International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842
Volume 1 Issue 2 (October 2014) www.irjcs.com

DNA Pattern Analysis using Finite Automata

Qura-Tul-Ein, , Yousaf Saeed, Shahid Naseem, Fahad Ahmad, Tahir Alyas, Nadia Tabassum

Abstract -- DNA is a sequence of genes that is made of the combination of nucleotides. DNA contains genetic information of living organisms. Any change in the sequence of nucleotide can change the genetic information that may cause many problems in living organisms. Therefore, it is important to analyze the genetic or DNA pattern to prevent such problems. If any problem arises then possible solutions can be used to overcome the issue. There are many methods for analyzing the DNA patterns in every field like in biology, mathematics, and statistics; similarly, Finite Automata is use in computer science for analyzing such patterns. This paper proposes analysis technique by converting patterns into Non-Deterministic finite state Automata (NFA) and then into Deterministic Finite state Automata (DFA). The purpose of using FA for DNA pattern analysis is that any change, alteration or duplication in gene or in genetic information can be detected.

Keywords --DNA, Non-deterministic Finite state Automata, Transition Table, Deterministic Finite state Automata

1. INTRODUCTION

Finite Automata (FA) has built on the idea of states. Simplest computing device model is provided by FA and the computations changes from state to state. As the state changes the computations also changes. The purpose of FA is that it is used to recognize the languages and those languages are called regular languages. There are two types of FA, Non Deterministic Finite state Automata (NFA) and Deterministic Finite state Automata (DFA).

NFA having almost the same functionality but the difference lies in the initial state. In NFA, there are multiple initial states and there can be multiple transitions [7]. It is not possible to construct NFA for every input and output because in NFA there are multiple inputs and single final state whereas it is possible to construct DFA for every input and output because there is only single input and multiple final states.

In this paper NFA and DFA is used to analyze the DNA pattern. Deoxyribonucleic acid (DNA) is present in all living organisms. DNA contains all genetic information related to the development and the functioning of the organism. This genetic information is called genes. There are certain sequences in the DNA that shows certain functions and any change in the sequence can change the functionality, which may cause abnormality in the organism.

DNA is a double stranded molecule that forms double helix shape. This double helix has nucleotides that repeat in some sequence forming a chain. This double helix DNA contains the information that ultimately converted into protein after some steps. In some papers, detail conversion of DNA into protein has discussed. Firstly the DNA is converted into pre-mRNA (messenger Ribonucleic acid) then pre-mRNA is converted into mRNA and then into RNA (ribonucleic acid) which finally transformed into protein in the final and last state. This all conversion takes place with the help of enzymes that are only responsible for conversion [1].

In the standard genetic code in DNA, four nucleotides in different sequences are used to form genetic information. These nucleotides are called codons and by the combination of these codons, amino acids forms which helps in protein synthesis. These nucleotides include Adenine, Cytosine, Guanine and Thymine are abbreviated as A, C, G, T respectively [5]. For DNA sequencing gene finder is used [6]. Pattern recognition is important for DNA problem recognition [4].

There are many methods that are used to describe patterns but mostly regular expressions are used. A sequence of characters is the basic pattern where each character shows the one amino acid or multiple amino acids. The different gene finding processes does identification of DNA [1]. Identifying genes in DNA sequence is very important [3] because if the pattern is known or the sequence is known then any abnormality in gene can be detected. It is also important to manufacture new drugs or medicines for genetic diseases.

The best way to find the sequence or to analyze the patter of DNA is FA. If there are some missing transitions then the automata is not complete [7]. First, the Non-Deterministic finite state automaton is constructed and the string input to the NFA. This input reaches either the acceptance or the rejection state. Then NFA converted into DFA in which there is only single input at a time and multiple finial states.

II. Advantages of using Finite Automata

2.1 Irrelevant Mutation

Sometimes mutation can change the gene and in severe situations after the mutation, any gene may be missing which imposes very adverse effects. So to find which gene is missing or which gene altered by mutation, finite automata can be used. While giving inputs in NFA as the rejection state occurs, the detection of particular gene is possible.

2.2 Sequencing Errors

Sequencing error is that error which arises due to change in the sequence of nucleotides in DNA or duplication of any codon in nucleotide. Any change in the nucleotides arrangement imposes very adverse effect on the synthesis of protein and at the same time, the arrangement of amino acid may change. Therefore, finite automata can detect sequence error and after the detection, necessary steps are required to balance protein synthesis in living organisms.

2.3 Incomplete Specifications

Another advantage of using finite automata for DNA pattern analysis is that any incomplete information or specification in the DNA can be easily identified by using Finite Automata.

