.

Since the initial break-through of discovering double-helical structure by Watson and Click in 1953, molecular biology has been revealing complicated mechanisms of biological life and harvested a hand full of attainments.

Although the molecular biology has been such successful, vague but huge mountains of biological enigma is lying ahead of researchers in this field. Besides, the enigma is too hard to solve without computational aids. Today, newly formed inter-disciplinary research fields has been arisen and valued higher and higher than ever. Bioinformatics is one of such newcomers. There are several ways to define this new research field.

One of them describes it as:

> Bioinformatics is an integration of mathematical, statistical and computer methods to analyze biological, biochemical and biophysical data. (excerpt from home page of School of Biology, Georgia Institute of Technology)

Some of the goals of the Bioinformatics include:

(i) clarification of biological functions of genes

(ii) prediction of protein structure (secondary/tertiary)

(iii) detection of regulatory signals (promoters, enhancers, origins of replication, etc.) in genomic DNA sequences

(iv) inferring evolutionary history from comparison of homologous gene or protein sequences (or genomes).

In this paper we focused on one of the most significant open problem in this field, that is prediction of precise protein coding region structure in genomic sequences (commonly known as "gene finding", "gene prediction", or "gene identification") of every species ranging from prokaryota(mainly bacteria) and eukaryota(such as yeast, plant, worm, and human). Because, the clarification of biological functions of genes has been one of the primary goals of the Bioinformatics and desperately demanded especially as the next generation research topic since on-going global genome projects which will be completed in a short time. The *gene finding* is very important for the functional clarification of genes and complete automation of the gene finding is also demanded because a large quantity of genomic sequence data is being piled up on a host of databases and waiting to be analyzed. Such incessant increase of demands makes the gene finding more crucial.

Main stream of the gene finding has been targeting higher eukaryotic (especially human) genomic sequences which have more complicated genomic structure thus more challenging to predict than prokaryotic genome. The gene finding with higher eukaryotic genomic sequences requires precise definition of the sequence dependence of molecular biological mechanisms such as:

(i) the basic biochemical processes of DNA to RNA transcription

(ii) RNA translation and splicing(*i.e.* exons and introns)

(iii) knowledge about the sequence properties of known genes

Although the above mechanisms have been under intensive investigation, heaps of knowledge are still waiting to be discovered. So the gene finding has been a computational and analytical method to full fill the void of knowledge on these mechanisms. While the prediction problem is hard and challenging, it increases its importance regarding recent shift in emphasis of the Human Genome Project from reading every nucleotide of human genomic sequence to finding functional role of every genes. In order to get clues to find the functions of genes, we need to find exactly where those genes reside in a lengthy sequence of nucleotides with aids of computational methods.

This paper emphasizes precision modeling of genomic sequences; especially discrimination of protein coding/non-coding regions based on the dicodon usage measure which has been known as one of the most precise among protein coding measures. Although Fickett and Tung gave an objective and quantitative evidence to the superiority of the dicodon usage measure [11], there has been no sufficient investigation taken for the dicodon usage measure to clarify the biological background to explain why the dicodon works such well. This paper aims to investigate and clarify the biological semantics of the dicodon usage measure.

1.2 Overview of Gene Finding

Gene Finding(*a.k.a.* gene prediction or gene identification) is a computational method to find protein coding regions out of genomic sequences and has been studied extensively for

nearly a couple of decades. The history of gene finding has been a history of finding the best model that distinguish coding and non-coding regions in a genomic sequence. The models are designed to detect molecular biological attributes and signals which discriminate coding and non-coding regions.

Since initial break-through of de-ciphering genetic code by Nirenberg and Matthaei in 1961, molecular biology has been clarified a hand full of differences between coding and non-coding regions to be used for fine description of protein coding regions. Some of the well known generic attributes are codon usage bias and C+G content. Both of them are explained as probabilistic differences of nucleotide sequence between coding and non-coding regions which are consequences of the evolutionary mutational pressure [22]. Naturally, the pressure varies in the coding regions, which are relatively conservative to the mutation and in the non-coding regions, which are neutral to the mutation. It was very straight forward to use the attributes for defining probabilistic models to recognize coding regions from a genomic sequence. Early studies on gene finding by Shepherd [17], Fickett [10], and Staden & McLachlan [18] showed that statistical measures related to biases in amino-acid and codon usage could be used to approximately identify protein coding regions in genomic sequences [6].

Since the early initiation of stochastic and computational approach, a bunch of gene expression models have been developed and contributed to this research field. Summaries and comprehensive evaluation for gene finding have been proposed by many researchers. The recent summary of the gene identification problem was contributed by Fickett [12] and an evaluation of gene structure prediction programs was offered by Burset and Guigo [7]. Gene Finding approaches are roughly divided into two categories:

- Sequence similarity search
- Stochastic models based on statistical regularities in coding region; *coding measures*

Sequence similarity search is one of the oldest methods of gene finding, based on sequence conservation due to functional constraint, and is to search for regions of similarity between the sequence under study (or its conceptual translation) and the sequences of known genes (or their protein products). A clear advantage to searching for genes by similarity is that, if a significant similarity is found, it is likely to yield clues as to the function, as well as the existence, of the new gene. In addition, if the search is carried out at the amino-acid, rather than the nucleotide, level, the additional advantage may be had of lowered sensitivity to the "noise" of neutral mutations. The obvious disadvantage of this method is that when no homologue to the new gene are to be found in the databases, similarity search will yield little or no useful information. More detailed review can be found in [12].

At the core of most gene recognition algorithms are one or more *coding measures*. They are functions which calculate, for any window of sequence, a number or vector that measures attributes correlated with protein coding function. Aggregate properties of such function values on coding regions thus from templates for exons in general. Common examples of coding measures include the codon usage vector, the base composition vector, and some type of Fourier transform of the sequence.

1.3 The Dicodon Usage Measure

This paper focuses on the statistical regularities in coding regions, where the dicodon usage measure should be discussed. Fickett and Tung evaluated every coding measures known to the public and showed that the dicodon usage measure is one of the best measures among others [11]. Every protein coding region is translated from nucleotides to amino-acids, in a triplet basis, under a rule of genetic code. The triplet is called *Codon*. Figure 1.1 shows how the translation occur in the molecular world. It is well known that the occurrence of codon has peculiar bias which means not every codon is used evenly in a genomic sequence and thus the codon can be used as a measure of coding region. Such unevenness is called *Codon Usage* and denoted as a conditional probability $p(c|A(c))$ where c for a codon and $A(c)$ for an amino-acid corresponds to the codon c. Table 1.1 to 1.2 show differences between coding and non-coding region for *E. coli*.

Although the codon usage measure offers simple description for coding regions, it just produces lower scores (specificity and sensitivity) than other measures such as dicodon usage measure [11] (*see also* Table 1.3). The measures that perform better than the codon usage measure belong to hexamer-n measure. The hexamer-n measure (for $n = 0, 1, 2$) counts all hexamers (*i.e.* six nucleotide) offset by n from the starting base. Dicodon usage measure is identical to hexamer-0 measure. Hexamer-1 and 2 measures perform slightly worse than dicodon usage measure. Dicodon usage measure can be denoted as a conditional probability $p(c_{i+1}|c_i)$ where c_i for a codon and c_{i+1} for its next codon.

The simple calculation shows that the codon usage measure has 1,220 parameters for 61 codons and 20 amino-acids, and the dicodon usage measure has 3,721 parameters. Notice that the dicodon usage measure performs slightly better than the codon usage measure that has only one-third of the parameters. This simple fact implies that the dicodon usage measure is more redundant than the gene finding actually requires. Besides, our preliminary examination (explained later) indicated the same conclusion. Fickett stated that the dicodon usage or hexamer-n measure contains all of other known measures such as codon usage, diamino-acid, and dinucleotide bias [11](*see also* Figure 1.2). According to the redundancy indicated above, it is reasonable that not all of these measures does not need to be included by the dicodon usage measure. This thesis focuses on this very point and tries to clarify which measure is the most important and which is the least important.

1.4 Hidden Markov Model

Wide variety of gene identification algorithms have been and will be developed. All integrated gene identification programs make use of the high level syntax of genes resulting from our basic understanding of transcription, splicing, and translation [12]. So it is very straight forward to assume that a computational linguistic method can be applied to the gene identification. Actually, some of the algorithms took a computational linguistic approach to the gene finding. Searls suggested that a linguistic approach to the analysis of features in DNA sequences could be beneficial [16]. This approach is first applied to the

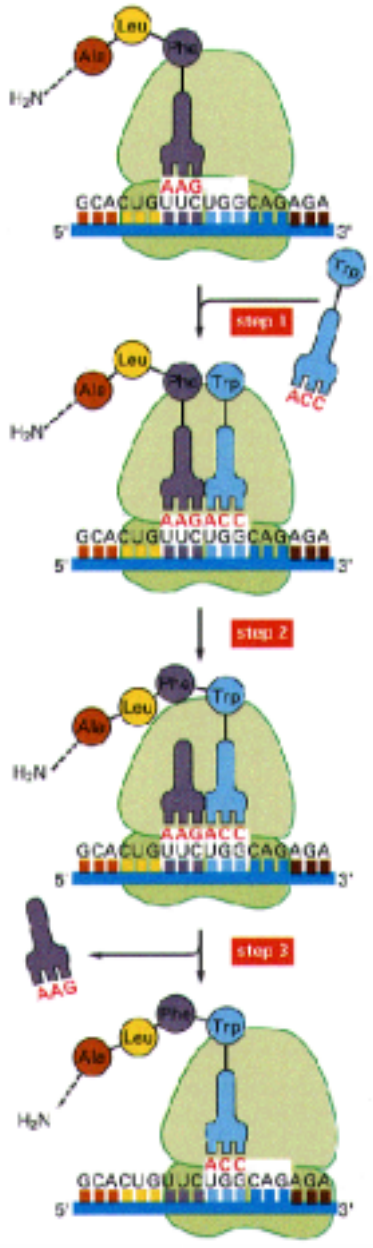

Figure 1.1: The elongation phase of protein synthesis on a ribosome. The three-step cycle shown is repeated over and over during the synthesis of a protein chain. An aminoacyl-tRNA molecule binds to the A-site on the ribosome in step 1, a new peptide bond is formed in step 2, and the ribosome moves a distance of three nucleotides along the mRNA chain in step 3, ejecting an old tRNA molecule and "resetting" the ribosome so that the next aminoacyl-tRNA molecule can bind.

Table 1.1: Codon Usage differences between coding (CD) and non-coding (NC) region in *E.coli* (a). Notice that codon usages in coding regions are heavily biased or apparently different from the non-coding regions.

Amino Acid	**CODON**	**USAGE(CD)**	**USAGE(NC)**
Ala	GCA	0.213	0.288
	GCC	0.270	0.241
	GCG	0.356	0.258
	GCT	0.161	0.213
Arg	AGA	0.039	0.177
	AGG	0.023	0.176
	CGA	0.065	0.132
	CGC	0.398	0.190
	CGG	0.098	0.168
	CGT	0.378	0.157
Asn	AAC	0.550	0.397
	AAT	0.450	0.603
Asp	GAC	0.372	0.358
	GAT	0.628	0.642
Cys	TGC	0.556	0.505
	TGT	0.444	0.495
Gln	CAA	0.347	0.518
	CAG	0.653	0.482
	GAA	0.689	0.622
	GAG	0.311	0.378
Gly	GGA	0.109	0.252
	GGC	0.403	0.284
	GGG	0.151	0.222
	GGT	0.337	0.243
His	CAC	0.429	0.406
	CAT	0.571	0.594
Ile	ATA	0.073	0.325
	ATC	0.420	0.256
	ATT	0.507	0.419
Leu	CTA	0.037	0.093
	CTC	0.104	0.116
	CTG	0.496	0.169
	CTT	0.104	0.168
	TTA	0.131	0.264
	TTG	0.128	0.191
Lys	AAA	0.765	0.680
	AAG	0.235	0.320
Met	ATG	1.000	1.000

Table 1.2: Codon Usage differences between coding and non-coding region in *E.coli* (b) (*continued*).

Amino Acid	**CODON**	**USAGE(CD)**	**USAGE(NC)**
Phe	TTC	0.426	0.328
	TTT	0.574	0.672
Pro	CCA	0.191	0.241
	CCC	0.124	0.224
	CCG	0.525	0.258
	CCT	0.159	0.277
Ser	AGC	0.277	0.161
	AGT	0.151	0.154
	TCA	0.124	0.239
	TCC	0.149	0.151
	TCG	0.154	0.125
	TCT	0.146	0.171
Thr	ACA	0.132	0.308
	ACC	0.434	0.219
	ACG	0.268	0.239
	ACT	0.166	0.234
Trp	TGG	1.000	1.000
Tyr	TAC	0.431	0.372
	TAT	0.569	0.628
Val	GTA	0.154	0.238
	GTC	0.216	0.190
	GTG	0.371	0.233
	GTT	0.259	0.340

Table 1.3: Percentage accuracy (average of specificity and sensitivity) of the coding measures in predicting phase-specific coding (*excerpt from* [11] *Table 3*).

Measure	Human 54 Penrose	Human 108 Penrose	Human 162 Penrose	*E.coli* 54 Penrose	Human 54 Classical
Dicodon Usage (Hexamer-0)	80.7	84.3	85.4	88.7	–
Hexamer-2	79.5	82.8	84.2	87.2	–
Hexamer-1	78.6	82.0	83.3	87.1	–
Codon Usage	78.0	81.0	82.1	86.9	81.7
Diamino-acid Usage	77.2	84.9	87.7	84.2	–
Amino-Acid Usage	75.3	81.1	83.6	83.3	76.2

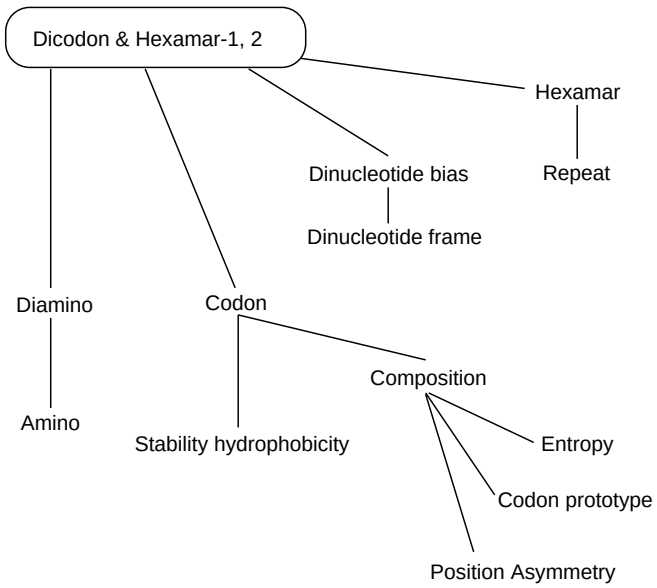

Figure 1.2: Derivability of coding measures: Each measure is derivable from any measure above it and connected to it by a line (*excerpt from* [12] *Figure 1*).

identification of protein coding region as GenLang by Dong & Searls [9], where a formal, definite clause grammar of genes is described.

