Implementation of DNA Pattern Recognition in Turing Machines

Sumitha C.H
Department of Computer Science and Engineering
Karunya University
Coimbatore, India
e-mail: sumithach@karunya.edu

Abstract—**Pattern recognition is the act of taking in raw data and taking an action based on the category of the pattern. DNA pattern recognition has applications in almost any field. It has applications in forensics, genetic engineering, bio informatics, DNA nanotechnology, history and so on. The size of the DNA molecules can be very large that it is a tedious task to perform pattern recognition for the same using common techniques. Hence this paper describes the pattern recognition for DNA molecules using the concept of Turing Machines. It also performs a simulation of the standard Turing Machine that performs DNA pattern recognition on the Universal Turing Machine.**

Index Terms— **DNA, DPRTM, delta rule, JFLAP, transitions, Turing Machine, UTM.**

I. INTRODUCTION

Pattern recognition is the act of taking in raw data and taking an action based on the category of the pattern. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space.

Pattern recognition can be done for any form of data that follows a specific pattern. The class of data includes images, binary data, decimal data and so on. This paper aims at performing the pattern recognition for DNA molecules using the concept of Turing Machines.

DNA molecules follow a specific pattern as they are composed of four constituents, Adenine, Thymine, Cytosine and Guanine. DNA pattern recognition involves feeding a specific DNA pattern and checking whether the pattern is present in the original DNA molecule. DNA pattern recognition has many applications in the fields of Genetic engineering, Forensics, Bioinformatics, DNA nanotechnology, History and anthropology and so on. The DNA molecules have a very long structure. It is very difficult to represent and manipulate it using common data structures. The pattern recognition involves input DNA pattern which can be even thousands of bit long and the original DNA structure will be much longer. A Turing Machine is a solution to this problem. A Turing Machine has infinite memory extendable in both directions. This paper implements the Pattern search for DNA molecules using the concept of automata and Turing Machines.

An automaton is a mathematical model for a finite state machine (FSM). A FSM is a machine that, given an input of symbols and transitions, jumps through a series of states according to a transition function. Turing Machines are the most powerful computational machines. They possess an infinite memory in the form of a tape, and a head which can read and change the tape, and move in either direction along the tape. Turing machines are equivalent to algorithms, and are the theoretical basis for modern computers. A Turing machine that is able to simulate any other Turing machine is called a Universal Turing Machine (UTM, or simply a universal machine).

A UTM is the abstract model for all computational models. A UTM T_U is an automaton that, given as input the description of any Turing Machine T_M and a string w, can simulate the computation of M on w. JFLAP represents a Turing Machine as a directed graph. In JFLAP, the simulation of a Turing Machine T_M in a UTM T_U is performed by providing as input the encoded string $<T_M ,w>$, where w is an input string for $T_{M.}$

The encoding is performed such that the string has three sections: list of final states, transitions of T_M and the tape contents of T_M prior to the start of execution. After each transition rule is simulated, T_U

Sumitha C.H,
Department of computer science and Engineering,
Karunya University,
Coimbatore, India
Phone: +91-9940 205532
Email: sumithach@karunya.edu

DOI: 10.5176_2010-2283_1.2.32

enters the final check section to determine if T_M has entered a final state.

II. DNA PATTERN RECOGNITION IN TURING MACHINES

The DNA molecule is composed of four constituents, Adenine, Thymine, Cytosine and Guanine. These constituents are represented with the letters A, T, C and G respectively. DNA pattern recognition involves feeding a specific DNA pattern and checking whether the pattern is present in the original DNA molecule. The four constituents of the structure, A, C, G and T are represented in binary format. The search involves providing a DNA pattern in the binary format and performing the operation on the Turing Machine which contains the original DNA structure in the binary format as well. This is encoded and given to the UTM along with the standard Turing Machine [1].

The DPRTM searches the original DNA pattern for the search pattern. The DPRTM algorithm works on the basis of an underlying linear search algorithm. The first symbol of the search pattern is marked and compared with the symbols in the original pattern. If the search becomes successful, the second symbol of the search pattern is marked and compared with the next symbol, where the symbol was first matched in the original pattern. This procedure continues till either the complete search pattern has been successfully matched in the original pattern or there is a mismatch in the symbols in the search pattern and the original pattern [4, 5].

The input to the UTM is encoded such that it contains three sections: (1) A list of the final states of TM , (2) The transitions of TM, (3) The tape contents of TM just prior to the start of execution. After each transition rule is simulated, T_U enters the final state check section to determine if TM has entered a final state.

