Global Journal of Computer Science and Technology: H
Information & Technology
Volume 22 Issue 1 Version 1.0 Year 2022
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Biological Analysis and Linear Block Hidden Markov Model for Gene and Labelled

By Dr. Suneel Pappala

Abstract- Hidden Markov models (HMMs) have been extensively used in biological sequence analysis. HMMs and their applications in a variety of problems in molecular biology. The difficulty of using computational approaches to discover genes in DNA sequences is yet unsolved. gene prediction from within genomic DNA are far from being powerful enough to elucidate the gene structure completely. We develop a hidden Markov model (HMM) to represent the degeneracy features of splicing junction donor sites in eucaryotic genes. he HMM system is fully trained using an expectation maximization algorithm and the system performance is evaluated using the 10-way cross-validation method. he HMM system is fully trained using an expectation maximization algorithm and the system performance is evaluated using the 10-way cross-validation method.

Keywords: hidden markov model (HMM), pair-hmm, profile-HMM, context-sensitive HMM (csHMM), profile-csHMM, sequence analys.

GJCST-H Classification: DDC Code: 572.8633 LCC Code: QP620

Biological Analysis and Linear Block Hidden Markov Model for Gene and Labelled

Dr. Suneel Pappala

Abstract- Hidden Markov models (HMMs) have been extensively used in biological sequence analysis. HMMs and their applications in a variety of problems in molecular biology. The difficulty of using computational approaches to discover genes in DNA sequences is yet unsolved. gene prediction from within genomic DNA are far from being powerful enough to elucidate the gene structure completely. We develop a hidden Markov model (HMM) to represent the degeneracy features of splicing junction donor sites in eucaryotic genes. he HMM system is fully trained using an expectation maximization algorithm and the system performance is evaluated using the 10-way cross-validation method. he HMM system is fully trained using an expectation maximization algorithm and the system performance is evaluated using the 10-way cross-validation method.

Keywords: hidden markov model (HMM), pair-hmm, profile-HMM, context-sensitive HMM (csHMM), profile-csHMM, sequence analys.

I. Hidden Markov Model

A *hidden Markov model (HMM)* is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. We call the observed event a `symbol' and the invisible factor underlying the observation a `state'. An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observable symbols. The hidden states form a *Markov chain*, and the probability distribution of the observed symbol depends on the underlying state. For this reason, an HMM is also called a doubly-embedded stochastic process.

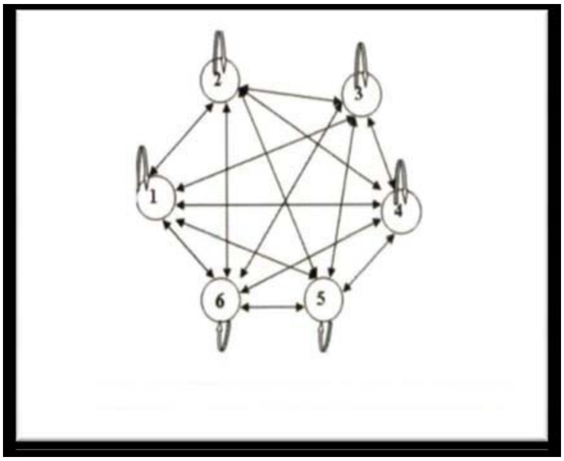


Figure 2.1: A Markov chain with 6 ststes (labelled 1 to 6)

II. Block-HMM for Labelled Sequences

Block-HMM restricts its search to a subset of HMM topologies made up of blocks of states. Each block is assigned a label that corresponds to one of the three secondary structure classes. The states that make up the blocks emit amino acid symbols. Secondary structure prediction is done by inferring the values of the hidden states for a given amino acid sequence, and examining the secondary structure labels of the blocks these states belong to. Four types of blocks are used: linear, self-loop, forward-jump blocks and zero blocks (figure 1).