Hidden Markov Model (HMM) [15] has been widely used for computational natural language processing. Application of HMM to genomic sequence analysis was first introduced by Churchill [8]. HMM is advanced model of Markov model to deal with problems that is unable to handle with Markov model. HMM can provide several advantages such as flexible description of *signal patterns*, virtually direct translation of genomic attributes to HMM network, and explicit definition of HMM parameters. The components and the rules of the *DNA language* are non-deterministic, it is necessary to combine the statistics and the linguistics for the *parsing* of DNA. That is why HMM are becoming widely used for gene recognition ([14, 38, 20, 19]). A particular advantage of the HMM approach of Krogh *et al* [14] is that it naturally provides a joint probability distribution over sequences and parses of those sequences. The HMM thus provides a very natural vehicle for considering the possibility of introducing a sequence correction to get a more probable parse [12].

In this paper, we used an HMM with dicodon usage measure to build simple probabilistic description model to recognize protein coding regions for prokaryotic genome. The HMM provides simple and intuitive modeling that facilitates analysis of gene finding result thus the HMM is chosen for our preliminary gene finding examination as a suitable test-bed.

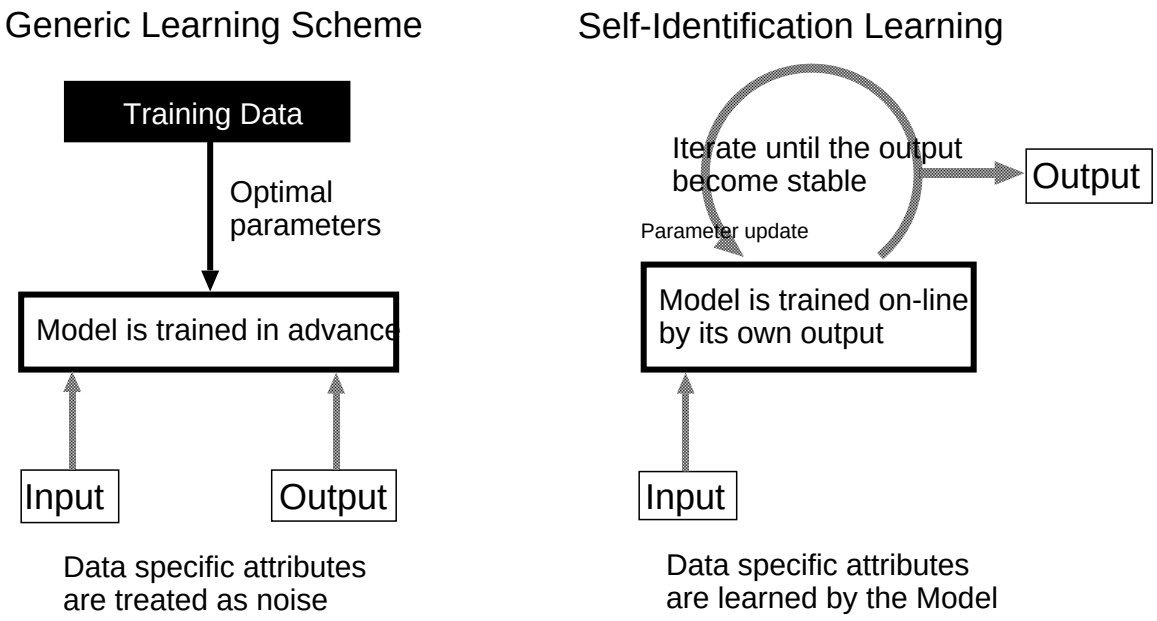

Figure 1.3: Side-by-side comparison between generic learning scheme and self-identification learning: Notice that the generic learning scheme needs training data which is based on previously acquired data although the self-identification learning does not need such data. Thus the self-identification learning reflects data specific attributes to its output while such attributes are treated as noise in the generic learning scheme.

1.5 Self-identification Learning Method

Self-identification learning [2, 3] is relatively new approach that does not require *training sequence* while most other algorithms require the *training sequence*. Conventional learning scheme is trained by *training sequence* in order to obtain optimum set of parameters. However, such strategy becomes totally impossible when there is no datum available for the training and, possibly in many cases, such circumstances can be arisen especially for practical gene finding where we can not expect to have *correct data* in advance. For example, practical gene finding often requires gene prediction against totally new species —which means there is no previously acquired similar or phylogenicaly related genomic sequence data— therefore no training data can be effective if not offered. Besides, the self-identification learning can directly reflect data specific attributes to its output although the generic learning scheme tends to treat such attributes as noise. This feature of the self-identification learning is very important especially for gene finding that is applied to new, thus previously unknown, species. Because such attributes are essential for gene finding against the new species and the generic learning scheme with training data, which obviously do not include *new* data, usually fails identify coding regions in such new species. The self-identification learning can obtain optimal parameters without training data in

following way(*see also* Figure 1.3):

- 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

Efficiency of the self-identification largely depends, by its nature, 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 of the model and the accuracy.

In this paper, we fed short fragments of microbial genomic sequence data to our gene finding system in order to evaluate the robustness of the self-identification learning against short training data.

Chapter 2

Evaluation of Self-identification Learning

In this paper, two examination/analysis were performed. The first is computational gene finding using dicodon-oriented HMM with self-identification learning and the second is evaluation of dicodon usage measure. The former provides reason of the latter evaluation that is the reason to ask *what make dicodon usage measure such redundant.* Firstly, evaluation of self-identification learning is provided in this section.

2.1 Method

We used a dicodon oriented HMM gene finding system with self-identification learning [2] as a test-bed for the evaluation. Two objectives are set and they are:

- to purely evaluate gene identification accuracy for our system
- to evaluate robustness of self-identification learning against short training data

2.1.1 System Overview

We built a gene finding system that is incorporated with HTK (HMM Tool Kit) [23] which is a commercial(*Entropic Inc.*) software toolkit for building continuous density HMM based speech recognizers. Although the HTK is designed for dealing with continuous density distributions, the differences are minor between the continuous and discrete probability distributions. Therefore HTK offers seamless platform to the gene finding. HTK uses Baum-Welch algorithm(*a.k.a.* Expectation-Maximization algorithm) [4] for learning its parameters, and uses Viterbi algorithm [13] for coding region recognition. Figure 2.1 provides at-a-glance overview of our gene finding examination. Actually, by nature of our evaluation method, we used only H2Vite which is a part of the toolkit and provides coding region recognition alone with Viterbi algorithm. For the examination, we used 17 microbial complete genomic sequence data [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40] available from GenBank [5](*see* Appendix). We evaluated

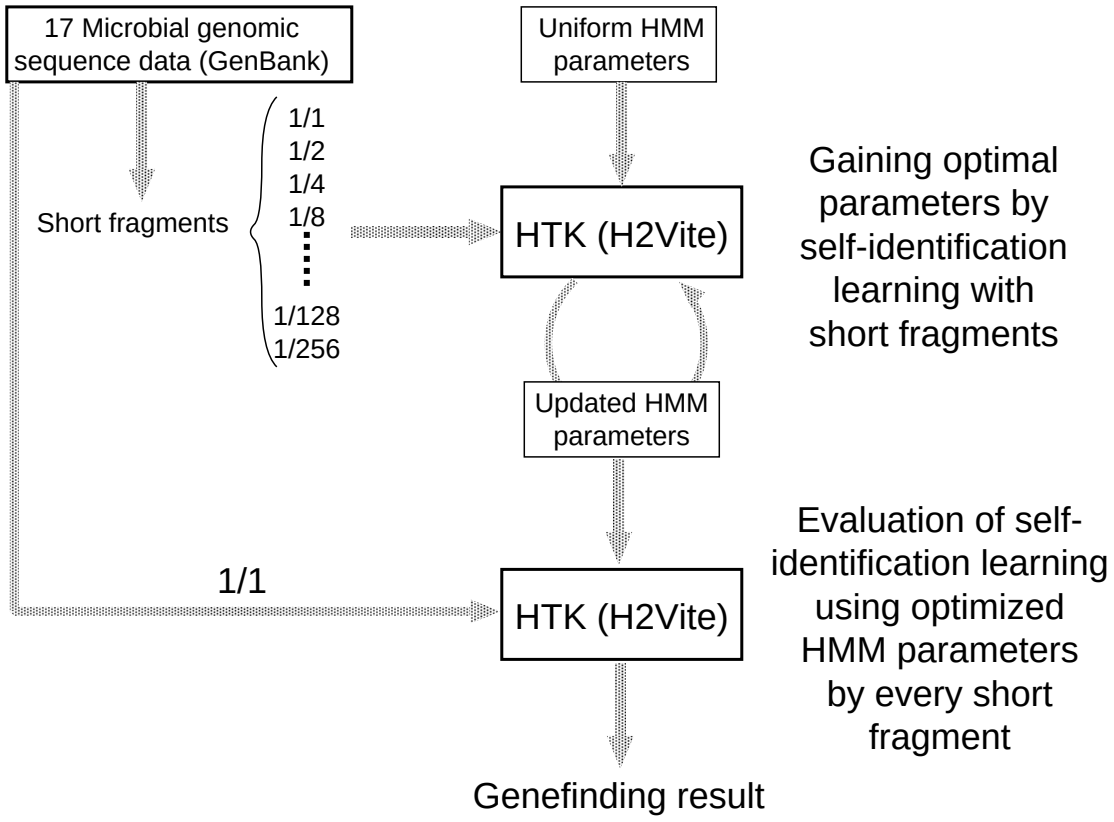

Figure 2.1: At-a-glance overview of the gene finding examination to evaluate self-identification learning.

the data size dependency of the self-identification learning with short fragments of the sequence data such as 1/2, 1/4, 1/8, ..., and 1/256 of complete sequence.

2.1.2 Dicodon Oriented HMM

Figure 2.2 shows overview of the dicodon oriented HMM network which uses simple grammar to describe protein coding regions in a microbial genomic sequence because we need to keep the system as simple as possible in order to facilitate analysis focused on the dicodon usage measure. In microbial genome, and also in several eukaryotic no-internal-exon genome such as *yeast*, every protein coding region can be described, for 5' to 3' strand, as arbitrary iteration of codons that is sandwiched by start(5') and stop(3') codons, and for 3' to 5' (complementary) strand, as arbitrary iteration of complementary codons that is sandwiched by complementary start(3') and stop(5') codons. Most of coding regions are connected by a *spacer* i.e. non-coding region which actually is arbitrary, but definitely shorter than coding regions, length of nucleotides. However, the genome structure is not such simple because:

- non-coding regions are occasionally not exist between coding regions
- coding regions are occasionally overlapped each other

Functions to handle these exceptions are not implemented in our system because such implementation has nothing to do with the evaluation of the dicodon usage measure and we just wanted to keep our system simple.

In figure 2.2, each rectangle corresponds to a certain structural item that forms a protein coding region structure. The non-code state corresponds to a non-coding region and is a single state and emits four output; A, C, G, and T. The start codon state corresponds to a start codon region and is a small HMM that has 11 single output states and 3 transition parameters inside when there are three possible start codons are expected[1]. The stop codon is conceptually identical to the start codon state. The Dicodon state corresponds to a coding region sandwiched by start and stop codons and is an HMM that has 185 single output states and 3,782 transition parameters inside. As the total, the HMM has 7,568 transition parameters.

2.1.3 Self-identification Learning

The initial parameters of the dicodon oriented HMM have uniform value. For example, non-code state has uniform distribution for every output probability i.e. 1/4 for A, C, G, and T. The self-identification learning begins the gene finding with this pre-learning condition. H2Vite outputs a file denoting the prediction where in the sequence belong to a state with likelihood calculated by Viterbi algorithm (*see* Table 2.1). The output from H2Vite is parsed by a *Perl* script and statistical data is accumulated to update HMM parameters i.e. emission parameters for non-code state and transition parameters

[1]When there are less than three possible start codons expected, the number of the HMM states are reduced to less than 11

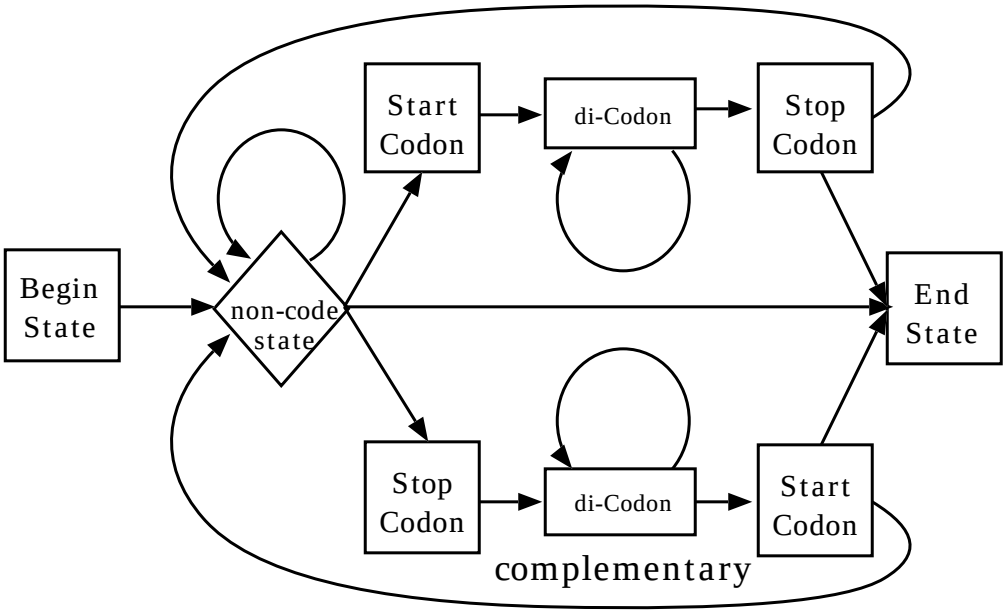

Figure 2.2: A network diagram of a dicodon oriented HMM. "Start Codon" state emits possible three start codons(ATG, TTG, GTG). "Stop Codon" state emits possible three stop codons(TAA, TAG, TGA). "di-Codon" state emits possible 61 dicodons iteratively.

