Identification of Protein Coding Regions in Genomic DNA Using Fuzzy Cellular Automata

T. Srinivasan[a], P. kiran Sree[b] and C. Gautam Arjun[b]

[a]Assistant Professor, Department of Computer Science and Engineering, [b] Dept of C.S.E.
Sri Venkateswara College of Engineering, Sriperumbudur, India.
tsrini1959@yahoo.com, kiran.sreee@gmail.com, gautam.arjaun@gmail.com

ABSTRACT

Genes carry the instructions for making proteins that are found in a cell as a specific sequence of nucleotides that are found in DNA molecules. But, the regions of these genes that code for proteins may occupy only a small region of the sequence. Identifying the coding regions play a vital role in understanding these genes. In this paper we propose a Cellular Automata (CA) based pattern classifier to identify the coding region of a DNA sequence.CA is simple, efficient and produces more accurate classifier than that have previously been obtained for a range of different sequence lengths. Experimental results confirm the scalability of the proposed FCA based classifier to handle large volume of datasets irrespective of the number of classes, tuples and attributes. Good classification accuracy has been established.

Keywords:

Cellular Automata (CA), Classifier, Genetic Algorithm (GA), DNA, Decision Tree, Fuzzy Cellular Automata (FCA), Coding Regions, Fuzzy Multiple Attractor Cellular Automata (FMACA), Pattern Classifier.

1.0 INTRODUCTION

Many of the challenges in biology are now challenges in computing. Bioinformatics, the application of computational techniques to analyze the information associated with bimolecules on a large scale, has now firmly established itself as a discipline in molecular biology. Bioinformatics is a management information system for molecular biology. Bioinformatics encompasses everything from data storage and retrieval to the identification and presentation of features within data, such as finding genes within DNA sequence, finding similarities between sequences, structural predictions.

For better understanding of the specified objectives, we presented CA, FCA fundamentals in Section 2; Section 3 extensively covers a special class of FCA termed as Fuzzy Multiple Attractor CA (FMACA) (Toffoli, 1998), Section 4 presents the design of FMACA based pattern classifier (Uberbacher & Mural, 1991; Maji & Chaudhuri, 2004) as well as rule formation and chromosome representation. In Section 5, we address the problem of protein coding region identification (Chattopadhyay, et al., 2000; Toffoli, & Margolus, 1987) in DNA sequences. In order to validate the design of proposed model, experimental results are also reported in this section 6.

2. 0 CELLULAR AUTOMATA (CA) AND FUZZY CELLULAR AUTOMATA (FCA)

A CA (Blaisdell, 1983; Vichniac, 1994; Fickett, 1982; Flocchini, *et al*., 1982), consists of a number of cells organized in the form of a lattice. It evolves in discrete space and time. The next state of a cell depends on its own state and the states of its neighboring cells. In a 3-neighborhood dependency, the next state $qi\ (t + 1)$ of a cell is assumed to be dependent only on itself and on its two neighbors (left and right), and is denoted as

$$q_i(t + 1) = f\,(q_{i-1}(t), q_i(t), q_{i+1}(t)) \quad \text{-----------(1)}$$

where $q_i\ (t)$ represents the state of the i^{th} cell at t^{th} instant of time, f is the next state function and referred to as the rule of the automata. The decimal equivalent of the next state function, as introduced by Wolfram, is the rule number of the CA cell. In a 2-state 3-neighborhood CA, there are total 256 distinct next state functions.

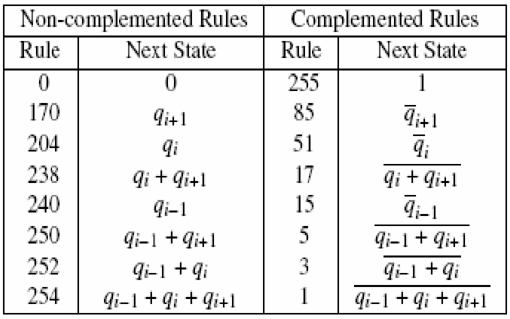
2.1 FCA Fundamentals

FCA (Langton, 2000; Flocchini, *et al*., 2000) is a linear array of cells which evolves in time. Each cell of the array assumes a state q_i, a rational value in the interval [0, 1] (fuzzy states) and changes its state according to a local evolution function on its own state and the states of its two neighbors. The degree to which a cell is in fuzzy states 1 and 0 can be calculated with the membership functions. This gives more accuracy in finding the coding regions. In a FCA, the conventional Boolean functions are AND, OR, NOT.