A. *DNA Pattern Recognition*

The four constituents of the structure, A, C, G and T are represented in binary format. The binary representation is given in Table 1.

Table 1. Binary Representation of DNA constituents

DNA Constituent	**Corresponding Binary Screen**
A	00
C	01
G	10
T	11

For example a sample DNA sequence
TTAAGGACCCCATGCCCTCGAATAGGCTTGA
GCTTGCCAATTAACGCGCACGGCTGGCCG…
…
can be represented in binary format as
111100001010000101010100111001010111011000
0011001010011111100010011111100101000011110
0000110011001000110100111101001011 0………

The image of the structure of the DNA is obtained. This image is decomposed at its lowest pixel levels. The pattern recognition involves obtaining the DNA sample that is used as the search pattern as well as the original DNA pattern in binary format [2, 15].

The DNA sample as well as the original pattern is fed to the TM. If a match is found the sample exists in the pattern, otherwise it does not. The implementation is done using the Turing Machine. The underlying algorithm is the linear search approach.

The TM for linear search is implemented using forward linear search algorithm, in a recursive fashion. The DNA patterns are represented in binary notation with a blank symbol as the separator between the search pattern and the original pattern.

B. *Working of the DNA Pattern Recognition Turing Machine*

The working of the DNA pattern recognition Turing Machine is explained in this section. The working can be explained with an example. The original DNA pattern consists of
TCGGTGGTCATTACTGTACCGTACGATGCAC
GTACGCTGATGTAGCTGATAGTCGG…..
and the DNA search pattern is GGTG. The original pattern is represented in binary format as
110110101110101101001100011110110001011 0110
001100011100100011011000110011110001110110 0
10011110001100101101101 0……
and the DNA search pattern in binary form is 10101110.

The input is given such that the search pattern is followed by a blank symbol which is followed by the original DNA pattern.
10101110□1101101011101011010011000111101 10
0010110110001100011100100011011000110011110
0011101100100111100011001011011010

The Standard Turing Machine for the DNA pattern Recognition (DPRTM) can be represented as

$$T_M = (Q, \sum, \Gamma, \delta, q_0, \square, F) \qquad (1)$$

where
Q= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
∑= {1, 0, a, x, y, z}
Γ= {1, 0, a, x, y, z, □},
F= {10}

and δ is the transition function, q_0 is the initial state and □ is the blank tape symbol [6]. This DPRTM is given in Fig. 1.

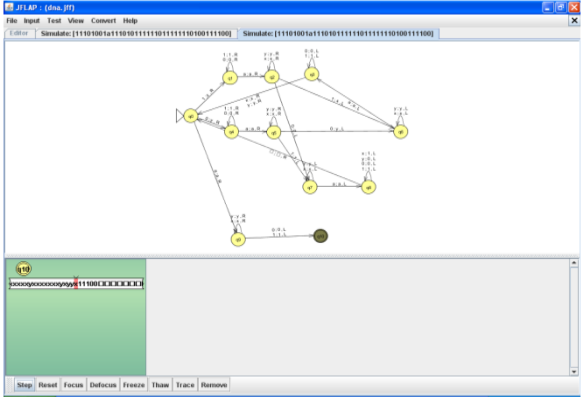
JFLAP : (dna.jff)
File Input Test View Convert Help
Editor
Simulate: [11101001a1110101111110111111110100111100]
Simulate: [11101001a1110101111110111111110100111100]
q10
Step Reset Focus Defocus Freeze Thaw Trace Remove

Fig. 1. DPRTM in execution

The input to the UTM T_U is given in the encoded format such that it contains three sections: list of final states, delta rules and the initial configuration of the input tape [12]. This is given in Fig. 2.

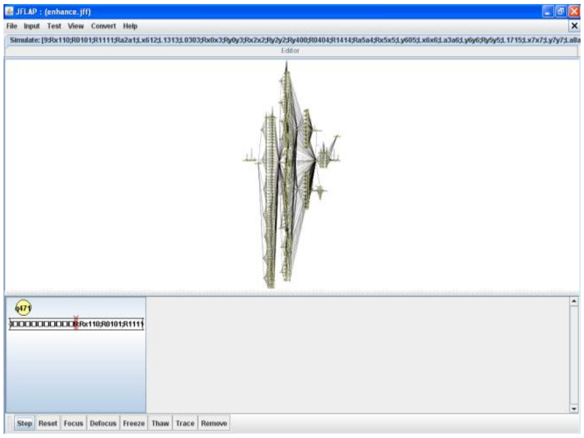
JFLAP : (enhance.jff)
File Input Test View Convert Help
Simulate: [9:Rx110;R0101;R1111;Ra2a1;Lx612;L1313;L0303;Rx0x3;Ry0y3;Rx2x2;Ry2y2;Ry400;R0404;R1414;Ra5a4;Rx5x5;Ly605;Lx6x6;La3a6;Ly6y6;Ry5y5;L1715;Lx7x7;Ly7y7;La8a
Editor
q471
Rx110;R0101;R1111
Step Reset Focus Defocus Freeze Thaw Trace Remove