Table 2.1: Example output format of H2Vite. The first column and second column show a position in genomic sequence data. The third column shows a state where the HMM predicts. The last column shows the log likelihood of corresponding prediction.

Begin	**End**	**State**	**Log Likelihood**
0	655	noncode	-1.392755
655	658	start	-0.116298
658	844	codon	-1.339531
844	847	stop	-0.223741
847	977	noncode	-1.440781
977	980	start	-0.116309
980	1049	codon	-1.343838
1049	1052	stop	-0.223737
1052	1109	noncode	-1.520269
1109	1112	start	-0.116287
1112	1184	codon	-1.324150
1184	1187	stop	-0.223724
1187	3710	noncode	-1.385260
3710	3713	start	-0.116362
3713	3791	codon	-1.359336
3791	3794	stop	-0.223705
3794	4289	noncode	-1.396874
4289	4292	start	-0.116303
4292	4781	codon	-1.352109
4781	4784	stop	-0.223693
4784	5050	noncode	-1.408878
5050	5053	start	-0.116273
5053	6808	codon	-1.229615

for start/stop codon dicodon state. During the statistical data accumulation, the system rejects apparently false answer that do not comply coding region grammar implemented in HMM network. Hence the HMM is trained by correct or possibly correct prediction results although it is never fed training data in advance. Then H2Vite try recognition again, but this time, with updated HMM parameters. The above procedures are iterated until the recognition accuracy become maximum.

In order to evaluate the robustness of the self-identification learning, we used short fragments of microbial genome sequence data such as 1/2, 1/4, 1/8, and such of whole genomic sequence data to train the HMM as described above. After the HMM is trained, the model starts to find protein coding regions from a whole genomic sequence data and we can see how the HMM can predict coding regions accurately with a short fragment of training data. Therefore we can evaluate data length dependency of self-identification learning.

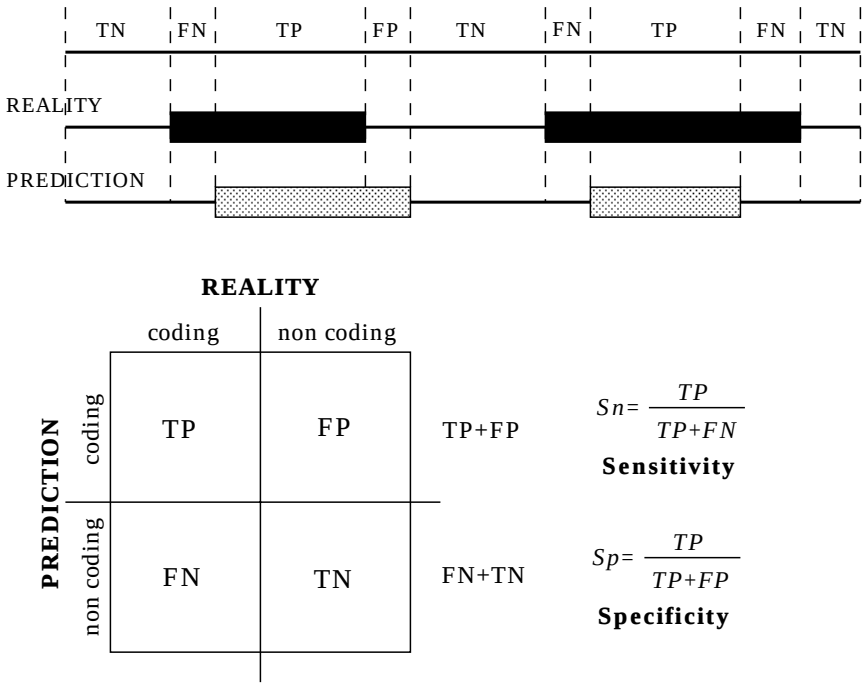

Figure 2.3: Measures of prediction accuracy at the nucleotide level (*excerpt from* [7] *Figure 1*).

2.2 Result and Discussion

The prediction accuracy is evaluated by counting TP(true positive): number of bases predicted as inside of coding regions correctly, TN(true negative): number of bases predicted as outside of coding regions correctly, FP(false positive): number of bases predicted as inside of coding regions incorrectly, and FN(false negative): number of bases predicted as outside of coding regions incorrectly. There are common measures to evaluate prediction accuracy at the nucleotide level [7] (*see also* Figure 2.3):

- Sensitivity: $Sn = \frac{TP}{TP+FN}$
- Specificity: $Sp = \frac{TP}{TP+FP}$
- Correlation Coefficient: $CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP+FN) \times (TN+FP) \times (TP+FP) \times (TN+FN)}}$

Additionally, simple nucleotide level prediction accuracy is given by $R = \frac{TP+TN}{TP+TN+FP+FN}$.

Table 2.2 shows the highest nucleotide level prediction accuracy R, sensitivity Sn, specificity Sp, and correlation coefficient CC for 17 microbial genomic sequence data. As a comparison for the score we got, the table includes scores obtained by another similar research by Audic and Claverie [3]. Please note that the objectives of this paper do not include getting high prediction accuracy and our system is far simple than that of

the Audic and Claveries', however our prediction accuracy exceeds their score over most species.

Figure 2.4 to 2.6 show dependency of R, Sn, Sp, and CC on training data size for each microbial genomic sequence data. Sn stays constantly high level over all sequence data because almost all bacteria contains very small number of non-coding regions comparing to coding regions. Hence FN is much smaller than TP. Sp, R, and CC draw similar proposition because of the same reason. There are apparent degradation of prediction accuracy for short training data size. However, the prediction accuracy stays high until the training data size is lowered to around 1/16 of whole data size.

Figure 2.7 to 2.9 show the number of learned HMM parameters versus training data size for each microbial genomic sequence data. The results shown in the figure vary widely for each sequence data because there are many differences caused by evolutionary diversity. Some of the sequence data require very small amount of HMM parameters, far below from the parameter size of codon usage measure i.e. 1,220, to identify protein coding regions. On the other hand, some of the sequence data require more than 1,220 parameters to be learned to attain good prediction accuracy. Besides the maximum number of HMM parameters often results in lower prediction accuracy i.e. *over fitting*. Typical proportion is shown in *Archaeoglobus fulgidus, Borrelia burgdorferi, Chlamydia trachomatis, Escherichia coli, Haemophilus infulenzae, Methanococcus jannaschii, Mycobacterium tuberculosis, Pyrococcus horikoshii, Rickettsia prowazekii, and Treponema pallidum*.

Therefore it is obvious that not all of HMM parameters are needed to identify protein coding regions with reasonable accuracy. Consequently, this evidence leads to a conclusion that the dicodon usage measure is redundant for the gene finding.

Figure 2.10 is a snapshot showing details of prediction result in a coding region basis. The numbers on a solid line represent base position in a genomic sequence data. The stripes right bellow of the solid line is showing correct coding regions; greens for 5' to 3' strand and reds for complementary (3' to 5') strands. The blue stripes and orange stripes are representing prediction result with short fragments of training data (top 1/1, bottom 1/256).

Notice that the predicted coding region is getting shorter, hence yielding more error, than the real coding region.

2.3 Conclusion for the preliminary examination

Our evaluation shows that the dicodon usage measure is redundant for the gene finding. The result implies that we can use a measure that has smaller size of parameters for gene finding in reasonable accuracy. Fickett and Tung indicated that the dicodon usage measure performs better than codon usage measure [11]. Their evaluation shows that prediction accuracy by the dicodon usage measure exceeds that by the codon usage measure but merely showing slightly better accuracy 1.3 considering the difference of parameter size among them. Although we found that the dicodon usage measure is too large in its parameter size and codon usage measure does not perform as good as the dicodon usage measure, the intermediate measure, which performs as accurate as the dicodon and has

Table 2.2: Recognition result for 17 microbial genomic sequence data. CC stands for correlation coefficient. R stands for recognition result. And R^* shows another self-identification gene finding result by Audic and Claverie [3].

Species	**Sensitivity**	**Specificity**	CC	R	R^*
Archaeoglobus fulgidus	0.965	0.967	0.647	0.939	0.92
Aquifex aeolicus	0.979	0.967	0.605	0.950	–
Borrelia burgdorferi	0.978	0.993	0.811	0.973	–
Bacillus subtilis	0.975	0.977	0.820	0.958	0.87
Chlamydia trachomatis	0.981	0.990	0.867	0.974	–
Escherichia coli	0.954	0.990	0.806	0.952	0.91
Haemophilus influenzae	0.982	0.962	0.797	0.951	0.90
Helicobacter pylori	0.980	0.968	0.746	0.953	0.93
Mycoplasma genitalium	0.978	0.924	0.538	0.911	0.96
Methanococcus jannaschii	0.984	0.978	0.851	0.967	0.89
Mycoplasma pneumoniae	0.975	0.947	0.688	0.931	0.92
Methanobacterium thermoautotrophicum	0.970	0.989	0.814	0.963	0.93
Mycobacterium tuberculosis	0.964	0.975	0.723	0.946	–
Pyrococcus horikoshii	0.973	0.939	0.612	0.921	–
Rickettsia prowazekii	0.982	0.982	0.928	0.973	–
Synechocystis PCC6803	0.964	0.985	0.823	0.956	0.91
Treponema pallidum	0.972	0.970	0.653	0.946	–

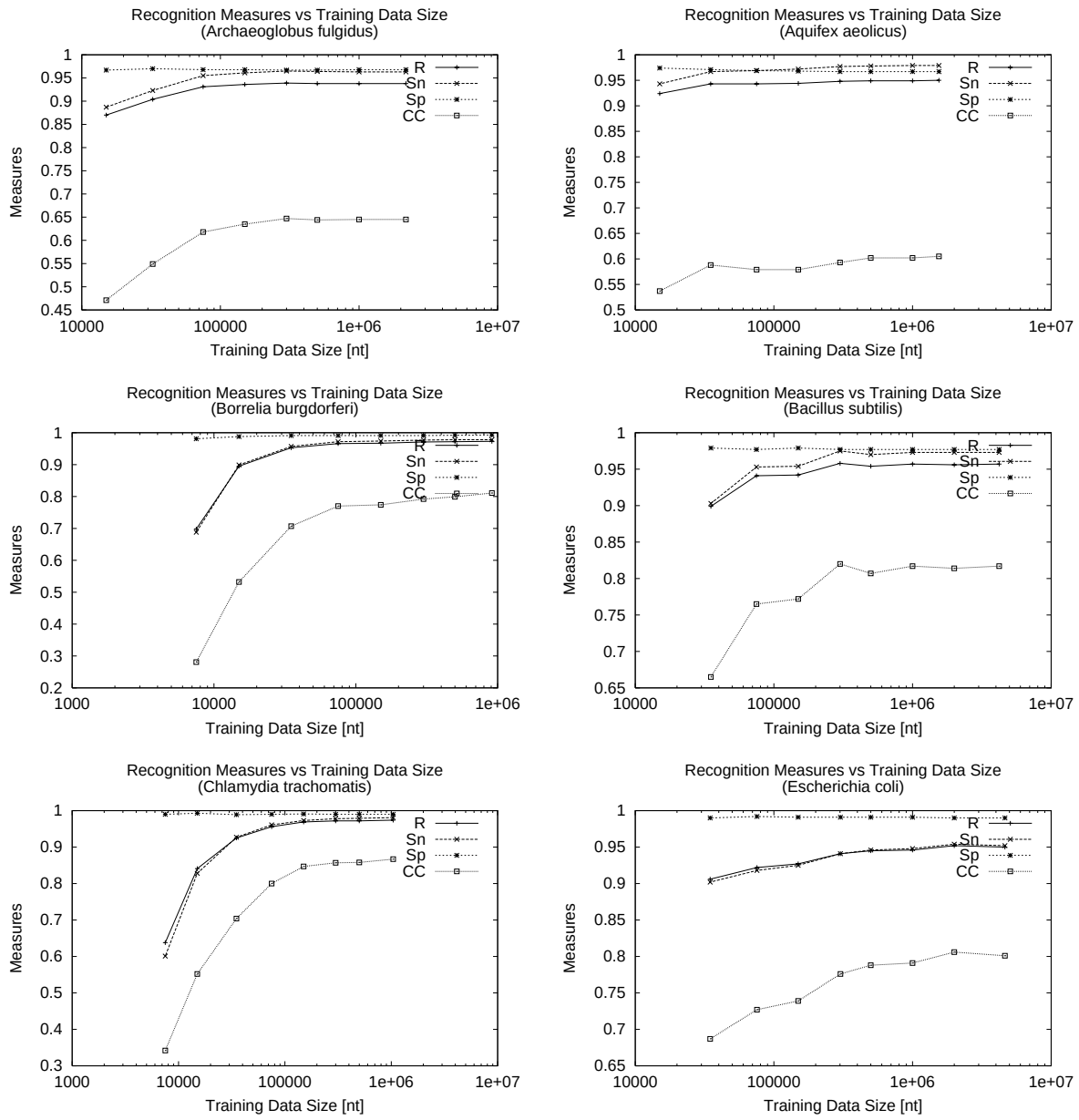

Figure 2.4: Results of gene finding (a). Measures: recognition accuracy (R), sensitivity (Sn), specificity (Sp), and correlation coefficient (CC) for 17 microbial genomic sequence data are shown. We used 1000,000, 500,000, 300,000, 150,000, 75,000, 32,500, 15,000, and 7,500 nt of fragments out of complete genomic sequences for training data of the dicodon-oriented HMM.