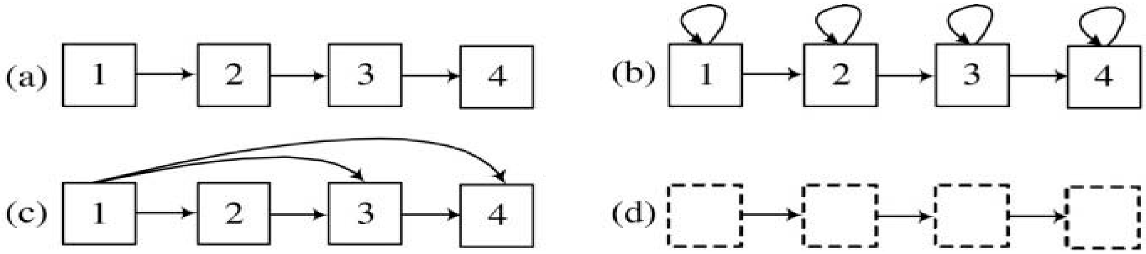


Figure 1

a) HMM blocks that compose the whole HMM structure

1. linear block
2. self-loop block (tying is optional)
3. forward-jump block (tying is optional)
4. zero block.

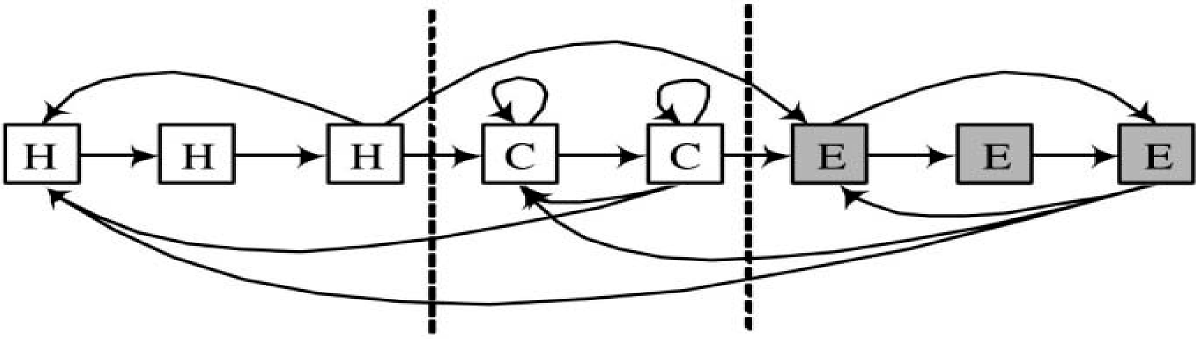
Linear blocks consist of *N* states (labelled from 1 to *N*) where state *n* is only connected to state $n + 1$ (with $1 \leq n < N$). Self-loop blocks are linear blocks in which each state has an additional loop to itself. A

Author: e-mail: suneelpappala@gmail.com

forward-jump block is a linear block where the first state is also connected to the last *M* states (with 1 <= *M* <*N*). Zero blocks are empty blocks with no states: they can replace other block types during the GA procedure and thus allow the exploration of simpler topologies.

The self-loop and forward-jump blocks can be either tied (in the figures, tied blocks are shaded) or untied. When a block is tied all the emission and transition probabilities of states inside the block are equal. In the case of linear blocks we did not consider tying because tying a linear blocks is equivalent to a single-state self-loop block.

The various blocks can model different types of sequence fragments. A linear block can model a particular conserved sequence pattern. The self-loop block can model a sequence of any length, while the forward-jump block can be used to represent subsequences with varying length up to some fixed length. Initially, the blocks are fully linked to form HMM architectures. In this context, fully linked means that the end state of each block is connected to the starting states of all other blocks and itself. Each block is labelled with one of the three protein structure classes 'H' (helix), 'E' (strand), or 'C' (coil). Figure 2 shows a simple example of HMM structure. The HMM structure is composed of 3 blocks. From the left it has blocks labelled with 'H','C' and 'E'. Each block also can be tied. After training, most of the transition probabilities are close to zero, resulting in a final structure that is typically much simpler than the fully connected HMM shown in the figure.



b) An Example if an HMM Composed of Blocks Resulting from the Block-HMM Procedure

Three blocks are used in this model and all the blocks are fully connected to each other. The blocks are divided by dotted lines. The states in tied blocks are shaded in grey.

