Gene Finding with Hidden Markov Models

Kensho Kazama s1070057 Supervised by Hiroshi Toyoizumi

Abstract

The purpose of this research is to develop a gene finding program with HMMs and to evaluate the performance. The gene finding program uses the Viterbi algorithm which is one of the most common algorithms of HMMs. HMM is a graphical probabilistic model and is advantageous for organizing the gene structure. We modeled some HMMs for gene finding in prokaryotes, and executed the gene finding program with them. Also, we succeeded in improving the performance by adding the information of Markov Chain to HMMs.

1 Introduction

All creatures consist of cells, and each cell has DNA. DNA is a macromolecule that consists of four nucleotides: A (Adenine), T (Thymine), G (Guanine), and C (Cytosine). In a DNA sequence, there are many genes needed for the formation of creatures; the set of genes is called the "genome." Genomic DNA can be classified into two types: prokaryote and eukaryote. Prokaryotes are the creatures whose cells do not have nuclear membranes such as bacteria. DNA sequences of prokaryote are short and gene coding regions are densely distributed; approximately 90% of the sequence is the gene-coding region. Eukaryotes are advanced creatures such as animals and plants whose cells have nuclear-wrapped membranes. DNA sequences of eukaryote are much longer than those of prokaryotes and the gene structure is complex. By elucidating the mechanism of DNA, it is possible to solve hereditary diseases and to develop new medicines.

In 1988, the Genome Analysis Project began; the purpose of this project was to determine the complete DNA sequence of a creature and analyze it. Recently, the genome database has increased rapidly as computer performance has progressed. As the size of the genome database has grown, the development of gene finding methods, which are the techniques of predicting coding regions as gene candidates automatically, has been required. Recently, gene finding methods with Hidden Markov Models (HMMs) have been successful. HMM is a probabilistic model and is advantageous for the understanding of the structure of DNA probabilistically. Today, several gene finding programs using HMMs have been developed and are in practical use for prokaryotic DNA [1].

The purpose of this research is to develop a gene finding program that predicts the gene-coding region from long DNA sequences using HMMs, to model optimal HMMs for gene finding, and to evaluate the performance by testing how many genes can be found. In making the program, the Viterbi algorithm was used. The Viterbi algorithm is an algorithm to find the optimal state path from the symbol sequence through the HMM. For the tests, the genome data file of Escherichia coli (E.coli) K-12 was used as one of the typical prokaryotic genomes. The genomic files including E.coli can be downloaded on the GenBank ftp server [2] [3].

This paper is organized as follows. Basic knowledge of DNA and gene structure of prokaryotes is in Section 2. The method for finding genes using HMM is in Section 3. The experiments for gene finding and these results are in Section 4. The conclusion is given in Section 5.

2 Basic Knowledge of DNA

2.1 DNA and Amino Acids

A DNA sequence can be represented as a sequence of four nucleotide letters: A, T, C, and G. When part of sequence is coded as a gene, codons in the sequence, the DNA nucleotide triplets such as "TTT," "GCG," and "ACC" are translated to amino acids letters. There are 64 (4^3) kinds of codons and 20 kinds of amino acids; several codons correspond to a common amino acids. The table of correspondence between codons and amino acids is shown if Table 1. For example, a DNA sequence "CTCGGAGTTACC" can be translated into the amino acid sequence "LGVT" by referring to the table.

TTT	F	TCT	S	TAT	Y	TGT	C
TTC		TCC		TAC		TGC	
TTA	L	TCA		TAA	STOP	TGA	STOP
TTG		TCG		TAG		TGG	W
CTT	L	CCT	P	CAT	H	CGT	R
CTC		CCC		CAC		CGC	
CTA		CCA		CAA	Q	CGA	
CTG		CCG		CAG		CGG	
ATT	I	ACT	T	AAT	N	AGT	S
ATC		ACC		AAC		AGC	
ATA		ACA		AAA	K	AGA	R
ATG	M	ACG		AAG		AGG	
GTT	V	GCT	A	GAT	D	GGT	G
GTC		GCC		GAC		GGC	
GTA		GCA		GAA	E	GGA	
GTG		GCG		GAG		GGG	

Table 1: Codon table

2.2 Gene Structure in Prokaryotes

The structure of genes in prokaryotes is very simple. As given in Figure 1, there are gene-coding regions in

prokaryotic DNA sequences. Looking at a coding region in detail, the coding region is a sequence of codons, and starts with one start codon such as ATG, TTG, and GTG, and continues with regular codons (excluding stop codons such as TAG and TGA), and finally ends with one stop codon [4].