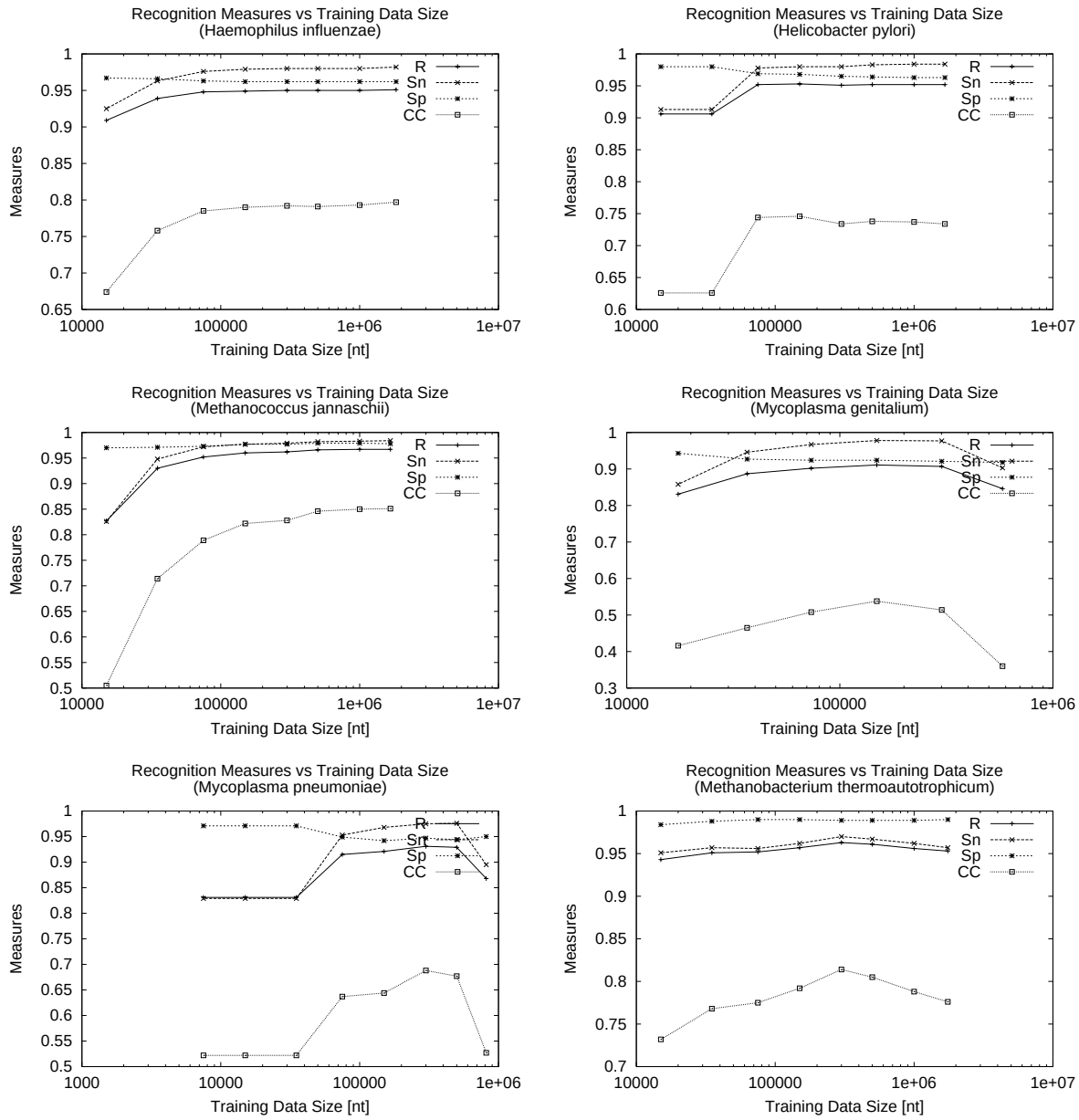

Figure 2.5: Results of gene finding (b) *continued*

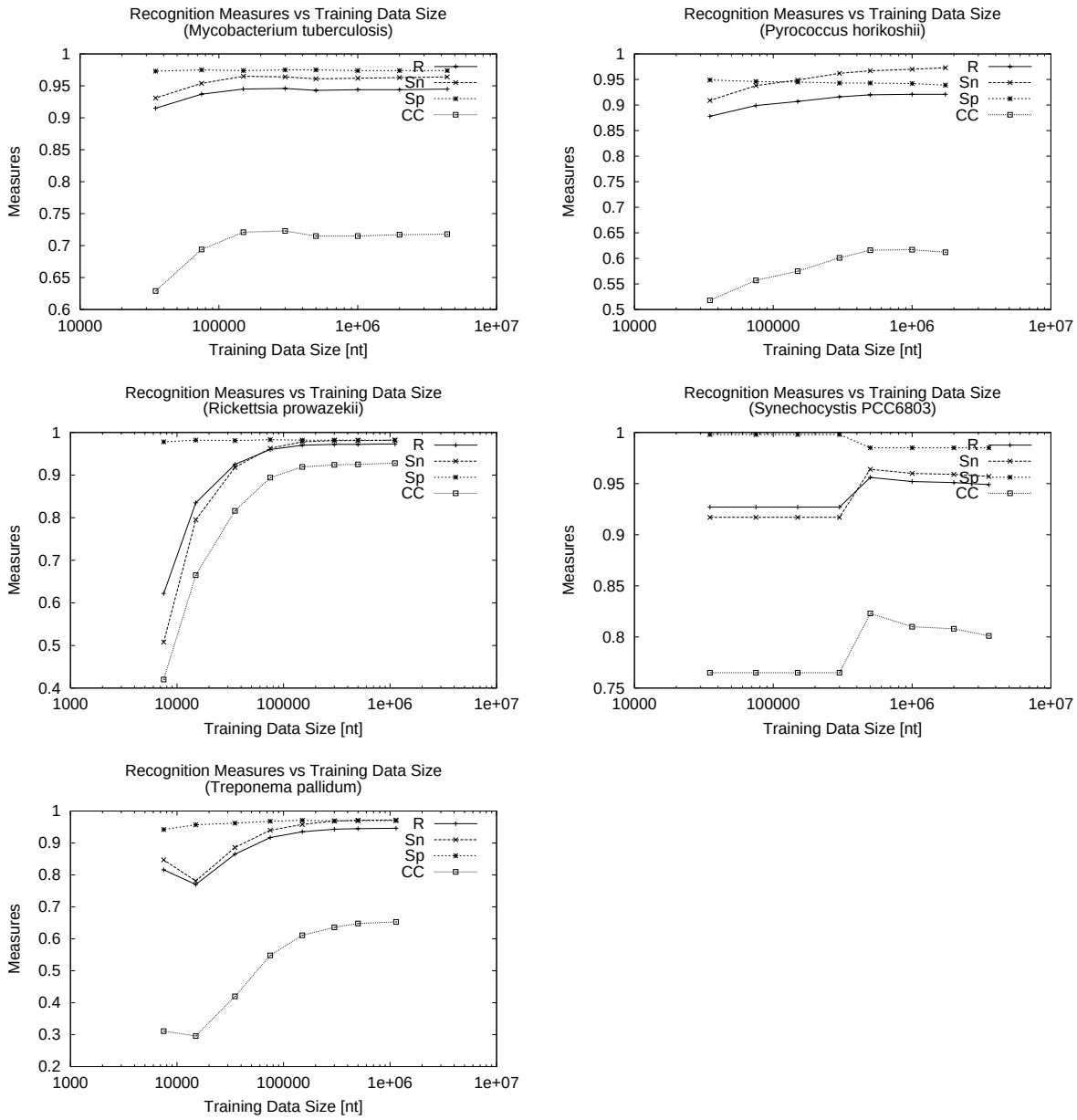

Figure 2.6: Results of gene finding (c) *continued*

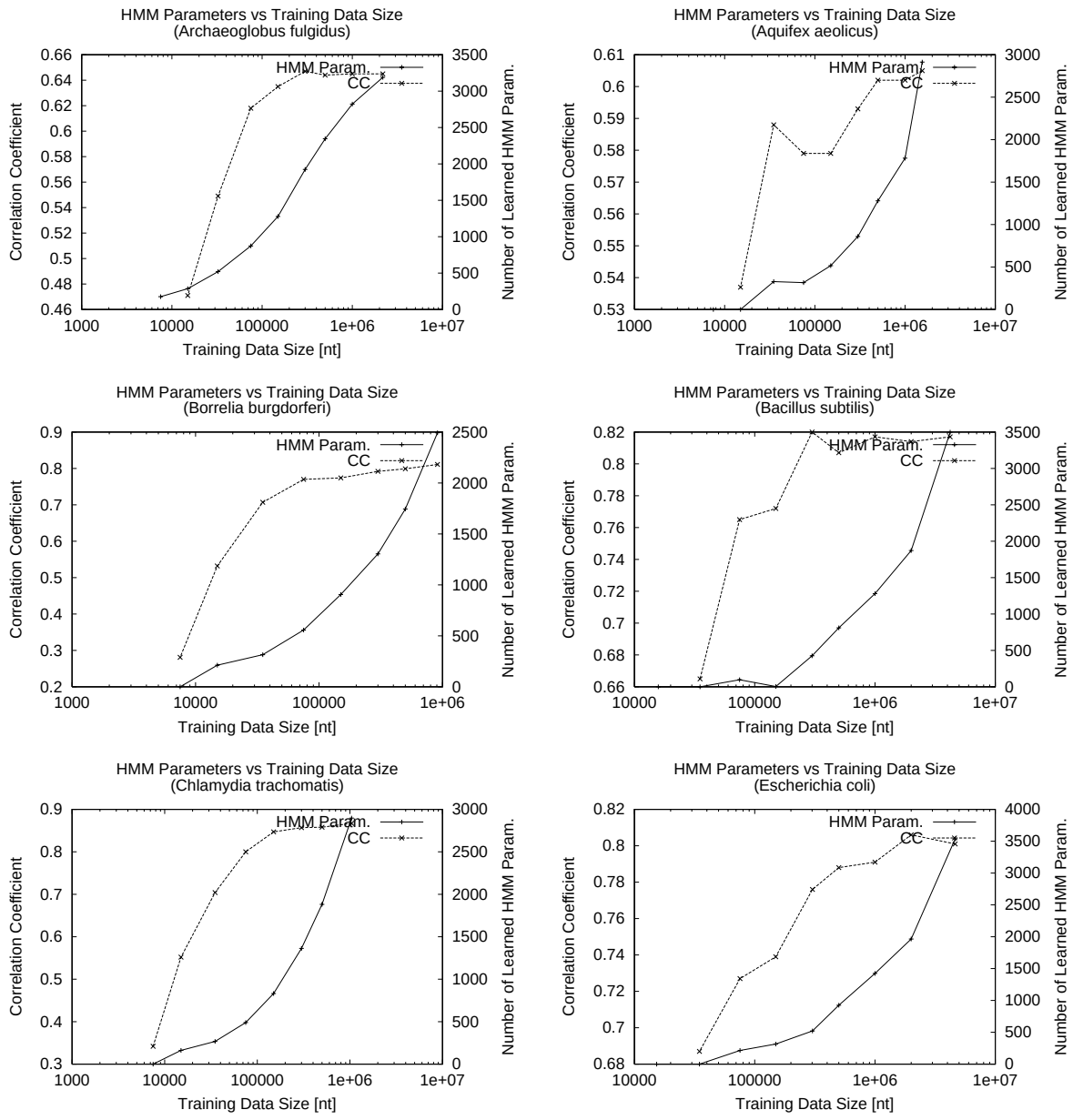

Figure 2.7: Correlation Coefficient and the number of trained HMM parameters for 17 microbial genomic sequence data. (a)

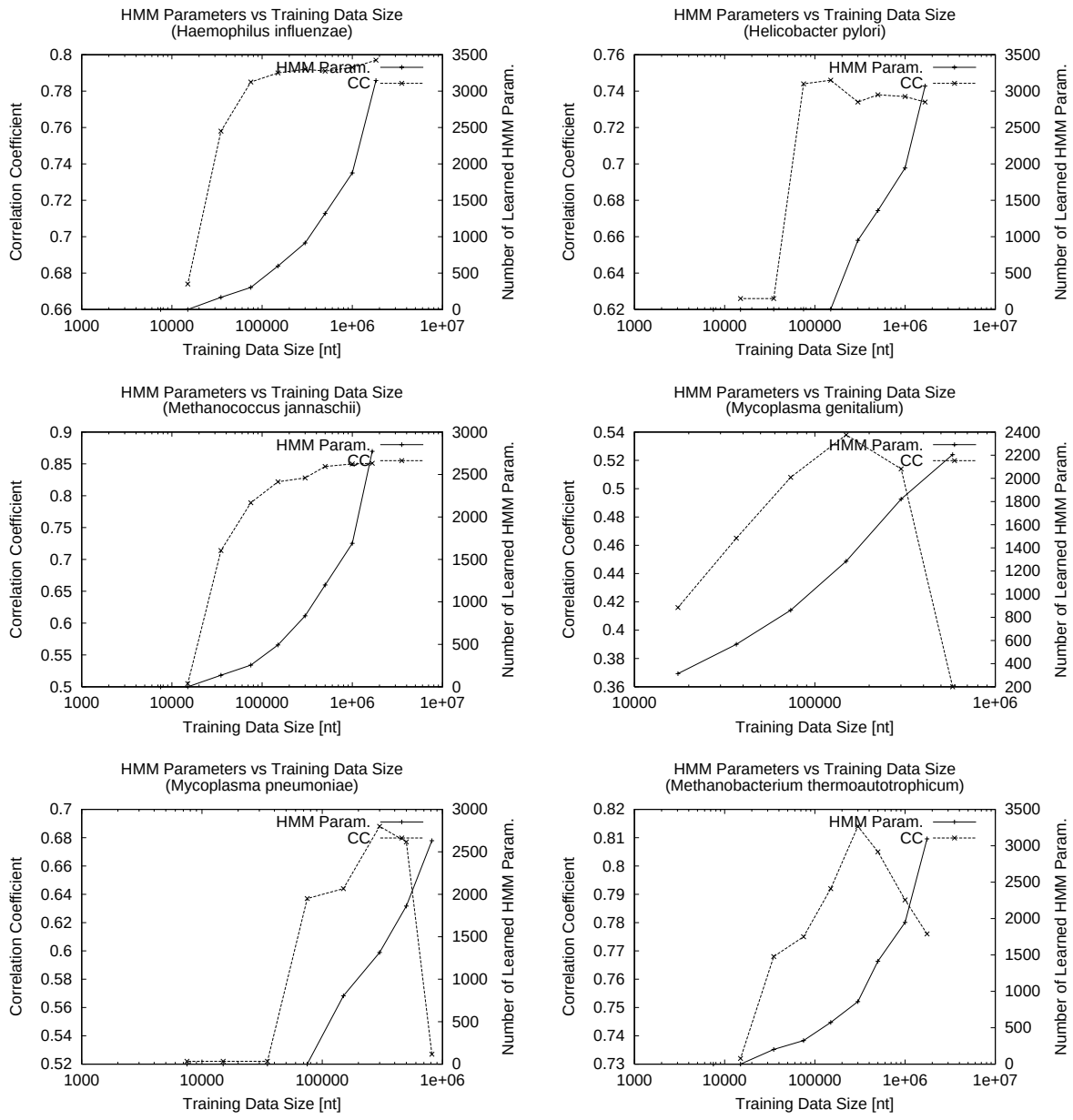

Figure 2.8: Correlation Coefficient and the number of trained HMM parameters for 17 microbial genomic sequence data. (b) *continued*

