Linköping Electronic Articles in
Computer and Information Science
Vol. 6(2001): nr 17

Finding Genes from DNA Sequences by Using Multi-Stream Hidden Markov Models

Kiyoshi Asai

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/2001/017/

Published on August 30, 2001 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Typeset by the author using LaTeX
Formatted using étendu *style*

Recommended citation:
<Author>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. 6(2001): nr 17. http://www.ep.liu.se/ea/cis/2001/017/. *August 30, 2001.*

This URL will also contain a link to the author's home page.

Abstract

It is necessary to integrate various types of information in order to recognize the genes in DNA sequences. The information might be the four letters of DNA sequences, stochastic frequency of the corresponding codons, homology search scores, splice cite signal strengths. We have developed a software system of multi-stream Hidden Markov Models¡¡ with which those information can be easily integrated with a consistent measure of probabilities. The output symbols of HMMs are the signals that we can observe. In the field of bioinformatics, the output symbols of the HMMs are mostly the four letters of nucleic acids or the twenty letters of amino acids. However, the output symbols can be anything, as far as we can attach their probability distributions. They can be discrete symbols, real values, real valued vectors, and multiple streams of those values. We propose gene annotation using HMMs with multiple streams, which combine the sequence and other pre-processed information. The important feature of multi-stream HMMs is that the weights of the streams can be optimized for each model. The multi-stream HMM with adjustable weight permits very flexible design of the gene finding systems. (URL: http://www.genedecoder.org/)

Author's address

Computational Biology Research Center
National Institute of Advanced Industrial Science
and Technology
Tokyo, Japan

E-mail: `asai@chrc.jp`

1 Introduction

The progress of the sequencing projects, including the final stage of human genome, have been producing large sequence data. The next stage is to locate the genes and find their functions. The exact locations of the genes and the splicing patterns are proved by experiments, but the experiments are time-consuming. Computational systems for gene finding are widely used, although the accuracy of the finding for eukaryote genome is not satisfactory.

The genome of eukaryotes consist of several chromosomes, and the genes are distributed in those chromosomes. The genes have the structures of exons and introns. Coding exons are translated into amino-acid sequences, and the introns are spliced out before the translation. It is relatively easy to locate the approximate positions of the genes, but it is very difficult to determine the exact boundaries of the exons/introns.

Because genes have a structure like a language, linguistic methods are effective in order to understand their structure [10]. However, the components and the rules of the DNA behave as though non-deterministic, it is necessary to combine the statistics and the linguistics for the parsing of DNA. That is why hidden Markov models (HMMs) have been widely used for gene finding [6, 7, 8, 9, 11, 12].

A stochastic model is a signal source, which produces signals according to probabilistic mechanisms. For example, the first order stationary Markov model of the four nucleotide acids (A,C,G,T) produce DNA like sequences according to their 16 transition probabilities. In such a case, the states of the Markov model are the output symbols, and the probabilistic relations of the symbols are directly expressed by the parameters (i.e. transition probabilities). In most cases, however, it is not true to identify the direct relations of the symbols.

Such relations are mostly observed in more abstract concepts, but they cannot be observed directly. Genes have exon/intron structures, that we cannot directly observe from DNA sequences. Some complicated mechanisms identify which part of the DNA sequences are exon/intron, but the 'labels' are not written in the DNA sequences. The structures of exon/intron are the 'hidden existence,' that have distinct meanings in biology, but those 'hidden existence' are not explicitly observed in DNA sequences. The stochastic features of those 'hidden existence,' however, are very different between exons and introns (Therefore, it is an easier problem to identify exons and introns if the segmentations are given).

This situation is exactly what the stochastic models are assuming. In hidden Markov models, the 'hidden existence' that we cannot observe are called hidden states, and the observed DNA sequences are called outputs. The stochastic mechanism of the transitions of the hidden states are called state transition models. If we adopt the Markov process as the state transition models, the stochastic model is the hidden Markov model.

The authors have proposed a gene finding system, the GeneDecoder, which is built by using hidden Markov models (HMMs) and stochastic parsing [4]. Recently, a new idea of using multi-stream HMMs has been combined to the GeneDecoder. In this paper, we discuss the usefulness of multi-stream HMMs for gene finding and introduce the new version of the GeneDecoder.