2.2 Dependency Matrix for FCA

Rules defined in equations like (1) should be represented as a local transition function of FCA cell. That rules are converted into matrix form for easier representation of chromosomes (Toffoli, 1998).

Table 2 FCA rules (complemented and non-complemented).



Non-complemented Rules		Complemented Rules	
Rule	Next State	Rule	Next State
0	0	255	1
170	q_{i+1}	85	$\overline{q}_{i+1}$
204	q_i	51	$\overline{q}_i$
238	$q_i + q_{i+1}$	17	$\overline{q_i + q_{i+1}}$
240	q_{i-1}	15	$\overline{q}_{i-1}$
250	$q_{i-1} + q_{i+1}$	5	$\overline{q_{i-1} + q_{i+1}}$
252	$q_{i-1} + q_i$	3	$\overline{q_{i-1} + q_i}$
254	$q_{i-1} + q_i + q_{i+1}$	1	$\overline{q_{i-1} + q_i + q_{i+1}}$

Example 1: A 4-cell null boundary (Definition 3) hybrid FCA with the following rule < 238, 254, 238, 252 > (that is, < (*qi*+*qi*+1), (*qi*-1+*qi*+*qi*+1), (*qi* + *qi*+1), (*qi*-1 + *qi*) >) applied from left to right, may be characterized by the following dependency matrix

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

While moving from one state to other, the dependency matrix indicates on which neighboring cells the state should depend. So cell 254 depends on its state, left neighbor, and right neighbor. Now we represented the transition function in the form of matrix. In the case of complement FMACA we use another vector for representation of chromosome.

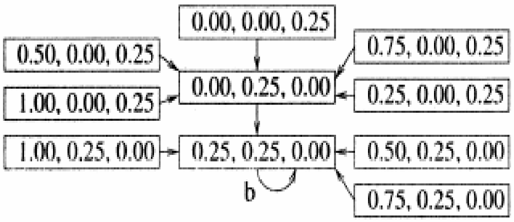
2.3 Transition from one state to other

Once we formulated the transition function, we can move form one state to other. For the example 1 if initial state is P (0) = (0.80, 0.20, 0.20, 0.00) ,
then the next states will be ,

P (1) = (1.00 1.00, 0.20, 0.20),
P (2) = (1.00 1.00, 0.40, 0.40),
P (3) = (1.00 1.00, 0.80, 0.80),
P (4) = (1.00 1.00, 1.00, 1.00).

3.0 FMACA BASED PATTERN CLASSIFIER

FMACA (Toffoli, 1998) classifies a given set of patterns into *k* distinct classes, each class containing the set of states in the attractor basin. A FMACA is a special class of FCA that can efficiently model an associative memory to perform pattern recognition classification task. Its state transition behavior consists of multiple components - each component, as noted in Fig. 1, is an inverted tree, each rooted on a cyclic state. A cycle in a component is referred to as an attractor. In the rest of the paper we consider only the FMACA having the node with self loop as an attractor state. The states in the tree rooted on an attractor form an attractor basin.



b is the attractor basin

Example 2: Let us have two pattern sets *S*1 =*{*(0.00,0.00, 0.25), (0.00, 0.25, 0.00), (0.25, 0.25, 0.00), (0.00,0.50, 0.00), (0.00, 0.00, 0.00), (0.25, 0.00, 0.00), (0.50,0.00, 0.00), (0.00, 0.00, 0.25), (0.00, 0.00, 0.75), (0.00,0.50,0.25)*}* (Class I) and *S*2 = *{*(0.75, 1.00, 0.00), (1.00,0.75, 0.50), (1.00, 1.00, 1.00), (0.75, 1.00, 1.00),(1.00,1.00, 0.75), (1.00, 0.75, 1.00), (0.50, 0.75, 1.00), (1.00,0.75, 0.75), (0.75, 1.00, 0.75), (0.75, 0.75, 1.00)*}* (ClassII) with three attributes.

In order to classify these two pattern sets into two distinct classes, Class I and II respectively, we have to design a FMACA such that the patterns of each class falls in distinct attractor basins.

When the FMACA is loaded with an input pattern say *P* = (1.00, 0.50, 0.00) and is allowed to run in autonomous mode, it travels through a number of transient states and ultimately reaches an attractor state (0.50, 0.50, 0.00) the attractor representing Class II.Here (0.00, 0.25, 0.00), (0.50, 0.50, 0.00) are attractor basins named b, d respectively.