Fig. 2. UTM performing DNA Pattern Recognition

The Universal Turing Machine T_U for the DPRTM can be given as

$$T_U = (Q, \sum, \Gamma, \delta, q_0, \square, F) \quad (2)$$

where
$Q= \{0, 1,2,3,4, 1001\}$,
$\sum= \{a, 1, 0, x, y, z\}$
$\Gamma= \{a, 1, 0, x, y, \square\}$,
$F= \{443\}$
and δ is the transition function, q_{11} is the initial state and □ is the blank tape symbol.

The UTM is constructed by first creating the states and then creating the transitions.

The DPR TM is implemented in JFLAP. For the standard TM the input is given directly and the execution trace can also be viewed. The TM halts in accepting state when a match is found [3].

The DPRTM can be simulated in the Universal Turing Machine (UTM) also. The UTM reduces memory usage as compared to having multiple Turing Machines. The UTM halts in final state if a matching pattern is found in the original string. The UTM does the same as the DPRTM. If the DPRTM halts in accepting state, the UTM also halts and if the DPRTM halts in a non final state when a matching pattern is not found, the UTM also does the same [11].

The DNA patterns can be easily represented and manipulated in Turing Machines as they provide an infinite memory in the form of an infinite tape extendable in both directions.

III. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

A. Performance Analysis

DNA Pattern Recognition Turing Machine (DPRTM) works very well for patterns of any length. There is no limit on the size of the input pattern as the tape of the Turing Machine is of infinite length. The various performance metrics are discussed in detail in this section.

Success Rate

Success rate refers to the ratio of the correct outputs obtained to the total number of outputs obtained.

Success ratio (S) =

$$\frac{\text{Total number of correct outputs obtained}}{\text{Total number of outputs obtained}} \quad (3)$$

The DPRTM halts in an accepting state when it successfully finds the search pattern in the original DNA pattern. The experiment has been performed using 250 samples and it has been found that the DPRTM has a success rate of 96.4 %.

Space complexity

Space complexity refers to the amount of storage space required by the problem. It is computed as a function of the input. For Turing Machines, the space complexity is computed as the number of tape cells required for the problem. Thus, for DPRTM as well as UTM, the space complexity is of the order O (n) where n is the number of tape cells occupied.

Time Complexity

The time complexity of an algorithm quantifies the amount of time taken by an algorithm. Time complexity is calculated as a function of the size of the input to the problem. For Turing Machines, the time complexity can be measured as the number of steps taken to compute the result. For DPRTM, if the number of steps taken for computation is n, then the time complexity can be expressed as O (n).

False Positives

False Positive is the condition that the output obtained misleads to a correct result. The result is not correct exactly but it gives a misunderstanding that the result is true. To find the false positive rate of the project, the experiment was conducted using 250 samples of varying length. DPRTM has a very low rate of false positive of 13.4 %.

False Negatives

False Negative leads the user to believe that the output obtained is false. The result is true actually but it gives a belief that it is not true. This is a very important factor as it misleads the user. The experiment was conducted with 250 test samples of DNA pattern and it has been found that DPRTM does not have any false negative. The False negative rate is 0%.

The experiment has been performed using 250 samples and it has been found that the DPRTM has a very good performance. The values obtained for the metrics is listed in Table 2.

Table 2. Performance Evaluation of DPRTM

Success rate	96.4 %
space complexity	O (n)
false positive	13.4 %
False Negatives	0 %

Success rate refers to the ratio of the correct outputs obtained to the total number of outputs obtained

DPRTM has been compared with many other DNA pattern recognition techniques and the results has been analyzed and discussed in this section.

B. Comparison between DPRTM and Recurrent Neural Networks using False Positive metric

A recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. A recurrent Bayesian neural network is used where outputs are recursively fed back to the input layer until a stable output pattern is resulted [13].

When ran ten times for completeness varying from 60 to 80 % and from noise 0 to 5 %, a positive predictive value of 80.4 % and a negative predictive value of 99.1 % was obtained. This results in a False positive rate of 19.6 % and a False negative rate of 0.9 %. The experiment with DPRTM has been conducted using sets of 50 patterns to 250 patterns. The False positive value and the False negative value are very less compared to the Recurrent Neural Network, having a value of 13.4 % and 0% respectively [14]. This is given in Table 3.