2 Gene finding by using hidden Markov models

In order to build a stochastic model of the genes, we model the components of the gene structure by HMMs and connect them by the rules that represent the gene structure. From a viewpoint of stochastic grammar, an HMM is a stochastic regular grammar. The networks of the symbols can express regular grammar. A nice feature of regular grammars is their modularity. A network of the networks that represent regular grammars becomes a regular grammar. HMMs have the same property: a network of the networks that represent HMMs becomes an HMM. If we model the promoters, exons, introns, amino acids, and other objects on DNA by HMMs, the networks of these objects form a new HMM. This means we can parse the whole DNA sequence by the combined models using the Viterbi algorithm.

2.1 GeneDecoder overview

The authors have been developing the GeneDecoder, a gene finding system for eukaryote genomes. The GeneDecoder consists of the following software and the models.

- HMM definitions. They correspond to 'phonemes' of DNA language.
- A dictionary. It defines 'words' of DNA language as the sequence of 'phonemes' (i.e. HMMs).
- A grammar. It defines stochastic network of 'words'.
- Parsing software. The current implementation uses MultiHmmParse, which the authors have developed.

An overview of the system is shown in Figure 1. The diagram correspond to the grammar, and each component of the diagram corresponds to a word. The original version of GeneDecoder only use single-stream HMMs, which produce the symbols of 'A','C','G','T'. The current version, which use multi-stream HMMs, has several streams that represent various aspects of DNA sequences. Those streams are specifically used in each model. The state transition normally begins with intergenic model, then to the start codon, first codon, internal codons. The reading frames are considered by three types of exon fragments at donor/acceptor sites. The intron model behaves differently in three contexts, depending on how many bases are placed in the fragment of the exon at the donor site. For example, if the number of fragment bases are two at donor site, the intron model behaves as type two, which is followed only by the fragment of one base at the acceptor site. If there is a fragment of one base at the donor site, a fragment of two bases at the acceptor site.

2.2 Exon model

The exon model basically consists of internal-codon, donor/acceptor fragments each of which has three types. The internal-codon model consists of one HMM, the codon bi-gram. The di-codon usage is known as an important statistics[11]. In the original single-stream implementation, it consists of 61

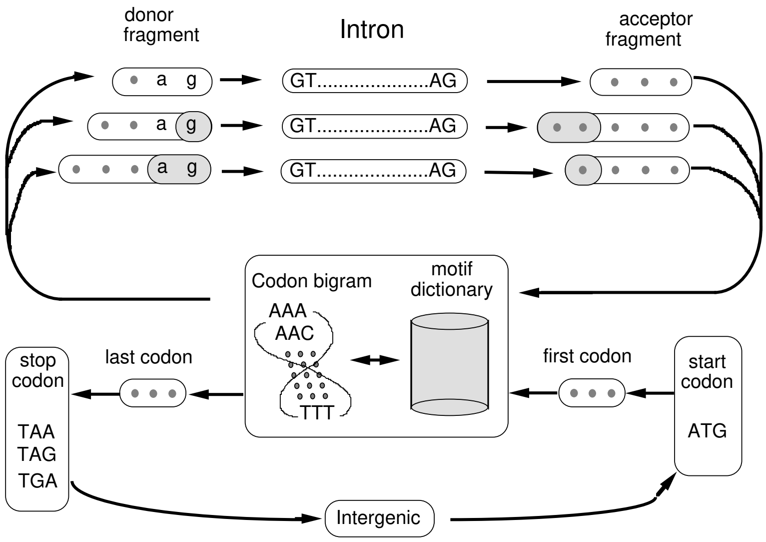

Figure 1: Overview of GeneDecoder

three state blocks, which correspond to internal codons, and these blocks are mutually connected for transition to form a first order Markov model of the codons. By using multi-stream HMMs, the di-codon model can be implemented by only three states. The detail will be discussed in the later sections. The donor/acceptor fragments have three types, according to the 'reading frames' of the exons. They could have the lengths of zero/one/two in base. However, because distributions of the base letters in these positions are very specific, the lengths are set to three/four/five in base.

2.3 Motif model

The motif models consist of motif entries in the dictionary, where each motif entry is a sequence of amino acid models (HMMs), which is similar to the regular patterns in PROSITE. Although each motif has different sequence of 'genetic phonemes,' they are all treated as one node in the stochastic grammar. This reduces the cost of the parsing in time. The purpose of using motif dictionary is to combine homology information into gene finding, and to find a clue to identify the functions of the genes. Different types of integration of homology information are discussed later.