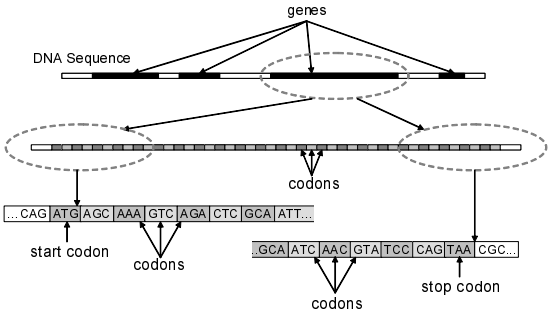


Figure 1: Structure of genes in prokaryotes

2.3 Open Reading Frames

To find a coding region, the first thing to do is to find the region starting with a start codon, continuing with regular codons, and ending with a stop codon. Such a region is called an Open Reading Frame (ORF). Finding ORFs is very easy, but there are many ORFs that are not coding regions; these ORFs are called NORFs (non-coding ORFs). Furthermore, there are overlapping ORFs which have a common stop codon as in Figure 2. This problem makes it difficult to find genes by just finding ORFs.

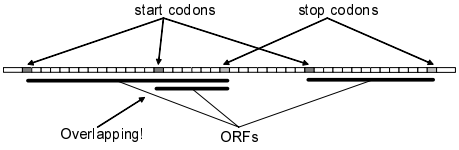


Figure 2: ORFs and overlapping ORFs

3 Gene Finding Methods with Hidden Markov Models

3.1 Hidden Markov Models (HMMs)

HMM is a graphical probabilistic model, and it is useful to model the inner structure of sequence. There are the concepts of state and symbol. The definition and the description of HMM are written as follows [5].

3.1.1 The Definition of HMM

Sequence of state, which is called path π, is not observable. It follows a simple Markov Chain. Let the ith state of path π_i, and then the transition probability of the Markov chain of path is defined as follows:

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k). \quad (1)$$

Each state emits a symbol from a distribution of all possible symbols. The probability that symbol b will be observed at state k is defined as follows:

$$e_k(b) = P(x_i = b | \pi_i = k). \quad (2)$$

This parameter is known as emission probability. If the state path π emitting symbol sequence x is known, it is possible to calculate the joint probability of π and x. It is defined as follows:

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{a_{\pi_i \pi_{i+1}}} \quad (3)$$

3.1.2 Example of HMM: The Occasionally Dishonest Casino

In a dishonest casino, they use a fair die most of the times, but occasionally switch to a loaded die by the probability of 0.05. Then they use a loaded die repeatedly, and switch back to a fair one by the probability of 0.10. When using a loaded die, the probability of 6 is 0.5 and the others are 0.1. The HMM for the dishonest casino can be written down like Figure 3:

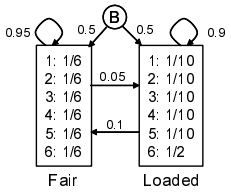


Figure 3: HMM for dishonest casino

Fair (F) and Loaded (L) states are represented as states and follow a Markov Chain. The probabilities of the number of spots of each die are represented as emission probabilities and are written in the boxes (states).

For example, suppose that someone rolled dies in the order of “FFLL,” and the results of rolls are ”1266.” Then, the joint probability of states and symbols (rolls) through this HMM is as follows:

$$\begin{aligned} P(x,\pi) =& a_{BF} \times e_F(1)a_{FF} \times e_F(2)a_{FL} \\ & \times e_L(6)a_{LL} \times e_L(6)a_{LE} \\ =& \frac{1}{2} \cdot \frac{1}{6} \cdot 0.95 \cdot \frac{1}{6} \cdot 0.05 \cdot \frac{1}{2} \cdot 0.9 \cdot \frac{1}{2} \cdot 1 \\ =& 1.48438 \times 10^{-4} \end{aligned}$$

3.2 The Most Probable Path: Viterbi Algorithm

It is impossible to know that what states emit the symbols by just looking at the observed symbol sequence. But it is important to discover the state sequence in order to understand the structure of symbols. There are a number of approaches to predict the most probable state path, and the most common one is called the Viterbi Algorithm [6].