III. Genetic Operators for Block-HMM

Genetic algorithms evolve a population of solutions with genetic operators. Inside the genetic cycle, genetic operators select members of the population (called parents) and evolve them to produce new members (called children). New children after the genetic operators along with the remaining old members in a population are evaluated to calculate fitness. According to the fitness selection procedure select a number of members in a population for the next genetic cycle.

We used three genetic operators in Block-HMM: crossover, mutation and type-mutation. The number of blocks is kept fixed but the number of the states of an HMM can be changed by the genetic operators. Crossover swaps a number of blocks in two parents to create two children. The crossover points and the number of blocks are chosen randomly. Figure 3 shows an example of the crossover scheme. The last block of the first child crosses with the first block of the second child. To simplify the diagram, transitions between blocks are not shown here. The crossover operator enables HMMs to exchange states without breaking basic blocks. Several blocks can be chosen to be crossed, which allows GA to search broad area of solution space. Mutations can take place inside any block of the HMM. A forward-jump block can have 6 different types of mutation, which are illustrated in figure 4. It can delete or insert.

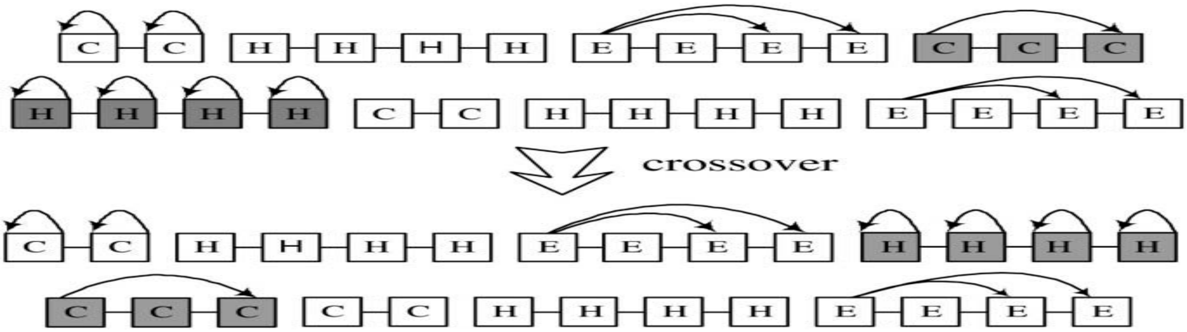


Figure 3

a) Crossover in Block-HMM

Crossover swaps the HMM states without changing the properties of an individual HMM block. Here, the last block of the first child crosses with the first block of the second child. To simplify the diagram, transitions between blocks are not shown.

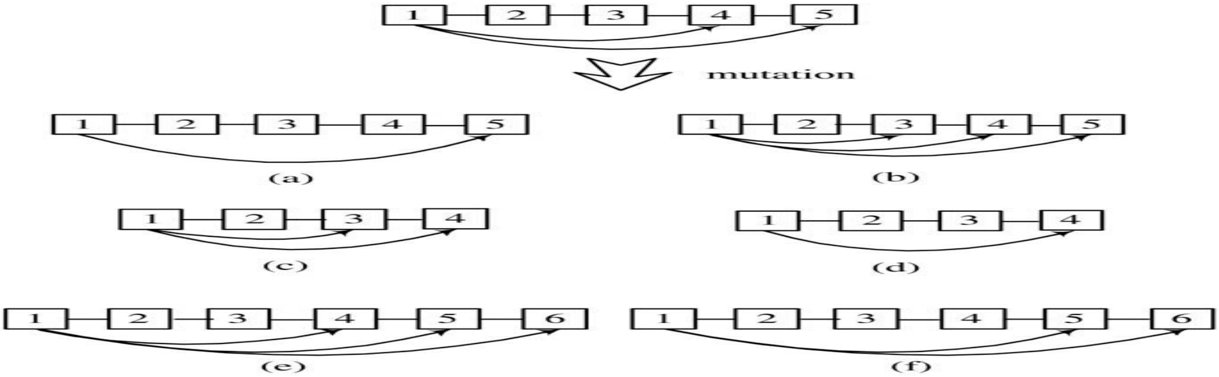


Figure 4

b) Mutation in Block-HMM

Six possible types of mutations from a 5-state forward-jump block: (a) a transition from the first to the fourth state is deleted (b) a transition from the first to the third state is added (c) the second or the third state is deleted (d) the fourth state is deleted (e) a state is added between the fourth and the fifth state (f) a state is added between the first and the fourth state.