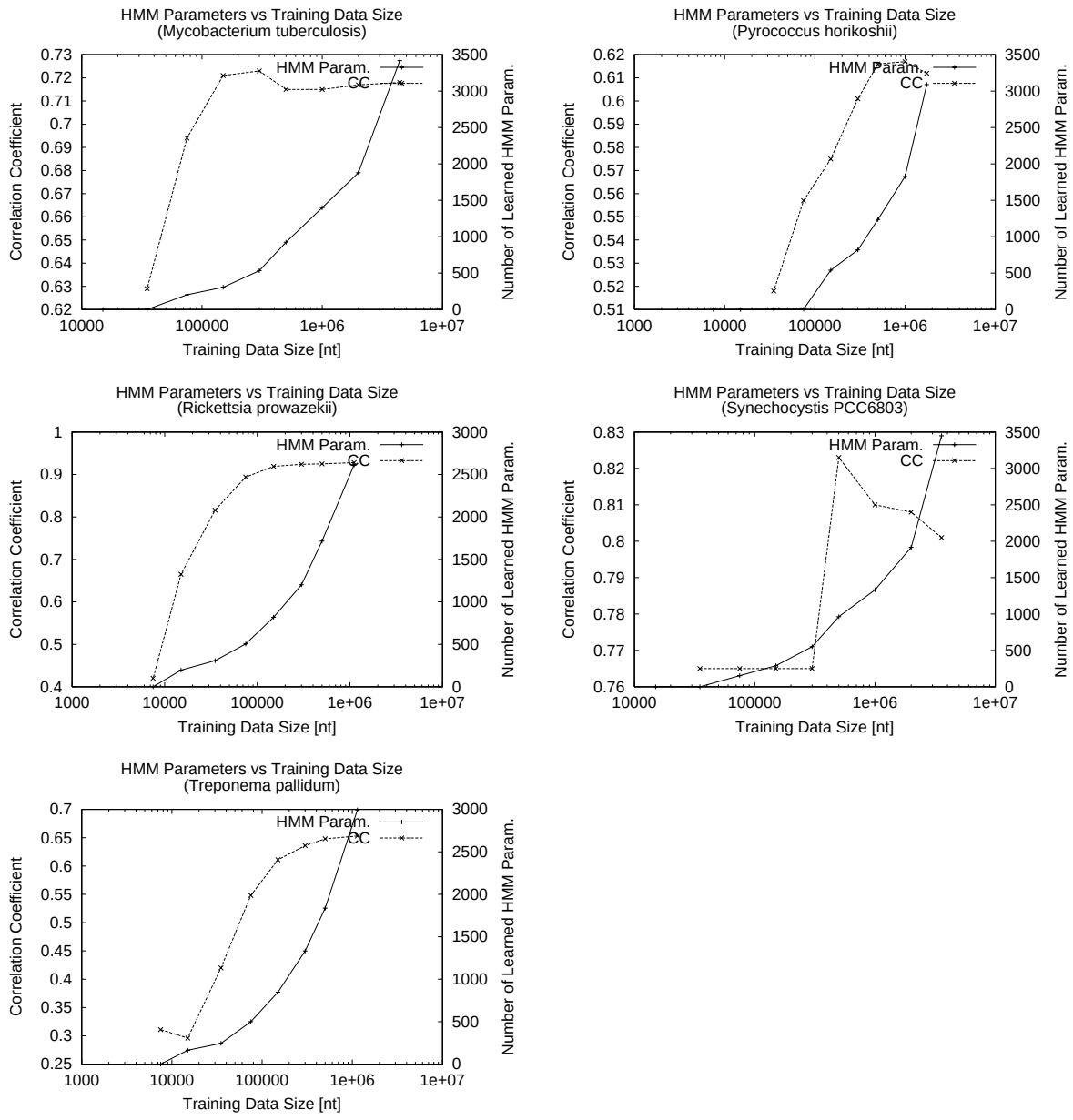

Figure 2.9: Correlation Coefficient and the number of trained HMM parameters for 17 microbial genomic sequence data. (c) *continued*

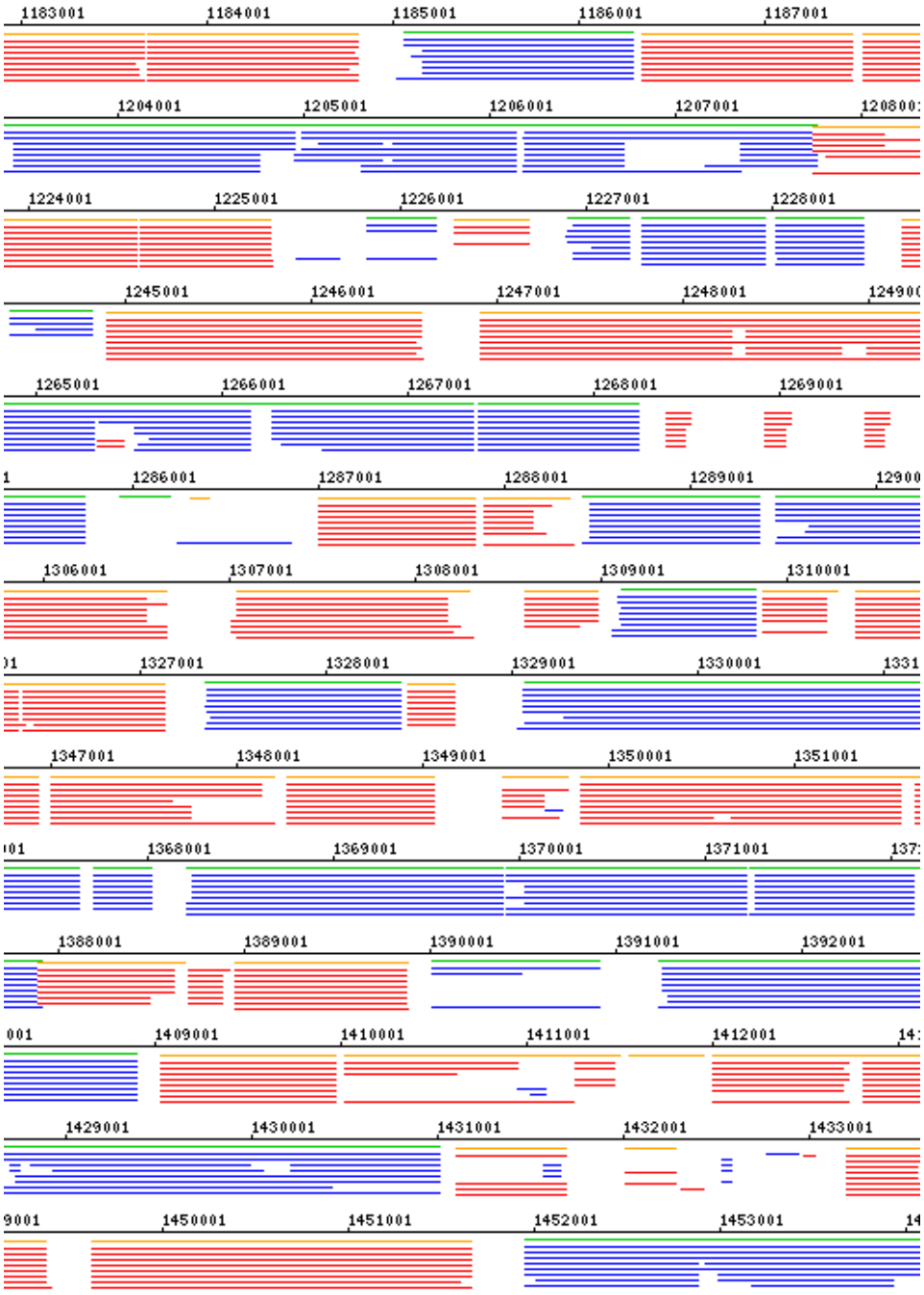

Figure 2.10: A snapshot of detailed prediction results with short fragment of training data for *E.coli* (*originally colored bitmap*). The numbers over the black solid line represent base position in a genomic sequence data. The stripes right bellow of the solid line is showing correct coding regions; greens for 5' to 3' strand and reds for complementary (3' to 5') strands. The blue stripes and orange stripes are representing prediction result with short fragments of training data (top 1/1, bottom 1/256).

less parameter size than the dicodon, is not discovered yet. Thus our next investigation should be to find the most significant element in the dicodon usage measure which make it better than the codon usage measure so that we can discover the intermediate measure. The next section deals with the investigation.

Chapter 3

Evaluation of Dicodon Usage Measure

According to our preliminary examination described above, the redundancy of the dicodon usage measure should be investigated. In this section, we prepared several different probabilistic models to emulate the dicodon model with smaller size of parameters. The size of parameters ranging from 461 to 1,024, is far bellow from the dicodon model which has 3,721. However, as our preliminary examination showed, protein coding regions in some microbial genomic sequence data require very small number of HMM parameters. Thus the models with small size of parameters should be evaluated objectively and quantitatively.

3.1 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. The self-identification learning with too many parameters usually fails to produce good result 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 [11] 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. Figure 3.1 depicts each attributes contained in a nucleotide hexamer.

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

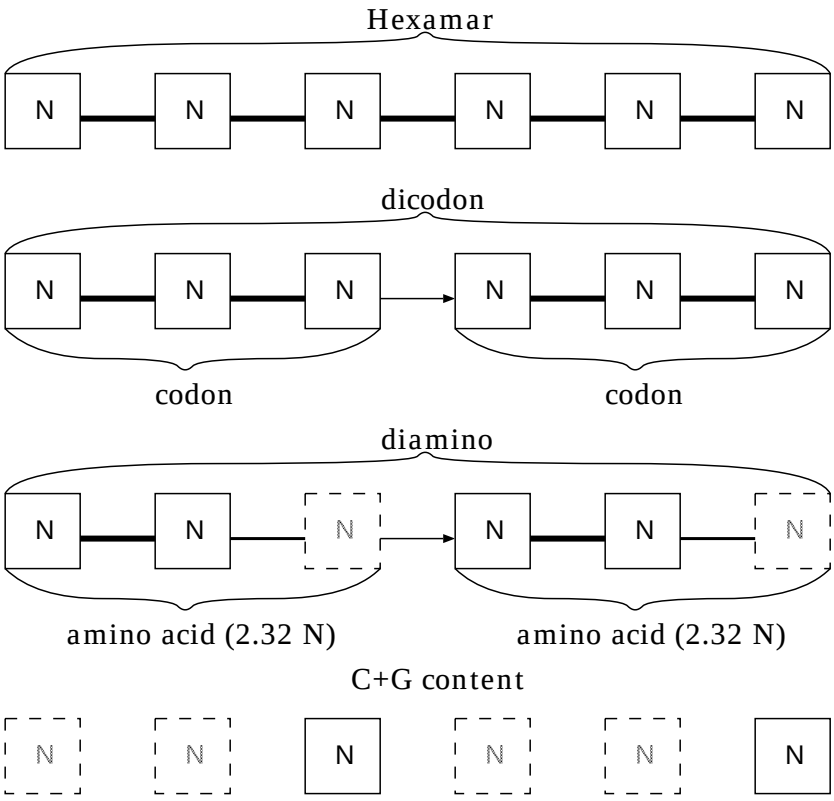

Figure 3.1: The hexamer treats six nucleotide as one datum thus is identical to 6 nucleotide window frame examination. The codon binds three nucleotide as one datum thus the dicodon stands for a pair of codons. A codon corresponds to an amino-acid but an amino-acid corresponds to one or more codons and there are only 20 possible amino-acids while there are 64 possible codons. 8 amino-acids(*family box*) are determined by 2 nucleotides. 12 amino-acids(*2-codon set*) are determined by 2.5 nucleotides. 1 amino-acid(*2-codon set+1*) is determined by 2.75 nucleotides. Approximately, an amino-acid is determined by 2.32 nucleotides. C+G content stands for a biased possibility to have a C or G in the third position of codon. Thus it can not be defined by single nucleotide but a certain length of window frame should be considered.

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. Every model is a probabilistic representation of nucleotide hexamer with emphasis on the codon usage, C+G content and diamino-acid. 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 C+G 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 C+G content of the preceded codon. The model F uses four codon usage sets, which are used based on nucleotide-wise rather on C+G 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 position. 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

$$p_A(c_j|c_i) \equiv p(c_j|c_i). \tag{3.1}$$

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

$$p_B(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j)). \tag{3.2}$$

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

$$p_C(c_j|c_i) \equiv p(A(c_j)|A(c_i))\{\lambda_B p(c_j|A(c_j)) + (1-\lambda_B)p(f_{gc}(c_j)|f_{gc}(c_i))\}. \tag{3.3}$$

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

$$p_D(c_j|c_i) \equiv p(A(c_j)|A(c_i))\{\lambda_C p(c_j|A(c_j)) + (1-\lambda_C)p(f_{atgc}(c_j)|f_{atgc}(c_i))\}. \tag{3.4}$$

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

$$p_E(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j), f_{gc}(c_i)). \tag{3.5}$$

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

$$p_F(c_j|c_i) \equiv p(A(c_j)|A(c_i))p(c_j|A(c_j), f_{atgc}(c_i)). \tag{3.6}$$

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

$$p_G(c_j|c_i) \equiv p(S(c_j)|S(c_i)). \tag{3.7}$$

$A(c)$ 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. *Shrunk* codon does not distinguish G-C, and A-T. For instance, $S(XXG) = S(XXC)$ and $S(XXA) = S(XXT)$.

3.2 Evaluation of models

In order to evaluate these six models(B to G) against the dicodon model, We used 17 microbial genomic sequences [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40] and *C. elegance* [41] genome sequences obtained from GenBank and took following procedure:

(i) So-called *Jack knife strategy* is applied here.

(ii) Several size of *Learning* sets and *Testing* sets are prepared in order to evaluate performance and robustness of each model.

(iii) When an examined genomic sequence has N genes, we take N/n genes out of the sequence randomly($n = 1.3, 1.7, 2, 4$).

(iv) The extracted genes are used for the *Learning* sets.

(v) Rest of the genes and the non-coding regions are used as the *Testing* set.

(vi) Train six models and the dicodon model using the *Learning* set.

(vii) Accumulate coding potentials cod_x of every coding region in the *Testing* set based on the six models.

(viii) Train the dicodon model using the *Testing* set and accumulate a coding potential cod_o.

(ix) Obtain profiles of coding potentials for coding regions and non-coding regions.

(x) Evaluate every models in two ways: Approximation error and Learning/Testing evaluation.