4. 0 FMACA BASED TREE-STRUCTURED CLASSIFIER

Like decision tree classifiers (Flocchini, et al., 2000) FMACA based tree structured classifier recursively partitions the training set to get nodes (attractors of a FMACA) belonging to a single class. Each node (attractor basin) of the tree is either a leaf indicating a class; or a decision (intermediate) node which specifies a test on a single FMACA.

Suppose, we want to design a FMACA based pattern classifier to classify a training set $S = \{S1, S2, \cdot , SK\}$ into K classes. First, a FMACA with k-attractor basins is generated. The training set S is then distributed into k attractor basins (nodes). Let, S' be the set of elements in an attractor basin. If S' belongs to only one class, then label that attractor basin for that class. Otherwise, this process is repeated recursively for each attractor basin (node) until all the examples in each attractor basin belong to one class. Tree construction is reported in (Fickett, 1982). The above discussions have been formalized in the following algorithm. We are using genetic algorithm classify the training set.

***Algorithm 1**:* **FMACA Tree Building**

```
Input  :  Training set $S = \{S1, S2, \cdot\cdot, SK\}$
Output:  FMACA Tree.
Partition($S, K$)
Step 1: Generate a FMACA with $k$ number of attractor basins.
Step 2: Distribute $S$ into $k$ attractor basins (nodes).
Step 3: Evaluate the distribution of examples in each attractor basin (node).
Step 4: If all the examples (S') of an attractor basin (node) belong to only one class, then label the attractor basin (leaf node) for that class.
Step 5: If examples (S') of an attractor basin belong to $K'$ number of classes, then Partition (S', $K'$).
Step 6: Stop.
```

5.0 IDENTIFICATION OF PROTEIN CODING REGION IN DNA SEQUENCE

In this section we concentrate on application of FMACA to protein coding region identification (Maji & Chaudhuri, 2004; Farber, Lapedes, & Sirotkin, 1992). The idea of new method is to use the existing work of FMACA based tree structure classifier. Lot of research has been done for finding protein statistically. By using the standard codon frequencies, (Toffoli, 1998) we can identify whether the sequence contain protein coding regions or not.

Example 3:

Consider the sequence AGGACC
Since Codons will be in the form of triplets we split the input into three base sequences
So $P(S) = F$ *(*AGG*)* $\cdot F$ *(*ACC*)* = 0.22 * 0.38= 0.0836 using tables from, [11], [12].
In general, Let $F0(c)$ be the frequency of codon c in a non-coding sequence.
$P0\ (C) = F0\ (c1)\ F0\ (c2)\ldots F0\ (cm)$
Assuming the random model of non-coding DNA, F0(c) = 1/64= 0.0156 for all codons, P0 (S) = 0.0156 · 0.0156 = 0.000244. The log-likelihood (LP) ratio for S is LP(S) = *log (*0.000836/0.000244*)* = *log (*3.43*)* = 0.53.If LP(S) > 0, **S is coding.**

Like wise we can use Bayesian classifier to calculate the probability of finding the protein coding regions with accuracy up to 49. With our approach the average accuracy achieved is 75%.

5.1 Data and Method

The data used for this study are the human DNA data collected by Fickett and Tung. All the sequences are taken from GenBank in May 1992. Fickett and Tung Maji (2004) have provided the 21 different coding measures that they surveyed and compared.

The benchmark human data include three different datasets. For the first dataset, non-overlapping human DNA sequences of length 54 have been extracted from all human sequences, with shorter pieces at the ends discarded.

Every sequence is labeled according to whether it is entirely coding, entirely non-coding, or mixed, and the mixed sequences (i.e., overlapping the exon-intron boundaries) are discarded. The dataset also includes the reverse complement of every sequence. This means that one-half of the data is guaranteed to be from the non-sense strand of the DNA.

In the next section we will give the experimental results for finding this coding region for all sequence lengths.

6.0 EXPERIMENTAL RESULTS

The below tables shows the predictive accuracy of different algorithms on both coding and non-coding DNA sequences.

In this section we present the results on using FMACA for Fickett and Tung's dataset. Values are given for the percentage accuracy on test set coding sequences and the percentage accuracy on test set non coding sequences.