Generally, many state sequences may emit a common sequence of symbols. In the example of the HMM for the dishonest casino, the observed sequence (1, 2, 6, 6) may be emitted by three different state paths: (F, F, F, F), (L, L, L, L), and (F, L, F, L). Considering the three joint probabilities of the observed sequence and the state path in the casino's HMM, it is most likely that the symbol sequence 1266 came from the second state path, because the loaded die has a high probability of emitting 6. Thus, a predicted state path can be found by calculating the joint probability. The predicted state path with the highest probability is defined as follows:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, P(x,\pi).$$

The most probable state path π^* can be calculated by recursion. Suppose the Viterbi probability $v_k(i)$ of the most probable path ending in state k with the ith observation is known for all the states k. Then the value $v_l(i+1)$ of observation $i+1$ can be calculated by recursion:

$$v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i)a_{kl}).$$

In the Viterbi algorithm, the sequence starts in state 0 (the beginning state) at observation 0. Then for i=1…L, the values of v are calculated and the pointer *mathrmptr* holds the previous states with the highest probability. Finally, the highest probability of the whole sequence is calculated and the most optimal state path can be found by tracing back the pointers from the tail. The full algorithm is as follows.

Algorithm: Viterbi

Initialization(i=0):	$v_0(0) = 1, v_k(0) = 0 \, for \, k > 0.$
Recursion(i=1…L):	$v_l(i) = e_l(x_i) \max_k (v_k(i-1)a_{kl})$;
	$\mathrm{ptr}_i(l) = \mathrm{argmax}_k (v_k(i-1)a_{kl})$.
Termination:	$P(x,\pi^*) = \max_k (v_k(L)a_{k0})$;
	$\pi^*_L = \mathrm{argmax}_k (v_k(L)a_{k0})$.
Traceback(i=L…1):	$\pi^*_{i-1} = \mathrm{ptr}_i(\pi^*_i)$.

For example, Figure 2 is the table of $v_l(i)$ using the symbol sequence x (1,2,6,6) through the casino's HMM. By the Viterbi algorithm, the most probable path π^* of the symbol sequence (1,2,6,6) is (L,L,L,L).

l \ x_i	B=0	1	2	6	6
B=0	1	0	0	0	0
F	0	0.083	0.013	0.0023	0.00033
L	0	0.05	0.0045	0.0020	0.0003

Table 2: Values of $v_l(i)$

4 Experiments

The genome file of E.coli O157:H7 was used for the training to model HMMs. Also, the genome file of E.coli K-12 MG1655 was used for the experimental test. We conducted some experiments with four types of HMMs. The HMM in the first experiment is a simple model as an evaluative standard, and the others are probabilistically extended HMMs.

4.1 Experiment 1: Viterbi Algorithm with HMM for Codons

4.1.1 HMM for Codons

The organization of genes can be classified into four parts: non-coding regions, start codons, coding regions, and stop codons. First, the DNA sequence starts from the part of the non-coding region and continues with a meaningless sequence of codons. After that, the sequences come to the gene-coding region. As mentioned in section 2, this part starts with a start codon, continues with regular codons, and ends with a stop codon. Then the sequence returns to a non-coding region. Thus, this cycle loops with the number of genes. We can write the HMM for codons as Figure 4.

This HMM has four states and each state has 64 symbols. Each state emits one codon and has a different distribution of probabilities of emission.

4.1.2 Experimental Results of Exp. 1

The results of Exp. 1 are shown in Table 3. ORFs found are all gene candidates detected by the Viterbi Algorithm. They can be classified to three types: genes, overlapping ORFs, and irrelevant NORFs. Genes have

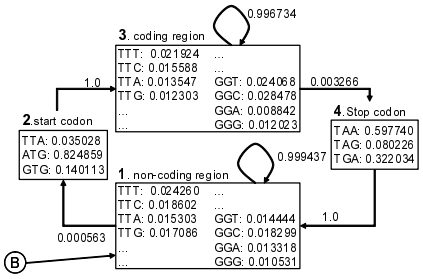
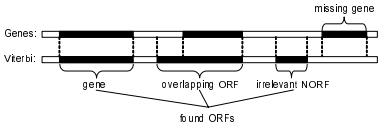


Figure 4: HMM for codons in experiment 1

	Exp. 1
found ORFs	5760
genes	2907
overlapping ORFs	1119
irrelevant NORFs	1734
missing real genes	262/4288
Sn (Sensitivity)	93.88(%)
Sp (Specificity)	69.89

Table 3: Results of Experiments 1

the same information of the start and stop positions of coding regions as that of the tested DNA. Overlapping NORFs have stop codons in common with real genes. Irrelevant NORFs have no information in common with real genes. Missing genes are real genes ignored by our algorithm.