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

$$cod_x(\mathbf{C}) = \frac{1}{n} \log p_x(c_1, c_2, \ldots, c_n) = \frac{1}{n} \log\{p_x(c_2|c_1) \ \ldots \ p_x(c_n|c_{n-1})\} = \frac{1}{n} \sum_{i=2}^{n} \log p_x(c_i|c_{i-1}). \tag{3.8}$$

3.2.1 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.
- A_T is the dicodon model that was trained with testing sequence.
- A_L is the dicodon model that was trained with learning sequence.
- Other models 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 cod_o and coding potentials of the other model x.

  $$D(x) = \sum_{\mathbf{C}} (cod_o(\mathbf{C}) - cod_x(\mathbf{C}))^2 \tag{3.9}$$

  where $x = A_L, B, C, \ldots, G$.
- This evaluation shows how these models accurately approximate the dicodon model.

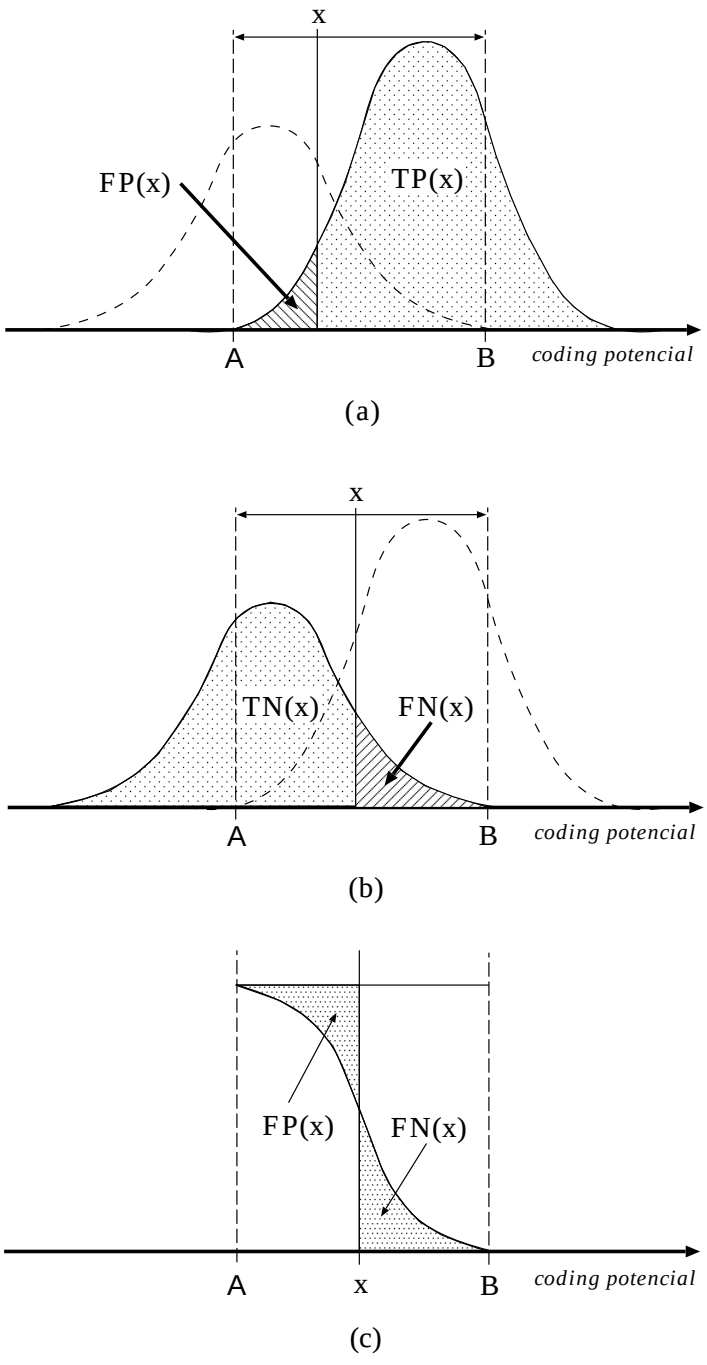

Figure 3.2: Profile of coding potentials for coding(right heap) and non-coding(left heap) regions. The two heaps have overlapped area $[A, B]$. We set a threshold coding potential x within $[A, B]$. (a) For coding potentials over x are taken to be coding regions. So cross-hatched area become *false negatives*. (b) For coding potentials under x are taken to be non-coding regions. The cross-hatched area become *false positives*. (c) We take x so that the sensitivity and specificity become equivalent. According to the definition of sensitivity and specificity, $FP(x) = FN(x)$ when $Sn(x) = Sp(x)$.

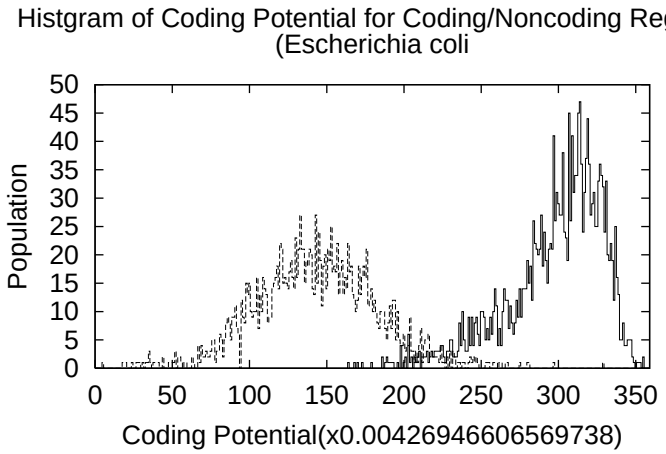

Figure 3.3: Actual histogram of coding/non-coding potential for *E.coli*

3.2.2 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 Figure 3.3. 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* Figure 3.2).

$$
\begin{aligned}
d = Sn(x_0) + Sp(x_0) \quad &, \quad Sn(x_0) = Sp(x_0) \\
Sn(x) = \frac{TP(x)}{TP(x) + FN(x)} \quad &, \quad Sp(x) = \frac{TP(x)}{TP(x) + FP(x)} \\
TN(x) = \sum_{i=x_{min}}^{x} h_{nc}(i) \quad &, \quad FN(x) = \sum_{i=x}^{x_{max}} h_{nc}(i)
\end{aligned}
\tag{3.10}
$$

$$TP(x) = \sum_{i=x}^{x_{max}} h_{cd}(i) \quad , \quad FP(x) = \sum_{i=x_{min}}^{x} h_{cd}(i)$$

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

3.3 Result

Table 3.1 shows maximum sensitivity+specificity of every model for 14 microbial genomic sequence data and 14 eukaryotic genomic sequence data. Mean sensitivity+specificity is shown in bottom of the table. Although the mean sensitivity+specificity shows that the dicodon and the model G (*shrunk dicodon model*) yield equivalent value, the details are different in each species. The dicodon scores higher than the model G in 15 species while the model G scores higher in other species.

Figure 3.4 to 3.8 show sensitivity+specificity versus relative training data size of 14 microbial genomic sequence data and 14 eukaryotic genomic sequence data. The sensitivity+specificity scores tend to wobble because of the jack knife strategy. The jack knife strategy requires approximation in order to get smooth result. However we did just one time examination.

Figure 3.3 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).

Table 3.1: Maximum sensitivity+specificity of every model for 14 microbial genomic sequence data and 14 eukaryotic genomic sequence data. Mean sensitivity+specificity is shown in bottom of the table.

Species	**dicodon**	**B**	**C**	**D**	**E**	**F**	**G**
Archaeoglobus fulgidus	1.976	1.960	1.962	1.960	1.961	1.965	1.976
Aquifex aeolicus	1.924	1.910	1.908	1.908	1.913	1.918	1.932
Borrelia burgdorferi	1.954	1.887	1.887	1.890	1.882	1.911	1.950
Bacillus subtilis	1.950	1.923	1.923	1.924	1.923	1.932	1.949
Chlamydia trachomatis	1.962	1.887	1.891	1.887	1.898	1.902	1.962
Escherichia coli	1.959	1.940	1.941	1.941	1.941	1.944	1.959
Haemophilus influenzae	1.951	1.926	1.924	1.926	1.920	1.932	1.946
Mycoplasma genitalium	1.881	1.821	1.785	1.813	1.833	1.840	1.873
Methanococcus jannaschii	1.973	1.920	1.921	1.918	1.924	1.940	1.970
Mycoplasma pneumoniae	1.856	1.823	1.833	1.823	1.831	1.835	1.856
Methanobacterium	1.956	1.946	1.946	1.947	1.950	1.950	1.954
Rickettsia prowazekii	1.975	1.920	1.923	1.920	1.931	1.930	1.970
Synechocystis sp.	1.946	1.922	1.922	1.923	1.923	1.932	1.950
Treponeuma pallidum	1.900	1.880	1.877	1.877	1.872	1.899	1.918
C. elegance(Chr I)	1.931	1.758	1.759	1.759	1.799	1.805	1.921
C. elegance(Chr II)	1.927	1.753	1.752	1.760	1.795	1.798	1.923
C. elegance(Chr III)	1.930	1.769	1.774	1.782	1.804	1.816	1.919
C. elegance(Chr IV)	1.946	1.816	1.819	1.822	1.846	1.859	1.938
C. elegance(Chr V)	1.929	1.739	1.743	1.744	1.778	1.798	1.918
Saccharomyces cerevisiae(Chr II)	1.755	1.615	1.644	1.639	1.620	1.681	1.780
Saccharomyces cerevisiae(Chr III)	1.674	1.545	1.523	1.545	1.536	1.549	1.709
Saccharomyces cerevisiae(Chr IV)	1.904	1.828	1.815	1.827	1.834	1.847	1.907
Saccharomyces cerevisiae(Chr VI)	1.724	1.597	1.605	1.580	1.642	1.591	1.742
Saccharomyces cerevisiae(Chr VIII)	1.847	1.672	1.678	1.714	1.700	1.754	1.832
Saccharomyces cerevisiae(Chr X)	1.835	1.727	1.725	1.727	1.748	1.762	1.843
Saccharomyces cerevisiae(Chr XI)	1.861	1.689	1.704	1.717	1.730	1.753	1.849
Saccharomyces cerevisiae(Chr XIII)	1.888	1.781	1.786	1.794	1.801	1.812	1.887
MEAN	1.896	1.808	1.809	1.812	1.822	1.834	1.896

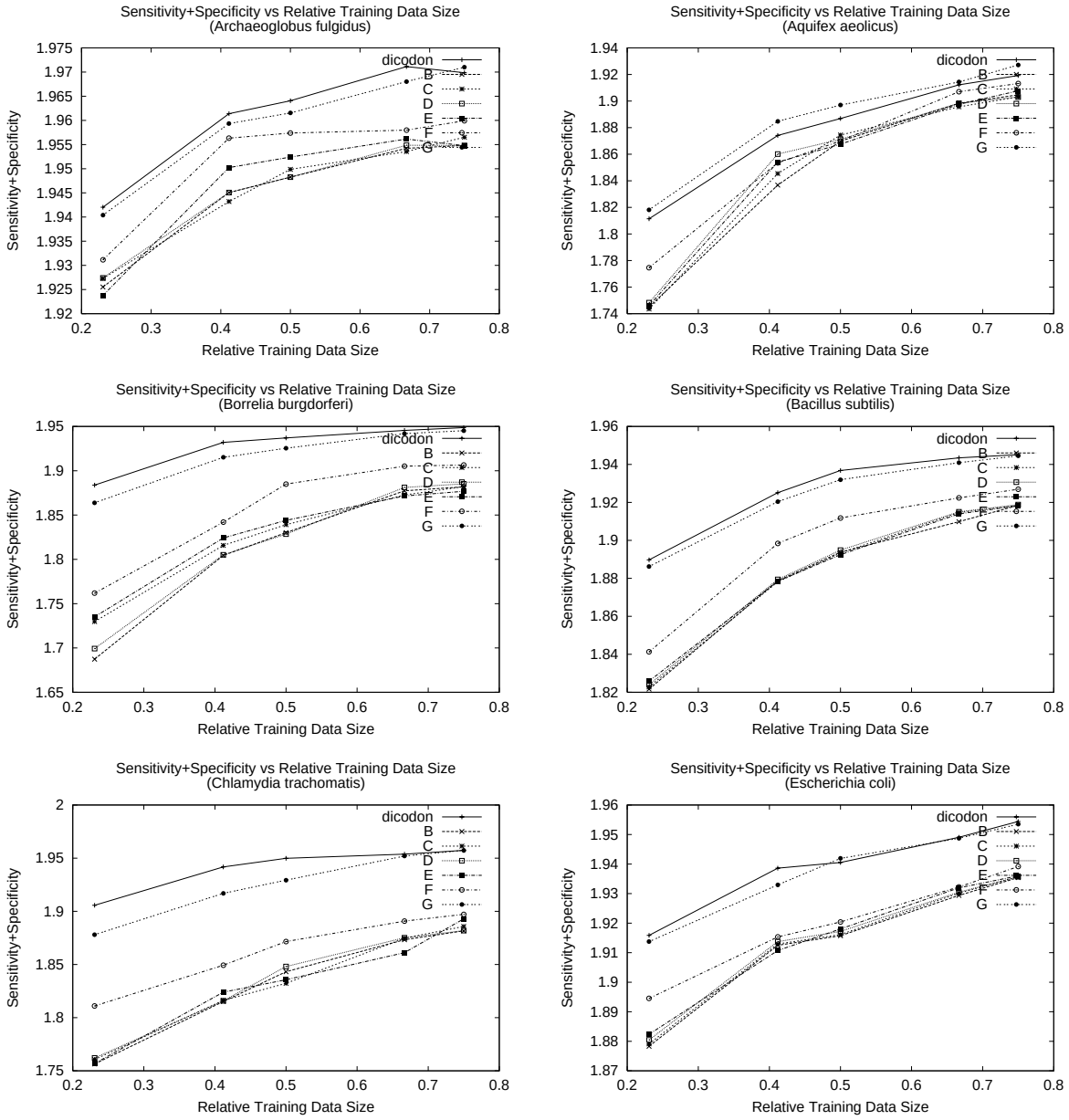

Figure 3.4: Sensitivity+Specificity versus relative training data size for 14 microbial genomic sequence data and 14 eukaryotic genomic sequence data (a)