Table 3. Comparison between DPRTM and Recurrent Neural Network

	Recurrent Neural Network	**DPRTM**
False Positive	19.6 %	13.4%
False Negative	0.9%	0%

The graph showing the false positive rate for different sets of patterns of variable length is given in Fig. 3.

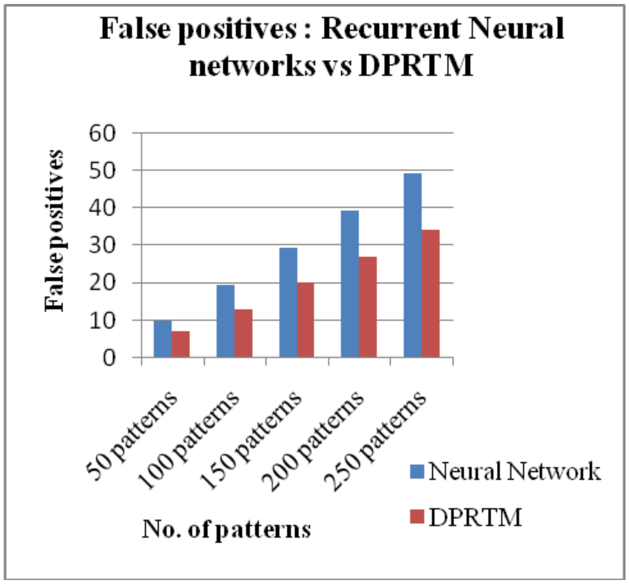

Fig. 3. Comparison between DPRTM and Recurrent Neural Networks for various DNA pattern sets

C. Comparison between DPRTM and other DNA pattern recognition techniques

DNA Pattern Recognition is a very important as well as a complex process. The DPRTM is compared with other techniques. Some of the techniques are listed below. The success rate has been obtained by performing the analysis on different sets of patterns of variable length.

RFLP Analysis

Restriction Fragment Length Polymorphism (RFLP) is one of the first methods for finding out genetics used for DNA profiling. It involves restriction enzyme digestion, followed by Sothern blot analysis. It has a success rate of 57.4 % [7].

PCR Analysis

With the invention of the polymerase chain reaction (PCR) technique, DNA profiling took huge strides forward in both discriminating power and the ability to recover information from very small (or degraded) starting samples. It has a success rate of 69 % [8].

STR Analysis

The method of DNA profiling used today is based on PCR and uses short tandem repeats (STR). This method uses highly polymorphic regions that have short repeated sequences of DNA. The success rate of STR analysis is very high, 92.3 % [9].

AmpFLP Analysis

Amplified Fragment Length Polymorphism is faster than RFLP analysis and use PCR to amplify DNA samples. It relies on variable number tandem repeat (VNTR) polymorphisms. However the success rate ranges between RFLP and PCR. The success rate is 61.1% [10].

DPRTM

DNA Pattern Recognition Turing Machine checks for the search DNA pattern in the original DNA pattern. The patterns are in binary format and this uses a Turing Machine to perform the process. Hence it has a very high success rate of 96.4%. The graph showing the success rate of all these techniques for various sets of patterns is given in Fig. 4.

It is clear that DPRTM is more beneficial than other DNA pattern recognition techniques as it has a greater value for success rate.

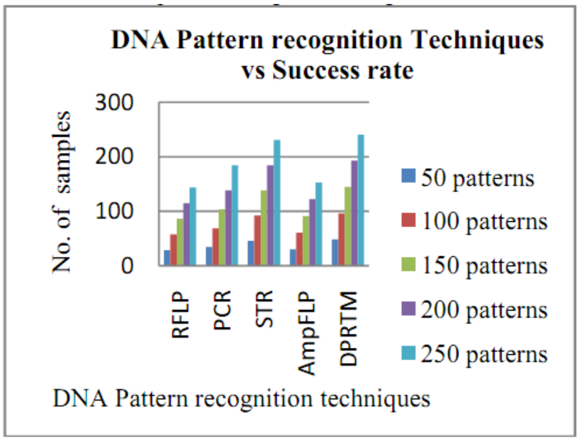

Fig. 4. Comparison between DPRTM and other DNA pattern recognition techniques

IV. CONCLUSION AND FUTURE WORK

The DNA molecules have a very long structure. It is very difficult to represent and manipulate it using common data structures. Hence it is very difficult to perform DPR using neural networks and other techniques as they have to undergo learning and training processes as well. A Turing Machine is a solution to this problem. A Turing Machine has infinite memory extendable in both directions. A UTM is a Turing Machine that can simulate any other standard Turing Machine.