2.4 Intron model and splice site

While the intron model behaves three ways, the intron model itself is same in all cases. It consists of intronhead, introncenter and introntail. Intronhead and the introntail are the fixed-length HMMs, which are equivalent of weight matrices of the probabilities, whose lengths are 7 and 17 respectively. Introncenter is a complex of a third order Markov model and a fixed-length HMMs that include branch point model. The third order Markov model is implemented as an HMM, by assigning different states for all second order contexts of the bases. In the original single-stream GeneDecoder, splice site statistics are distributed in the donor/acceptor fragment models of exons and the fixed length part of the intron models. These models are understood as position specific first order Markov models and context divided weight matrices.

2.5 Training

The training of the models has been performed simply calculating the statistics of the annotated regions of the training data set. Time-consuming Baum Welch training of the parameters of HMMs is not used.

2.6 Parsing

The recognition process is exactly the same as the dynamic programming parsing of the speech, using a grammar defined on these 'genetic words.' Having protein models with the system is different from the popular technique of homology search of the hypothetical coding regions. The latter searches the database *before or after* the system decides the candidate of the coding regions, while the former uses the information of the database *during* the process of deciding the candidate of the coding regions.

3 Multi-stream hidden Markov models

In a gene finding system, it is necessary to integrate various types of information in order to find the most likely structure of the genes. The information might be the four letters of DNA sequences, stochastic frequency of the corresponding codons, homology search scores, splice site signal strengths. We have developed a software system of multi-stream Hidden Markov Models¡¡ with which those information can be easily integrated with a consistent measure of probabilities.

The output symbols of HMMs are the signals that we can observe. In the field of bioinformatics, the output symbols of the HMMs are mostly the four letters of nucleic acids or the twenty letters of amino acids. However, the output symbols can be anything, as far as we can attach their probability distributions. They can be discrete symbols, real values, real valued vectors, and multiple streams of those values.

Because HMMs have some limitations to express the positional correlations of the bases, some of the models are often made by other methods, like artificial neural nets, and the generalized HMMs, which allow such non-HMM models behave as a part of stochastic parsing, have been used [9, 6]. We use multi-stream HMMs in order to combine the sequence and other pre-processed information.

There are two important features of our implementations of multi-stream HMMs.

- Direct valued streams. It is common to assume Gaussian or mixture of Gaussian as the distributions of the continuous streams of HMMs. However, the preprocessor often produce probabilistic confidence of pattern recognitions. In such a case, it is much more convenient that we can use the values of the stream as the probabilities themselves.

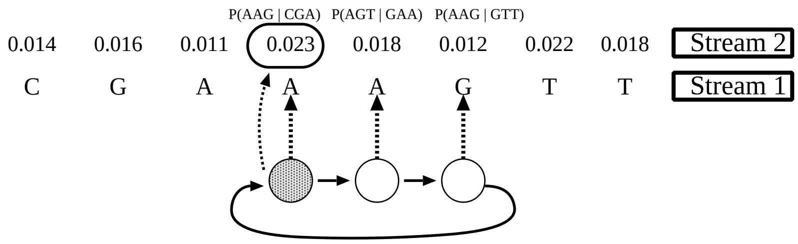

Figure 2: Dicodon model implemented by a multi-stream HMM

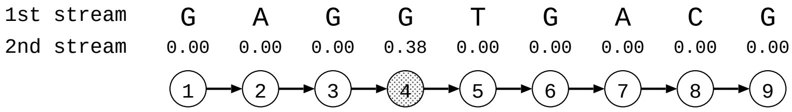

Figure 3: Donor model implemented by a multi-stream HMM

- State dependent weights of streams. The weights of the streams can be optimized for each model. The multi-stream HMM with adjustable weight permits very flexible design of the gene finding systems.

3.1 Example: Di-codon Model

Fig.2 shows that an multi-stream HMM works as the dicodon model. The conditional probability that each codon appears after the previous codon is pre-calculated, and the probability is put in the 'stream 2.' The multi-stream version of a di-codon model has only three states. The first state 'looks' at only the 'stream 2' and ignores the 'stream 1', the DNA sequence. The second and the third state do nothing, assign the probability 1 in any case. Comparing with the direct implementation of dicodon model without using multi-streams, the exactly same calculations are performed. The single stream version includes the conditional probabilities of the sequential codons as the transition probabilities between the codons. The multi stream version just fetch the value of 'stream 2' as the 'direct value' for the probability.