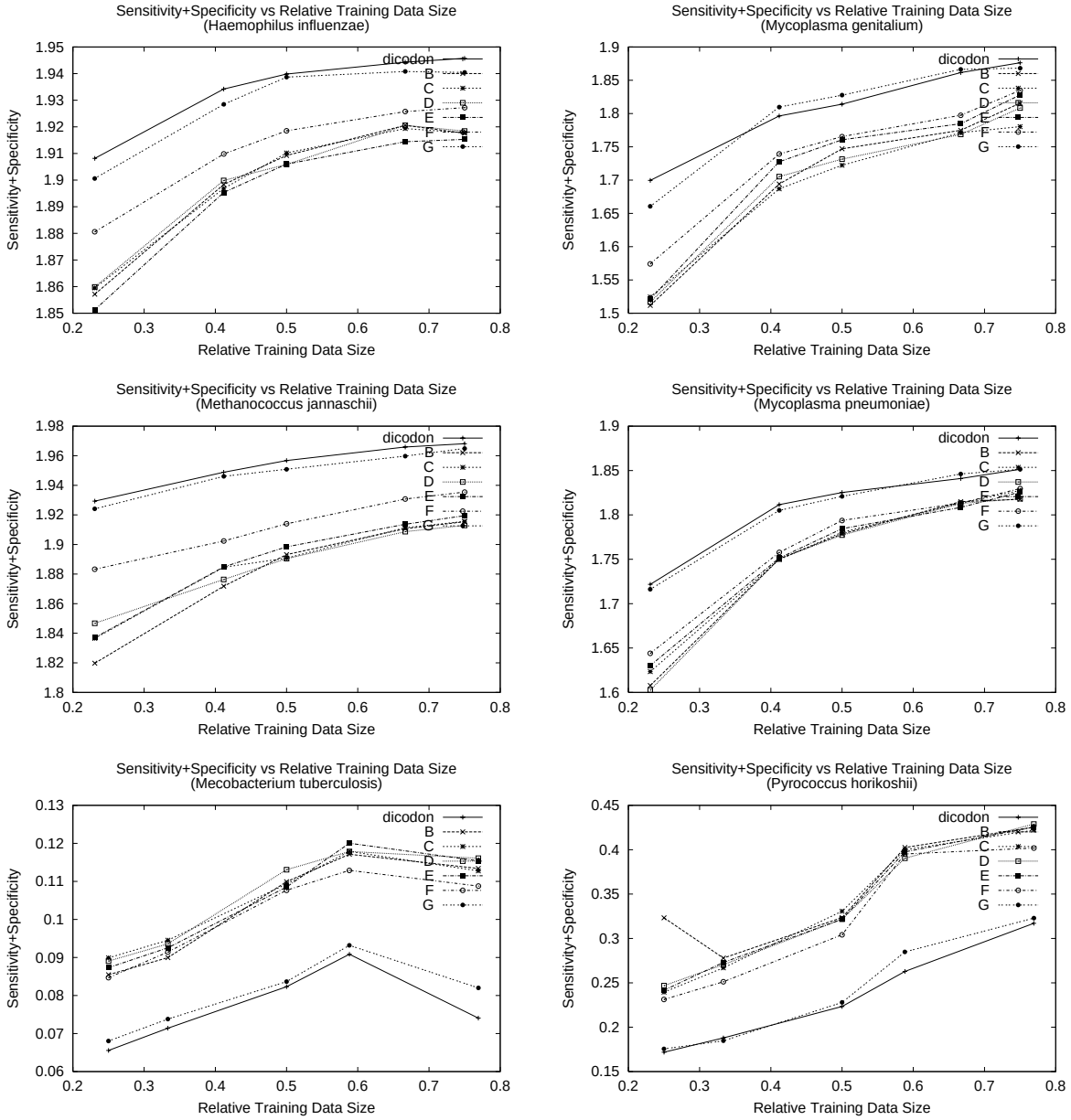

Figure 3.5: Sensitivity+specificity versus relative training data size (b) *continued*

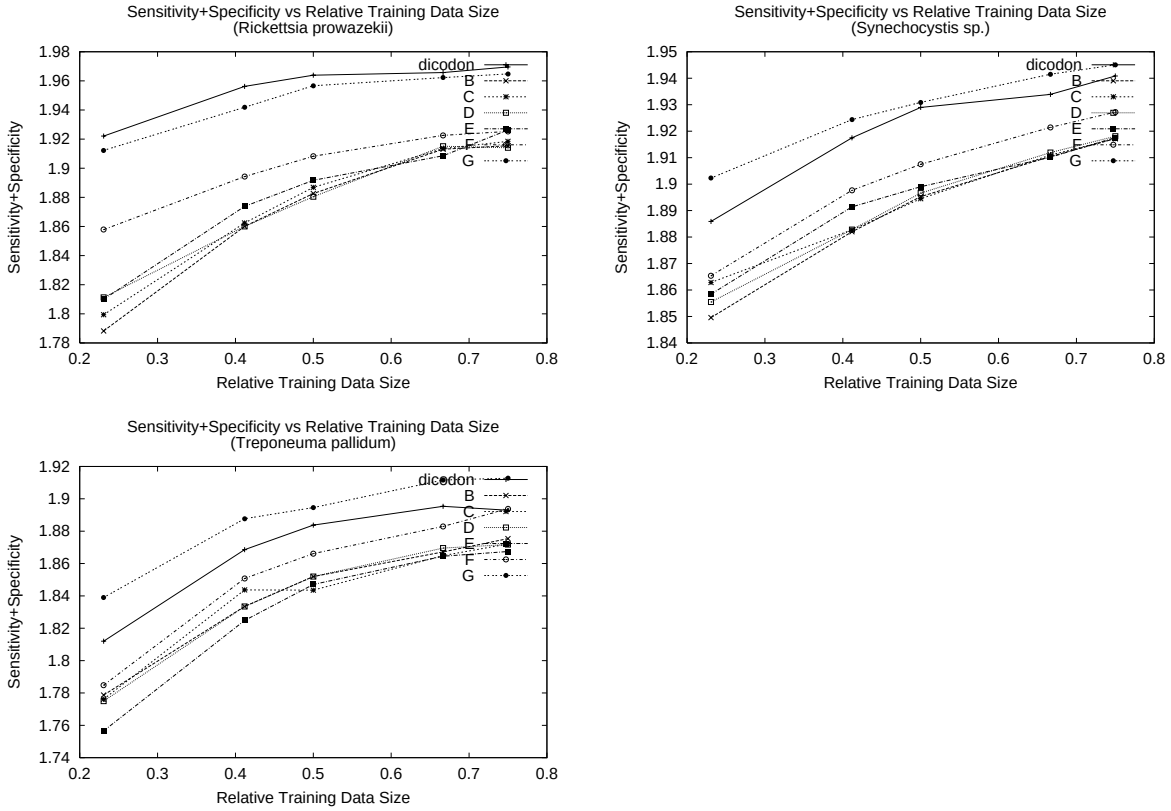

Figure 3.6: Sensitivity+Specificity versus relative training data size (c) *continued*

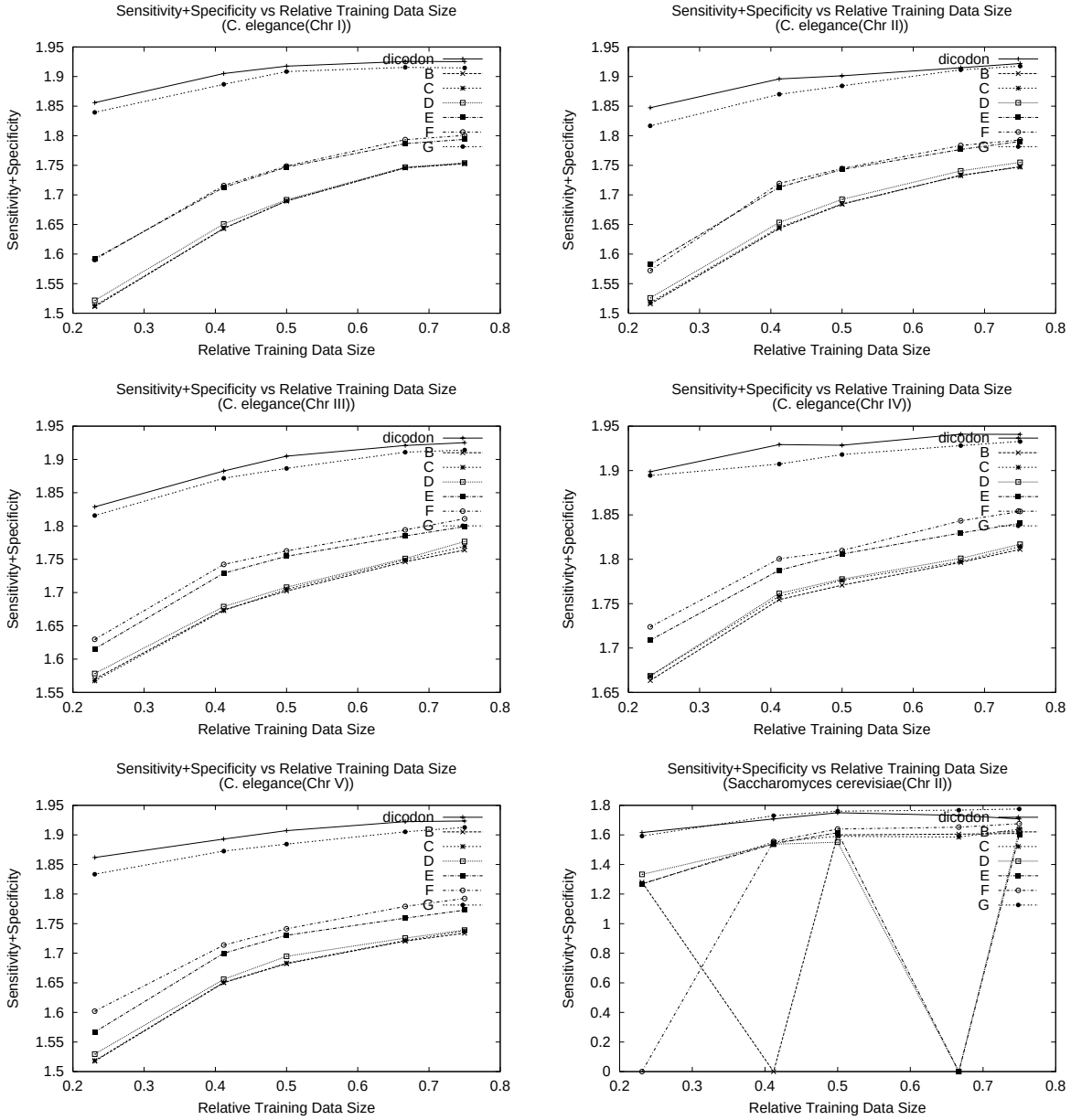

Figure 3.7: Sensitivity+Specificity versus relative training data size (d) *continued*

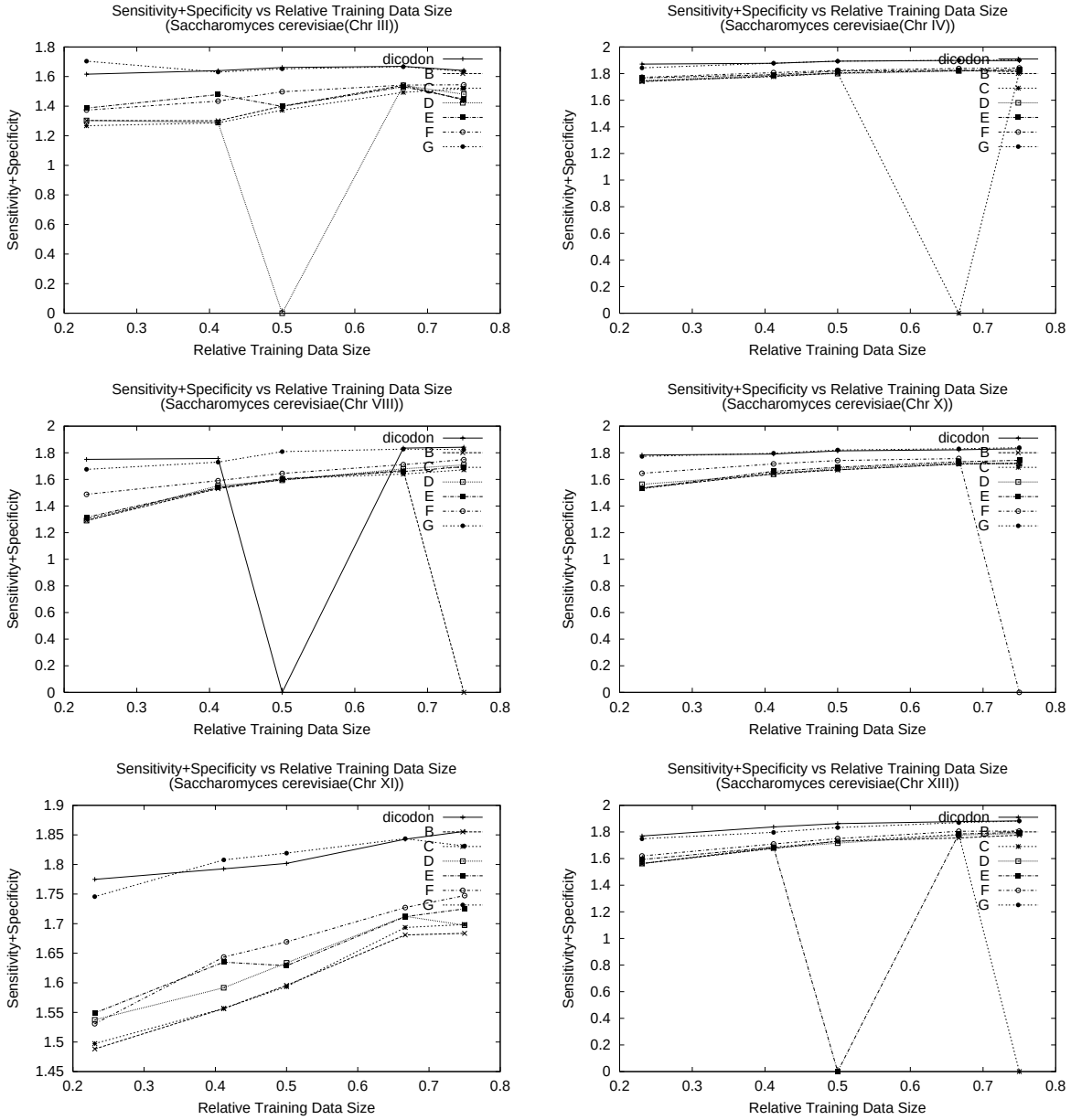

Figure 3.8: Sensitivity+Specificity versus relative training data size (e) *continued*