In addition to changing the length of a block and its transitions, we also allow another form of mutation, called *type-mutation*, that changes the type or label of a block. Type-mutation to a zero block is also allowed (figure 5). When a type mutation transforms the type of a block, new transition probabilities are generated randomly. Self-loop and forward-jump blocks can type-mutate between tied and untied versions. Zero-blocks can be type-mutated to any of the other block forms.

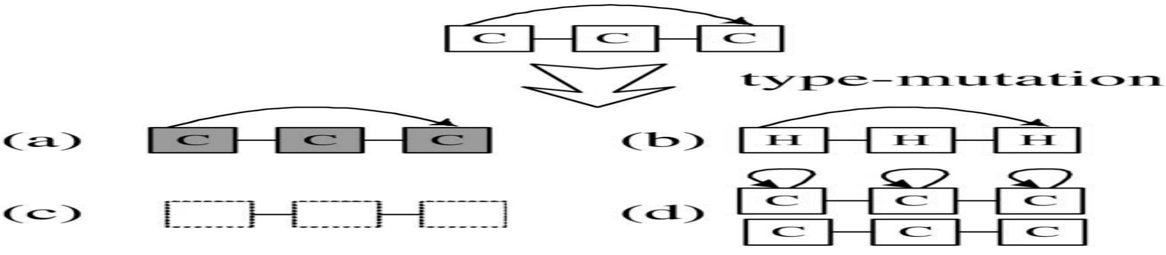


Figure 5

c) Type-Mutation in Block-HMM

A forward jump block is type mutated (a) to a tied block (b) to a block with a different label (c) to a zero block (d) to a self loop block or a linear block.

E ran the GA that hybridize the parameter learning method with these genetic operators that train the structure of HMMs. The detailed description of the whole procedure is on Methods.

IV. Analysis of the Evolved HMM

a) The Evolved Model

Figure 6 illustrates the structure of the best result of Block-HMMs. The simulation used 30 blocks, but the result shows only 26 blocks: the remaining 4 are zero blocks. Figure 7 shows the full HMM structure. Assigned with each state is one of the label of 3 states of secondary structure $l \in \{H, E, x\}$. It is composed of 22 states for helix (H), 15 for β-strand (E), and 15 for coil (x) region. Each state emits a set of symbols of 20 amino acids according to the given probability. The full HMM structure is trained using 1662 sequences (see Methods).

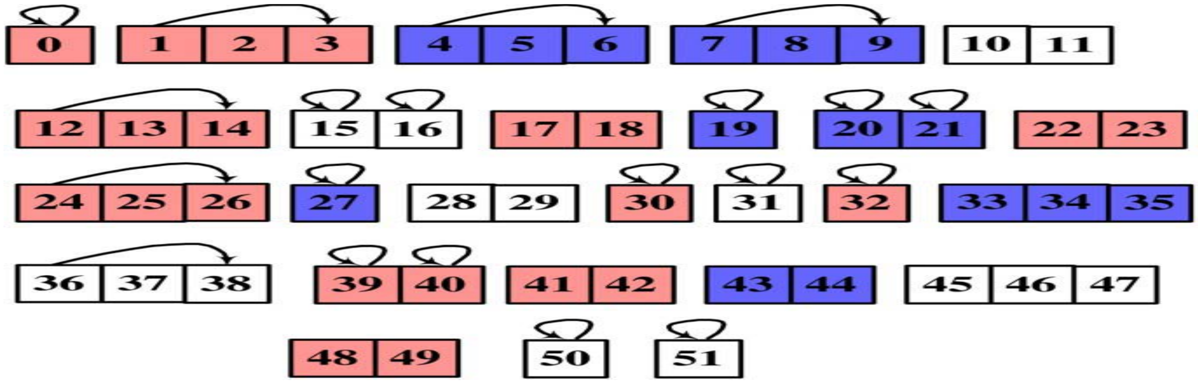


Figure 6

b) The Best HMM Topology

The best HMM topology evolved using Block-HMM. It is composed of 26 non-zero blocks and 52 states. Transitions between blocks are not shown here (including the transition from a block to itself). On each state a label is assigned ('H' for helices, 'E' for β-strands and 'x' for coils). Helix states are red colored and β-strand states are blue colored.