As the performance of gene finding, there are two values: sensitivity and specificity. Sensitivity is the percentage of gene candidates that are found, and specificity is the percentage of how correctly they are found. These two values are defined as fallows.

$$\text{Sn} = \frac{\text{genes + overlapping ORFs}}{\text{all genes}}$$

$$\text{Sp} = \frac{\text{genes + overlapping ORFs}}{\text{found ORFs}}$$

Looking at the result, more than 90 percent of proper gene candidates were found. However, the HMM in this experiment could not distinguish genes and overlapping ORFs. In addition, there are many irrelevant NORFs, so the specificity was not good performance.

4.2 Experiment 2 to 4: HMMs for Markov Chain

To model a highly accurate HMM that can find more genes than that of Experiment 1, we considered to adding the information of Markov Chains (MCs) to HMMs. There are three types of forms of this HMM as given below.

4.2.1 Experiment 2: HMM for Markov Chain of Codons

First, we convert a codon sequence to a sequence of codon MC. Then, one symbol has the information of two consecutive codons. To label the states for HMM, we have to add extra states to start and stop codons as in Figure 5 because the data of start and stop codons is in two consecutive symbols of the MC. Then, we can write a HMM of the Markov Chain for the codons as in Figure 6. Each state emits one codon as a symbol, but the emission probabilities vary with the last codon of the emitted symbol sequence. This HMM has 6 states and 4096 (64^2) symbols.

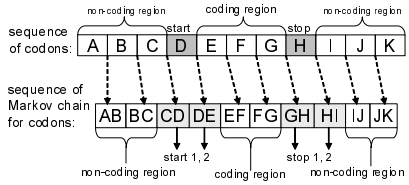


Figure 5: Gene structure of Markov chain for codons

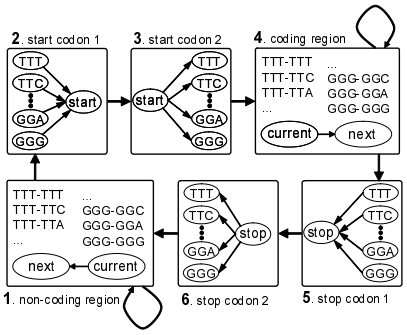


Figure 6: HMM for codon Markov chain

4.2.2 Experiment 3: Simple Markov Model for Markov Chain of Codons

There is another form of MC as shown in Figure 7. This HMM is classified into four parts as in Exp. 1, and in each part, one codon is represented as a state which emits its own codon with probability 1. In coding and non-coding regions, the transitions of states form simple Markov models. The number of states of this HMM is 134 and the number of emitting symbols is 64.

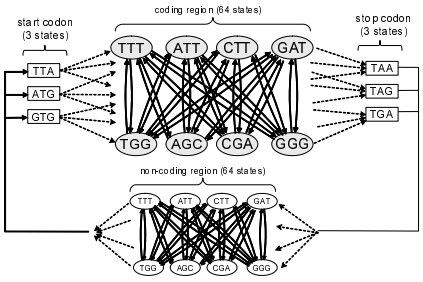


Figure 7: Simple Markov model for Markov chain of codons

4.2.3 Experiment 4: HMM for High Order Markov Chain of Nucleotides

In previous experiments, the codon was an emitting symbol, but in Exp. 4, a nucleotide is an emitting symbol. As shown in Figure 8, the HMM is classified into six parts as the HMM in Exp. 2. Each part has three states and each state emits one of four nucleotides as a symbol.

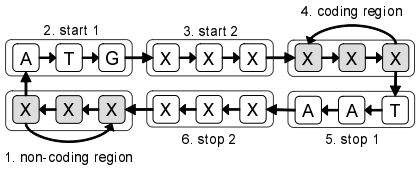


Figure 8: HMM for MC of nucleotides

Distribution of emitting probabilities of symbols varies according to the emitted symbols. In the case of the 2nd order, emitting probabilities of symbols depend on the last two emitted nucleotides (symbols). A HMM of the Nth order MC has $N + 1$ parameters of probabilities. For example, a probabilities table of a 2nd order Markov Chain is shown in Table 4. The number of states of HMM of the Nth order MC is 18 and the number of emitting symbols is 4^{N+1}.