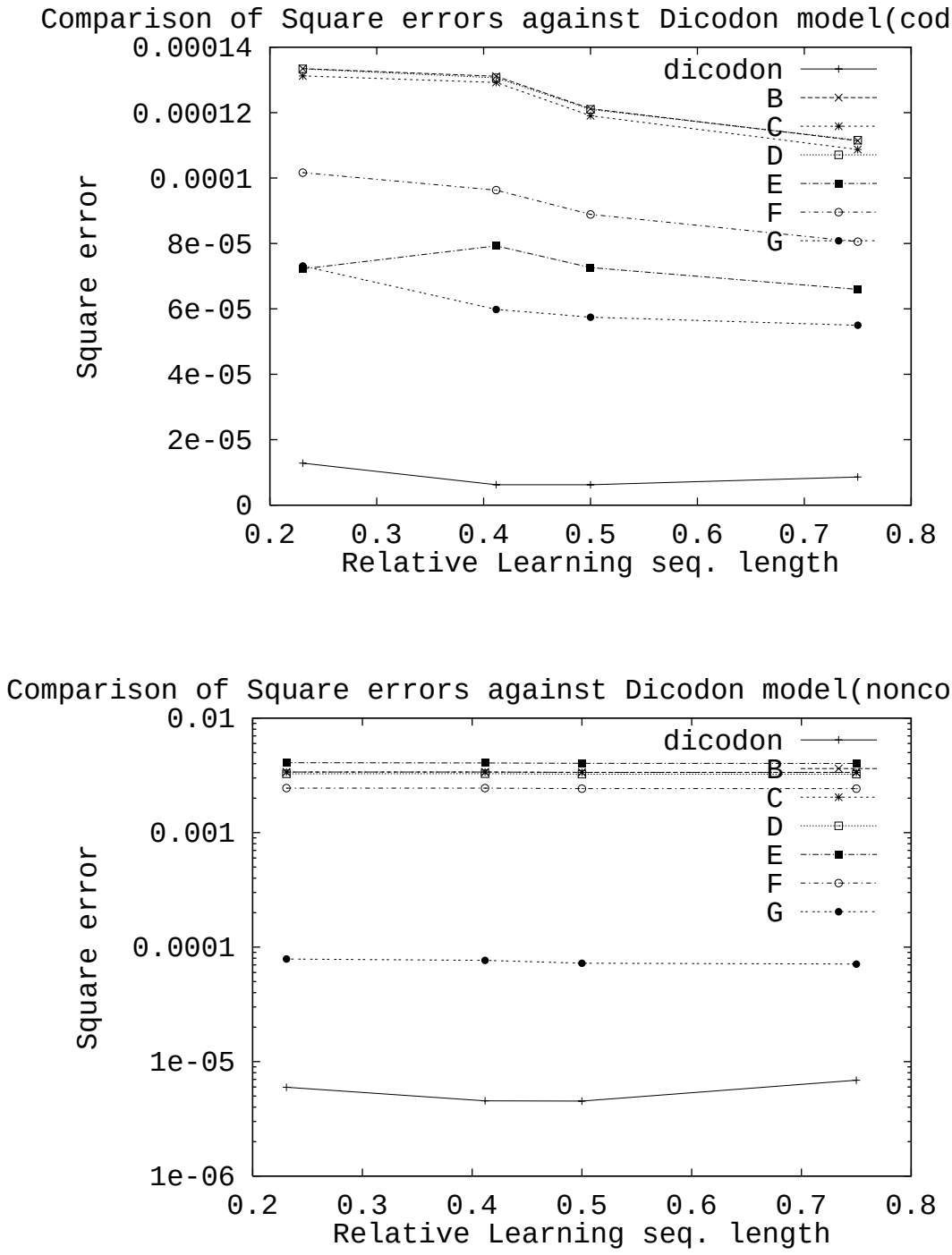

Figure 3.9: The left figure shows square errors of the coding potentials of testing sequences(above: coding regions, bellow:non-coding regions) for each models against the coding potential of dicodon model that was trained with testing sequences. The square errors are average values over 13 microbial and 2 eukaryotic genomes.

3.4 Discussion

Our evaluation shows that the dicodon model outperforms other six models(B to G) in terms of specificity and sensitivity (Table 3.1 and Figure 3.4 to 3.8). Besides, none of the emulation models(B to F) get closer than the model G in terms of approximation error(Figure 3.3).

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 C+G 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 C+G content.

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

The performance of the model E shows clearer improvements. This result indicates that the second codon usage depends on the C+G 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 C+G content. This indicates that a bias at the third nucleotide is not so uniform among G-C and A-T, and C+G 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 C+G content is not enough for describing peculiar bias which is found at the third nucleotide.

Chapter 4

Conclusion

Firstly, we proposed that the redundancy of dicodon usage measure for gene finding in Chapter 2 based on the result obtained from our preliminary gene finding examination using dicodon oriented HMM with self-identification learning method, which showed that the HMM could predict protein coding region in microbial genomic sequence data with far less parameter size than the HMM employed. According to the fact, we performed the evaluation of the dicodon usage measure using 6 probabilistic models that emulate the dicodon usage measure with less parameter size than that in order to clarify the most significant element consists of the dicodon in Chapter 3. However the all emulation models, except shrunk dicodon model, failed to attain such high accuracy provided by the dicodon model. Although the shrunk dicodon model produced the result close to that of the dicodon model, it can not be identical to the dicodon based on the result of our evaluation. This fact showed that the dicodon usage measure can not be described by codon usage, pair amino acid, and C+G content. This negative result negates the widely believed common sense and, more importantly, proposed a new fact that a certain important element other than codon usage, pair amino-acid, and C+G content is still missed and the missing element clarified. This paper does not deal with the missing element but indicated that the C+G content is not sufficient to emulate the dicodon model.

Chapter 5

Appendix

Table 5.1: 17 microbial genomic sequence data.

Species	**Acc. No**	**Length (nt)**
Archaeoglobus fulgidus	AE000782	2178400
Aquifex aeolicus	AE000657	1551335
Borrelia burgdorferi	AE000783	910724
Bacillus subtilis	AL009126	4214814
Chlamydia trachomatis	AE001273	1042519
Escherichia coli	U00096	4639221
Haemophilus influenzae	L42023	1830138
Helicobacter pylori	AE000511	1667867
Mycoplasma genitalium	L43967	580074
Methanococcus jannaschii	L77117	1664970
Mycoplasma pneumoniae	U00089	816394
Methanobacterium thermoautotrophicum	AE000666	1751377
Mycobacterium tuberculosis	AL123456	4411529
Pyrococcus horikoshii	Pyro_h	1738505
Rickettsia prowazekii	AJ235269	1111523
Synechocystis PCC6803	AB001339	3573470
Treponema pallidum	AE000520	1138011

Species	**Acc. No**	**Length (nt)**
Caenorhabditis elegans chromosome I	chr_I	16,183,833
Caenorhabditis elegans chromosome II	chr_II	17,004,925
Caenorhabditis elegans chromosome III	chr_III	12,114,540
Caenorhabditis elegans chromosome I	chr_IV	15,887,371
Caenorhabditis elegans chromosome V	chr_V	21,280,512
Caenorhabditis elegans chromosome X	chr_X	17,624,844

Acknowledgments

Firstly, I would like to put my best appreciation to my family. I would like to express my profound appreciation to Dr. Konagaya for his thickest support for my student life and research activity. This paper would not be published without his conscience advise and encouragement. I also greatly appreciate Dr. Asai for his remarkable criticism and contribution to my research activity. This work would not even exist without him. Dr. Satou has been the best industrious person as far as I know. His sound and accurate criticism has been of great help to me. Dr. Takahashi has been giving me coherent and practical advise to carry my research activity on. His influence is also gratefully appreciated. All of students of Genetic Knowledge System Laboratory of Japan Advanced Institute of Science and Technology has been of the best assistance in every aspect of my student life. I thank them from my heart profoundly.

Publication

- Kim, C., Konagaya, A., Asai, K.:
  A Generic Criterion for Gene Recognition in Genomic Sequences,
  Proc. of Genome Informatics Workshop 1999, pp.13-22.

- Kim, C., Konagaya, A., Asai, K.:
  A Gene Finding using a di-codon oriented Hidden Markov Model (in Japanese),
  Proc. of the 13th Annual Conference of JSAI, 1999., pp.330-331.

Bibliography

Papers

[1] Asai K., Itou K., Ueno Y.
Recognition of Human Genes by Stochastic Parsing.
Pacific Symposium on Biocomputing 1998:228-39

[2] Asai K., *et al.*
Automatic Gene Recognition without Using Training Data.
Genome Informatics Workshop Conf. Proc. 1997.

[3] Audic S., Claverie J. M.
Self-identification of protein-coding regions in microbial genomes.
Proc Natl Acad Sci U S A 1998 Aug 18;95(17):10026-31

[4] Baum, L. E., *et al*
A Maximization Technique Occuring in The Statistical Analysis of Probabilistic Functions of Markov Chains.
Annals of Mathematical Statistics, 1970, 41(1), 164-171

[5] Benson D. A., *et al*
GenBank.
Nucleic Acids Res 1999 Jan 1;27(1):12-7

[6] Burge, C.
Identification of Genes in Human Genomic DNA (Doctoral Thesis).
Stanford University March 1997

[7] Burset, M., Guigo, R.
Evaluation of Gene Structure Prediction Programs.
Genomics, 1996, **34**, 353-67

[8] Churchill, G. A.
Stochastic Models for Heterogeneous DNA Sequences.
Bull. Math. Biol., 1989, 51, 79-94

[9] Dong, S. and Searls, D. B.
Gene structure prediction by linguistic methods.
Genomics, 23, 540-0551, 1994.

[10] Fickett, J. W.
Recognition of protein coding regions in DNA sequences.
Nucleic Acid Research, 1982, 10, 5503-5518

[11] Fickett J. W., Tung C. S.
Assessment of protein coding measures.
Neucleic Acid Research, 1992, Vol. 20, No. 24:6441-50

[12] Fickett, J. W.
The Gene Identification Problem: An Overview For Developers.
Computers Chem., 1996, 20(1):103-118

[13] Forney, Jr. G. D.
The Viterbi Algorithm.
Proc. of the IEEE, 1973, 61(3):268-78

[14] Krogh A., *et al.*
A hidden Markov model that finds genes in E. coli DNA.
Nucleic Acids Res. 1994 Nov 11;22(22):4768-78.

[15] Rabiner, L. R., Juang, B. H.
An Introduction to Hidden Markov Models.
IEEE ASSP Magazine, Jan. 1986,

[16] Searls, D. B.
The linguistics of DNA.
American Scientist, 1992, 80, 579-591

[17] Shepherd, J. C. W.
Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence statistics, identification, and applications to genome project.
Proc. Natl. Acad. Sci. USA, 1981, 78, 1596-1600

[18] Staden, R., McLachlan, A. D.
Codon preference and its use in identifying protein regions in long DNA sequences.
Nucleic Acid Research, 1984, 12, 505-519

[19] Yada T., Hirosawa M.
Gene recognition in cyanobacterium genomic sequence data using the hidden Markov model.
Ismb 1996;4:252-60

[20] Yada, T., *et al*
Signal Pattern Extraction from DNA Sequences Using Hidden Markov Model and Genetic Algorithm.
IPSJ Trans., 1996, 37(6):1117-29

Books

[21] Durbin, R., Eddy, S., Krogh, A., Mitchison, G.
Biological sequence analysis.
Cambridge University Press 1998

[22] Osawa, S.
Evolution of the Genetic Code.
Oxford University Press 1995

[23] Steve Young, *et al.*
HTK Book (for HTK version 2.2), Entropic Inc. 1999,
ftp://ftp.entropic.com/pub/htk/HTKBook_a4.ps.gz

Complete Genomes

[24] Klenk H. P., *et al.*
The complete genome sequence of the hyperthermophilic, sulphate-reducing archaeon Archaeoglobus fulgidus.
Nature 1997 Nov 27;390(6658):364-70

[25] Deckert G., *et al.*
The complete genome of the hyperthermophilic bacterium Aquifex aeolicus.
Nature 1998 Mar 26;392(6674):353-8

[26] Fraser C. M., *et al.*
Genomic sequence of a Lyme disease spirochaete, Borrelia burgdorferi.
Nature 1997 Dec 11;390(6660):580-6

[27] Kunst F., *et al.*
The complete genome sequence of the gram-positive bacterium Bacillus subtilis.
Nature 1997 Nov 20;390(6657):249-56

[28] Stephens R. S., *et al.*
Genome sequence of an obligate intracellular pathogen of humans: Chlamydia trachomatis.
Science 1998 Oct 23;282(5389):754-9

[29] Blattner F. R., *et al.*
The complete genome sequence of Escherichia coli K-12.
Science 1997 Sep 5;277(5331):1453-74

[30] Fleischmann R. D., *et al.*
Whole-genome random sequencing and assembly of Haemophilus influenzae Rd.
Science 1995 Jul 28;269(5223):496-512

[31] Tomb J. F., *et al.*
The complete genome sequence of the gastric pathogen Helicobacter pylori.
Nature 1997 Aug 7;388(6642):539-47

[32] Fraser C. M., *et al.*
The minimal gene complement of Mycoplasma genitalium.
Science 1995 Oct 20;270(5235):397-403

[33] Bult C. J., *et al.*
Complete genome sequence of the methanogenic archaeon, Methanococcus jannaschii.
Science 1996 Aug 23;273(5278):1058-73

[34] Himmelreich R., *et al*
Complete sequence analysis of the genome of the bacterium Mycoplasma pneumoniae.
Nucleic Acids Res 1996 Nov 15;24(22):4420-49

[35] Smith D. R., *et al.*
Complete genome sequence of Methanobacterium thermoautotrophicum deltaH: functional analysis and comparative genomics.
J Bacteriol 1997 Nov;179(22):7135-55

[36] Cole S. T., *et al.*
Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence.
Nature 1998 Jun 11;393(6685):537-44

[37] Kawarabayasi Y., *et al.*
Complete sequence and gene organization of the genome of a hyper-thermophilic archaebacterium, Pyrococcus horikoshii OT3.
DNA Res 1998 Apr 30;5(2):55-76

[38] 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 1996 Jun 30;3(3):109-36

[39] Fraser C. M., *et al.*
Complete genome sequence of Treponema pallidum, the syphilis spirochete.
Science 1998 Jul 17;281(5375):375-88

[40] Andersson S. G., *et al.*
The genome sequence of Rickettsia prowazekii and the origin of mitochondria.
Nature 1998 Nov 12;396(6707):133-40

[41] The C. elegans Sequencing Consortium.
Genome sequence of the nematode C. elegans: a platform for investigating biology.
Science 1998 Dec 11;282(5396):2012-8