III.. Proposed Methodology

3.1 DNA Pattern

A string of characters specified as DNA sequence [2]. DNA sequence made by the combination of nucleotides. These nucleotides are also called as codons. For the formation of DNA information, four Codons participants are known as Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) [1]. This DNA is pattern converted into RNA with the help of different enzymes and Uracil (U) replaces the Thymine. These codons combine to form amino acid that is necessary for protein synthesis. Finite automata help in determining this sequence by state transition diagrams.

TTT	TTC	TTA	TTG	CTT	CTC	CTA	CTG	ATT	ATC	ATA	ATG	GTT	GTC	GTA	GTG	TCT	TCC	TCA	TCG
CCT	CCC	CCA	CCG	ACT	ACC	ACA	ACG	GCT	GCC	GCA	GCG	TAT	TAC	TAA	TAG	CAT	CAC	CAA	CAG
AAT	AAC	AAA	AAG	GAT	GAC	GAA	GAG	TGT	TGC	TGA	TGG	CGT	CGC	CGA	CGG	AGT	AGC	AGA	AGG
GGT	GGC	GGA	GGG																

UUU	UUC	UUA	UUG	CUU	CUC	CUA	CUG	AUU	AUC	AUA	AUG	GUU	GUC	GUA	GUG	UCU	UCC	UCA	UCG
CCU	CCC	CCA	CCG	ACU	ACC	ACA	ACG	GCU	GCC	GCA	GCG	UAU	UAC	UAA	UAG	CAU	CAC	CAA	CAG
AAU	AAC	AAA	AAG	GAU	GAC	GAA	GAG	UGU	UGC	UGA	UGG	CGU	CGC	CGA	CGG	AGU	AGC	AGA	AGG
GGU	GGC	GGA	GGG																

PROTEIN

Figure 1: DNA sequence converted into RNA sequence and RNA sequence converted into Protein

3.2 Finite Automata

In Figure-2 all states shows the codons that are used in DNA. Any input string given to the NFA and if it reaches the final state then it is in accepted state otherwise it is in rejected state. Eight states are having DNA stands. Input given to the NFA, is then converted into DFA by using transition table.

$Q = \{ 1,2,3,4,5,6,7,8 \}$
$\sum = \{ A,T,C,G \}$
$\sigma = Q \times \sum$
1 ← start state
8 ← Accept state

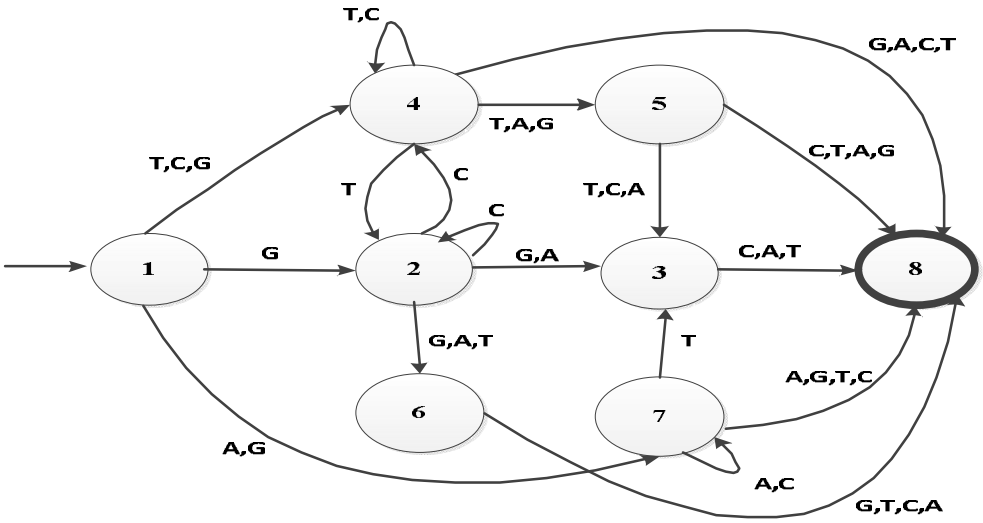

Figure 2: Non-deterministic Finite State Automata of DNA Pattern

Inputs are in the form of string so this string input passed through every state and as it reaches the final state, it means that input string is present in the DNA pattern. Once the NFA is constructed then for the construction of DFA, transition table is used. In transition table all possible outcomes from a particular state are mentioned and by using this transition table new DFA's will be formulated and finally NFA is converted into DFA.