The DPRTM developed using the JFLAP platform is discussed along with its working details. The DPRTM is beneficial than many of the techniques as the success rate is high.

In this paper, DPR has been successfully implemented with a standard Turing Machine and the DPRTM has been implemented using the UTM also.

The proposed method is experimented using more than 250 samples. The method has been compared with Recurrent Neural Networks using false positive metric, and many other DNA profiling techniques using success rate as the metric. The DNA pattern recognition has been implemented using a standard Turing Machine and using a Universal Turing Machine also and a comparison of the results is also presented.

The future work includes the enhancement of the model to support more number of states as well as to work for an input alphabet with more number of symbols. The model can also be enhanced to work taking two symbols at a time and comparing with the original pattern to improve the performance.

REFERENCES

[1] Nadia Nedjah, Luiza De Macedo Mourelle, “Complete Pattern Matching for DNA Computing”, *Journal of Information & Knowledge Management*, Volume 05, Issue 04, Pages: 337-343, 2006.

[2] K. Basu, N.Sriraam, R.J Richard, “A Pattern Matching Approach for the Estimation of Alignment between Any Two Given DNA Sequences”, *Journal of Medical Systems archive*, ISSN:0148-5598 , Volume 31 , Issue 4 , Pages: 247 - 253 , 2007.

[3] Jonathan Jarvis and Joan M.Lucas, “Understanding the Universal Turing Machine: An implementation in JFLAP”, *ACM Portal* Volume 23, Issue 5, Pages 180-188, 2008.

[4] Lei Chen, Shiyong Lu, Jeffrey Ram, "Compressed Pattern Matching in DNA Sequences," IEEE Computational Systems Bioinformatics Conference, *International IEEE Computer Society*, pages 62-68, 2004.

[5] M. Mohebbi, R. Mohammad, T. Akbarzadeh and A.M. Fard, “DNA structure, synthesis and fingerprinting”, *World Applied Sciences Journal* Volume 2, Issue 6, Pages:582-586, 2007.

[6] Daniel Fredouille, Chris Bryant, “Speeding-up parsing of Biological Context Free Grammars”, *Combinatorial Pattern Matching*, Volume 2 Issue 2, Pages: 43-47, 2005.

[7] Hee Van Kang, Yong Gu Cho, “A rapid DNA Extraction method for RFLP and PCR analysis from a dry seed”, *Journal of Plant Molecular Biology*, Volume 16, Issue 4, Pages 1-9,1998.

[8] Joshua S Yuan, Ann Reed, Feng Chan, C Neal Stewart, “Statistical analysis of real time PCR data”, *BMC Bioinformatics*, Volume 8, Issue 3, Pages: 74-86, 2006.

[9] Kelly J. Esslinger, Jay A.Siegel, Heather Spillane, “Using STR Analysis to detect human DNA from exploded bomb devices”, *Journal of Forensic Sciences*, Volume 49, Issue 3, 2004.

[10] Seung Hwan Lee, Jong yeol Kim, “Construction of new quadruplex Amp-FLP Systems”, *Elsevier*, Pages 123-133, 2007.

[11] Eric Gramond and Susan H.Rodger, “Using JFLAP to interact with theorems in automata theory”, *ACM Portal Proc. in SIGCSE*, Pages 336-340, 1999.

[12] Susan H.Rodger, Eric Wiebe, Kyung Min Lee, Chris Morgan, Kareem Omar and Jonathan Su, “**I**ncreasing engagement in automata theory with JFLAP”, *ACM Transactions*, Pages 403-407, 2009.

[13] Ivan Gabrijel, Andrej Dobnikar, “Online identification and reconstruction of finite automata with generalized recurrent neural networks”, *Elsevier*, Pages 101-120, 2003.

[14] C.Lee Giles, B.G Horne, T.Lin, “Learning a class of large finite state machines with a recurrent neural network”, *Elsevier*, Pages 1359-1365, 1995.

[15] Colin W.Garvie, Cynthia Wolberger, “Recognition of specific DNA sequences”, *Elsevier*, Pages 937-946, 2001.

Ms. Sumitha C.H is working as a Lecturer in the department of Computer Science and Engineering at Karunya University, Coimbatore, India. She completed her Masters in Computer science and Engineering from Karunya University in 2010 and B.Tech in Computer science and Engineering from Amrita Vishwa Vidya Peetham in 2008. Her area of interest is automata theory, compilers and networking.