References Références Referencias

1. Lim VI: Algorithms for prediction of alpha helices and structural regions in globular proteins. J Mol Biol 1974, 88: 873–894. 10.1016/0022-2836(74)90405-7
2. Chow PY, Fasman GD: Prediction of the secondary structure of proteins from their amino acid sequence. Adv Enzymol 1978, 47: 45–148.
3. Garnier J, Osguthorpe DJ, Robson B: Analysis and implications of simple methods for predicting the secondary structure of globular proteins. J Mol Biol 1978, 120: 97–120. 10.1016/0022-2836(78)90297-8
4. Qian N, Sejnowski TJ: Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 1988, 202: 865–884. 10.1016/0022-2836(88)90564-5
5. Bohr H, Bohr J, Brunak S, Cotterill R, Lautrup B, Nørskov L, Olsen O, Petersen S: Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 1988, 202: 865–884. 10.1016/0022-2836(88)90564-5
6. Rost B, Sander C: Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 1993, 232: 584–599. 10.1006/jmbi.1993.1413
7. Jones DT: Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices. J Mol Biol 1999, 292: 195–202. 10.1006/jmbi.1999.3091
8. Baldi P, Brunak S, Frasconi P, Soda G, Pollastri G: Exploiting the past and the future in protein secondary structure prediction. Bioinformatics 1999, 15(11):937–946. 10.1093/bioinformatics/15.11.937
9. Pollastri G, Przybylski D, Rost B, Baldi P: Improving the Prediction of Protein Secondary Structure in Three and Eight Classes Using Recurrent Neural Networks and Profiles. Proteins 2002, 47: 228–235. 10.1002/prot.10082
10. Lin K, Simossis VA, Taylor WR, Heringa J: A simple and fast secondary structure prodiction method using hidden neural networks. Bioinformatics 2005, 21(2):152–159. 10.1093/bioinformatics/bth487
11. Hua S, Sun Z: A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach. J Mol Biol 2001, 308: 397–407. 10.1006/jmbi.2001.4580
12. Ward JJ, McGuffin LJ, Buxton BF, Jones DT: Secondary structure prediction with support vector machines. Bioinformatics 2003, 19(13):1650–1655. 10.1093/bioinformatics/btg223
13. Guo J, Chen H, Sun Z, Lin Y: A Novel Method for Protein Secondary Structure Prediction Using Dual-Layer SVM and Profiles. Proteins 2004, 54: 738–743. 10.1002/prot.10634
14. Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, Lipman D: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl Acids Res 1997, 24: 3389–3402. 10.1093/nar/ x25.17.3389
15. Cuff J, Barton G: Application of Multiple Sequence Alignment Profiles to Improve Protein Secondary Structure Prediction. Proteins 2000, 40: 502–511. 10.1002/1097-0134(20000815)40:3<502::AID-PROT170>3.0.CO;2-Q
16. Albrecht M, Tosatto S, Lengauer T, Valle G: Simple consensus procedures are effective and sufficient in

secondary structure prediction. Protein Eng 2003, 16(7):459–462. 10.1093/protein/gzg063

17. Asai K, Hayamnizu S, Handa K: Prediction of protein secondary structure by the hidden Markov model. Comput Appl Biosci 1993, 9: 141–146.
18. Yoshikawa H, Ikeguchi M, Nakamura S, Shimizu K, Doi J: Prediction of Protein Structure Classes and Secondary Structure by Means of Hidden Markov Models. Systems and Computers in Japan 1999, 30(13):13–22. Publisher Full Text 10.1002 /(SICI)1520-684X(19991130)30:13<13::AID-SCJ2>3.0..