	A	T	C	G
1	{7 }	{4 }	{4}	{2,4,7}
2	{ 3,6}	{ 6}	{2,4}	{3,6}
3	{ 8}	{ 8}	{8}	φ
4	{5,8}	{ 4,5,8,2}	{4,8}	{5,8}
5	{3,8}	{3,8}	{3,8}	{8}
6	{8}	{8}	{8}	{8}
7	{7,8}	{3,8}	{7,8}	{8}
8	φ	φ	φ	φ

Table 1: Conversion of NFA to DFA of DNA pattern using Transition table

Non-deterministic Finite Automata is now converted into Deterministic Finite Automata by using transition table. For analyzing the DNA pattern, any input string such as TCAGAAGA, TGA, and AAG given to NFA and if such string exists in the DNA pattern then this input passes through every NFA state until it approaches the acceptance state. Either for verification, the input pattern exists or not in the DNA, NFA transition takes place, if the string exist then it reaches to the acceptance state otherwise there is a rejection state showing that the input string does not exist. For analyzing the DNA, whole DNA strand or some piece of strand in the form of string is chosen. This NFA is constructed for analyzing either the whole DNA strand or a string.

After the conversion of NFA into DFA by using Transition table, number of states increases because in NFA it is not possible that all input strings can reach the acceptance state and in NFA all connected states must be traversed which increases the time complexity whereas in DFA only relevant states are traversed which decreases the analysis time.

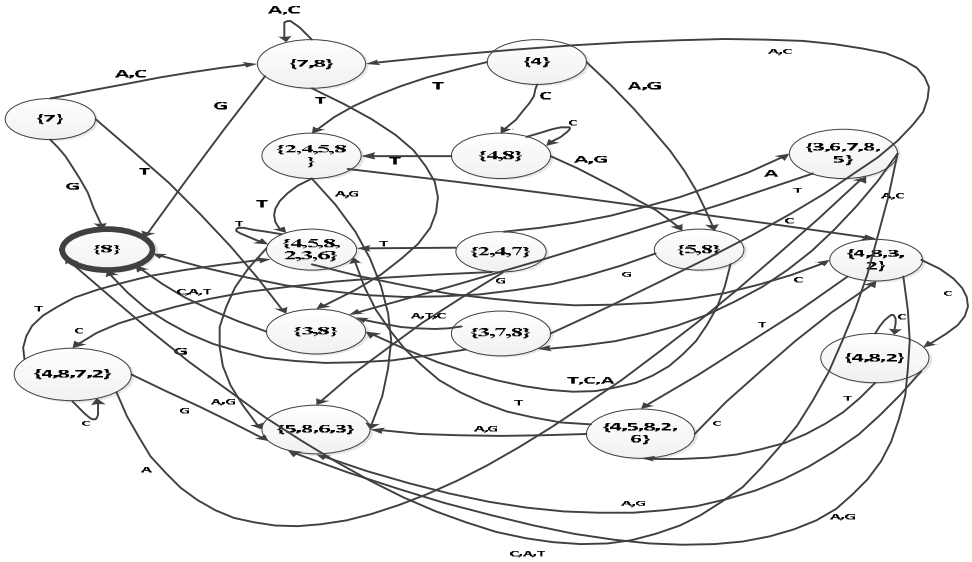

Figure 3: Deterministic Finite Automata of DNA pattern

2. CONCLUSION

It is concluded that Finite Automata can be used to analyze the sequences of DNA. Specific names assigned to such sequences and by using these names, Non-Deterministic Finite Automata is constructed. Input patterns in the form of strings are assigned to NFA and upon reaching the final state, acceptance state is achieved, however, failing to reach the final state causes rejection. Transition table is constructed for conversion of NFA into DFA and as a result, DNA pattern is analyzed.

In this paper analysis of DNA patterns are performed by using DFA due to its efficiency. The conversion of NFA to DFA takes place because it is not possible that NFA reaches the acceptance state for all inputs. Similarly, in NFA, input has to traverse all connected states that increase the time complexity. In DFA number of states increases so that only relevant states will be traversed. Therefore, to decrease the time complexity and to analyze the DNA pattern efficiently NFA is converted into DFA.

References

[1] Biedrzycki, R. (2007). DNA sequence analysis. *Greenwoog*, 8.
[2] Blanton, M., & Aliasgari, M. (2010). Secure Outsourcing of DNA searching via Finite Automata. *DBSec'10 Proceedings of 24th annual IFIP WG 11.3 working conference on Data and applications security and privacy* (p. 16). Heidelberg: Springer.
[3] Burge, C., & Karlin, S. (1997). Prediction of complete gene structure in Human Genomic DNA. *JMB*, 16.
[4] Howard, D., & Benson, K. (2003). Evolutionary computation mathod for pattern recognition of cis-acting sites. *Elsevier*, 9.
[5] Kahan, M., Gil, B., Adar, R., & Ehud, S. (2008). Towards molecular computers that operate in a biological environment . *Elsevier*,
[6] Krogh, A. (1998). Gene finding: putting the parts together. *Guide to Human Genome Computing*, 13.
[7] Lawson, M. V. (2009). *Finite Automata*. Edinburg: CRC Press.