	T *	C	A	G
TT *	0.23	0.5	0.07	0.2
TC	0.3	0.6	0.01	0.09
TA	0.1	0.3	0.5	0.1
...	...	...	...	...
GG	0.3	0.02	0.4	0.28

Table 4: Example of emission probabilities of 2nd order MC

4.2.4 Experimental Results: Exp. 2 to 4

Table 5 shows the experimental results of all experiments. There are five results of Exp. 4 using HMMs having second to sixth order MCs. Looking at the results of Exp 2 and 3, both performed almost the same and greater than that of Exp. 1. However, comparing Exp. 2 and 3, the processing time of Exp. 3 is much longer than that of Exp. 2 because the quantity of computing in the Viterbi algorithm is proportional to the square of the number of states in HMM.

As the order increases in Exp. 4, irrelevant NORFs are reduced drastically, so the specificity is raised to 95.90%. However, the sensitivity of the 6th order is lower than the other orders. The processing time is approximately 2600 seconds (about 44 minutes) and is a reasonable time compared to the time of the other experiments.

4.2.5 Consideration

From the experimental results, it is possible to reduce irrelevant NORFs by modeling a HMM having the information of Markov Chains. The processing time with a HMM which is simply modeled as in Exp. 2 is shorter than with one which is modeled in a complicated way and has many states as in Exp. 3. Therefore, in order to model a highly accurate HMM, it is necessary to increase the order of the Markov Chain in a HMM, but the number of states should be small in order to keep the processing time shorter.

Comparing the results of Exp. 2 and 4, the probabilistic relation between nucleotides is stronger than that between codons. By increasing the order, it is possible to reduce more irrelevant NORFs, but not to discover more proper gene candidates including overlapping ORFs. Actually, less proper gene candidates in the 6th order are found than in the other orders. It is considered that this problem is caused by the lack of samples for training HMM. The number of emission parameters in the 6th is 16,384 (4^7), and the number of genes for

*The symbols on the vertical line are the last two symbol sequences of emitted symbols, and ones on the horizontal line are the symbols on the next emission.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4				
				2nd	3rd	4th	5th	6th
found ORFs	5760	4683	4674	5551	5153	4761	4501	3955
genes	2907	2939	3009	2998	2942	3000	3052	3071
overlapping ORFs	1119	1125	1071	1088	1110	1080	1000	722
irreverent NORFs	1734	619	594	1465	1101	681	449	162
missing real genes	262/4288	224	208	202	236	208	236	495
Sn (Sensitivity)	93.88(%)	94.77	95.14	95.28	94.49	95.14	94.49	88.45
Sp (Specificity)	69.89	86.78	87.29	73.60	78.63	85.69	90.02	95.90
number of states	4	6	134	18	18	18	18	18
processing time	42.4(sec)	102.3	48195	2647	2642	2585	2652	2674

Table 5: Results of all experiments

parameter estimation in this research is 5360. Since the probabilities in the start and stop codon part are calculated by only 5360 times, it is not enough to estimate the parameters of HMM by one DNA. Therefore, in order to model a highly accurate HMM, more samples of genome data will be essential.

5 Conclusion and Future Works

This paper presented a gene finding method and some HMMs for it. HMMs can be modeled flexibly, and the Viterbi algorithm can apply to HMMs which differ from the official definition such as a HMM having the information of Markov Chain of symbols. This research succeeded in discovering proper gene candidates almost correctly with a rating of more than 95 percent. However, the HMMs in these experiments could not distinguish between real genes and overlapping ORFs completely. Future work for this research includes two things. First, it is necessary to consider the inner structure of genes in further detail in order not to mistake overlapping ORFs as real genes. The second is to apply the HMMs of high order MCs to gene finding in eukaryotic genomes that have more complex structure than prokaryotic genomes.

References

[1] Akihiko Konagaya, *Idenshi To Computer*, KYORITSU SHUPPAN CO., LTD., 2000.

[2] "Complete Genomes in KEGG," Kyoto Encyclopedia of Genes and Genomes, http://www.genome.ad.jp/kegg/java/org_list.html (current Dec. 2002).

[3] "GenBank Database FTP Server," National Center for Biotechnology Information, ftp://ftp.ncbi.nih.gov/genbank/ (current Dec. 2002).

[4] Richard Durbin, Seqn R. Eddy, Anders Krogh, Graeme Mitchison, *Biological sequence analysis*, Cambridge University Press, 1998.

[5] Michael Zuker, "Hidden Markov Models," Rensselaer Polytechnic Institute, http://www.rpi.edu/~zukerm/MATH-4961/hmm/node1.html (current Dec. 2002).

[6] "Viterbi Algorithm," University of LEEDS, http://www.scs.leeds.ac.uk/scs-only/teaching-materials/HiddenMarkovModels/html_dev/viterbi_algorithm/s1_pg1.html (current Dec. 2002).