Table 3: Predictive Accuracy for length 108 human DNA Sequence

Algorithm	**Coding**	**Non Coding**
Dicodon Usage	61%	57%
Bayesian	51%	46%
FMACA	78%	72%

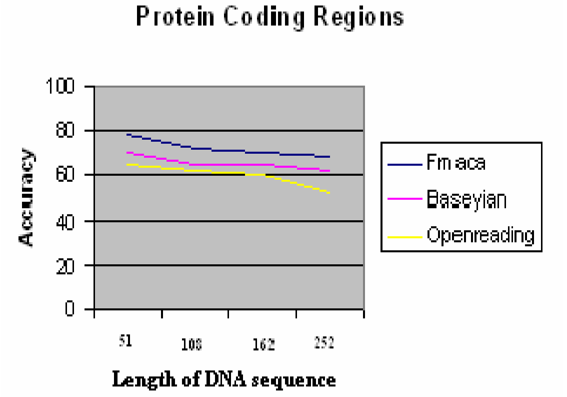
Table 4: Predictive Accuracy for length 108 human DNA sequence

Algorithm	**Coding**	**Non Coding**
Dicodon Usage	58%	50%
Bayesian	45%	36%
FMACA	74%	69%

Table 5: Predictive Accuracy for length 108 human DNA sequence

Algorithm	**Coding**	**Non Coding**
Dicodon Usage	65%	54%
Bayesian	50%	44%
FMACA	71%	70%

The graphs shows FMACA is comparable with other two. It shows that FMACA can be used to identify protein coding regions among all DNA sequence lengths. The accuracy reported also comparable with the others. The average accuracy reported is 75%. The data sets used are taken from Fickett and Tung collections. We trained the classifier using these dat sets and measured the accuracy for each individual feature.



This is the first algorithm to handle DNA sequence of length 252 and the time complexity is also drawn and it was also comparable with others.

FMACA overcome all the disadvantages of previous standard algorithms like fixing the position of the gene and static order of the DNA sequence.

7.0 CONCLUSION

This paper presents the application of FCA based pattern classifier to solve the problem of protein coding region identification in DNA sequences. Aside from developing a good classifier for this particular problem, the proposed model may be very much useful to solve many other bioinformatics problems like protein structure prediction, RNA structure prediction, promoter region identification, etc. .

8.0 ACKNOWLEDGEMENT

We are grateful to Dr R. Ramachandran, Vice Principal and Head of Department of Computer Science and Engineering, Sri Venkatewara College of Engineering, Sriperumbudur, India for his consistent encouragement and useful discussions. Authors are also thankful to the faculty of Department of Computer Science (SVCE) for their valuable suggestions. We also thank the Technical staff in our lab for their extended cooperation during the entire research work.

REFERENCES

Blaisdell, B. E. (1983). A prevalent persistent global non randomness that distinguishes coding and non-coding eukaryotic nuclear dna sequence. *J. Molec. Evol.*, 19, 122–133.

Langton, C.G. (2000). Self-reproduction in cellular automata. *Physica D*, 10, 135–144.

Uberbacher, E. & Mural, R. (1991). Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci.*, *USA*, 88, 11261–11265.

Vichniac, G. (1994). Simulating physics with cellular automata. *Physica D*, 10, 96–115, 1984.

Fickett, J. (1982). Recognition of protein coding regions in dna sequences. *Nucleic Acids Res.*, 10, 5303–5318.

Flocchini, P., Geurts, F., Mingarelli, A., & N. Santoro (2000),"Convergence and Aperiodicity in Fuzzy Cellular Automata: Revisiting Rule 90,"Physica D.

Maji, P. & Chaudhuri, P. P. (2004). FMACA: A Fuzzy Cellular Automata Based Pattern Classifier.

Proceedings of *9th International Conference on Database Systems*. Korea, 494–505, 2004.

Maji, P. & Chaudhuri, P. P. (2004). Fuzzy Cellular Automata For Modeling Pattern Classifier. *Accepted for publication in IEICE*.

Farber, R., Lapedes, A., & K. Sirotkin(1992). Determination of eukaryotic protein coding regions using neural networks and information theory. *J. Mol. Biol*., 226, 471–479.

Lippmann, R. (2004). An introduction to computing with neural nets. *IEEE ASSP Mag*., 4(22).

Chattopadhyay, S., Adhikari, S., Sengupta, S., &M. Pal .(2000). Highly regular, modular, and cascadable design of cellular automata-based pattern classifier. *IEEE Trans. Very Large Scale Integr. Syst*., 8(6).

Toffoli, T. & Margolus, N. (1987). Cellular Automata Machines, The MIT Press, Cambridge, MA.

Toffoli, T. (1998). Reversible computing. In De Bakker, J.W. & J. Van Leeuwen Automata, Languages and Programming, ed., (pp.632–644).