3.2 Example: Donor Site Model

Another example of multi-stream is shown in Fig.3. The pre-calculated probability as a donor site (exon/intron boundary) for each position is put in the '2nd stream.' The preprocessor for the calculation of those probabilities can be anything, position weight matrices, artificial neural nets, support vector machine, etc. While preprocessor can 'look' any length around the donor site, the model of donor site can be as short as one state.

4 Discussion and Conclusion

If the preprocessor for the multi-stream HMM is a position weight matrix, it is very easy to implement it as a hidden Markov model. The position weight

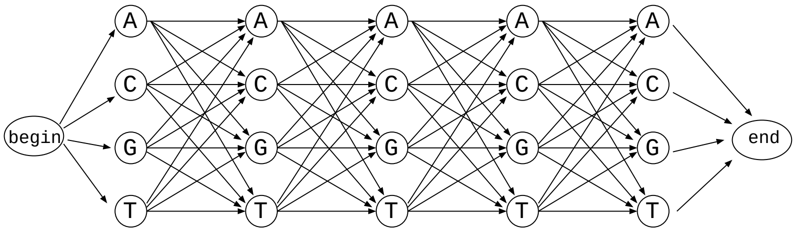

Figure 4: A position weight matrix conditioned by previous symbol and implemented by an HMM

matrix with conditional probabilities (conditioned by previous symbol) can also be implemented simply as a hidden Markov model (Fig.4). This model can also be understood as non-stationary first order hidden Markov model. In Genescan, one of the most widely used gene finding system, this type of model is used for the acceptor site (intron/exon boundary).

However, implementation of an arbitrary model as an hidden Markov model is impossible and unnecessary. Multi stream hidden Markov model can include any type of model as the preprocessor for the additional stream. One obvious and important implementation is adopting the homology search as a preprocessor. For example, the score of BLAST search can be converted into the direct stream, and the scores can be used an alternative coding potential instead of di-codon statistics. By adopting state dependent weights of the streams and the direct valued streams, multi-stream HMMs can combine any kind of additional information by preprocessing. In the field of genome research and in the study of pattern recognition, many new ideas of analysis, modeling, pattern recognition would be invented. They can be utilized flexibly and easily in gene finding by using the idea of multi-stream HMMs with the implementation discussed in this paper. The software can be downloaded from http://www.genedecoder.org.

References

[1] Asai,K.; Handa,K. and Hayamizu,S.: "Genetic Information Processing by Stochastic Model: HMM for Secondary Structure Prediction of Protein," *Genome Informatics,* **2**, 144-147 (in Japanese, 1991).

[2] Asai,K.; Hayamizu,S. and Onizuka,K.: "HMM with Protein Structure Grammar," *Proceedings of 26th HICSS*, **1**, 783-791 (1993).

[3] Asai,K.; Yada,T. and Itou,K.: "Finding Genes by Hidden Markov Models with a Protein Motif Dictionary," *Genome Informatics,* **7**, 88-97 (1996).

[4] Asai,K., Itou,K. and Ueno,K.: "Recognition of Human Genes by Stochastic Parsing," *Pacific Symposium on Biocomputing 1998.*

[5] Borodovsky,M. and McIninch,J.: "GENMARK: parallel gene recognition for both DNA strands," *Comp. Chem.* **17**, 123-133 (1993).

[6] Burge,C. and Karlin S.: "Prediction of Complete Gene Structures in Human Genomic DNA," *J.Mol.Biol.* **268**, 78-94 (1997).

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

[8] Krogh,A.: "Two methods for improving performance of an HMM and their application for gene finding," *ISMB97*, **5**, 179-186 (1997).

[9] Kulp,D.; Haussler,D.;Reese,M.G. and Eeckman,F.H.: "A Generalized Hidden Markov Model for the Recognition of Human Genes in DNA," *ISMB96*, **4**, 134-142, (1996).

[10] Dong,S. and Searls,D.B.: "Gene structure prediction by linguistic methods," *Genomics*, **23**, 540-551 (1994).

[11] Yada,T. et al: "Extraction of Hidden Markov Model Representations of Signal Patterns in DNA Sequences," *PSB96*, **1**, 686-696 (1996).

[12] Yada,T. and Hirosawa,M.: "Gene Recognition in Cyanobacterium Genomic Sequence Data Using the Hidden Markov Model," *ISMB96*, **4**, 252-260 (1996).