I J C T A, 9(28) 2016, pp. 21-30

REWRITING OF DNA-RNA BY APPLYING INDEXED GRAMMAR WITH CNF & GNF

C.Kotteeswaran* A.Rajesh ** and V.Khanna

Abstract: The term RNA-Editing describes a wide range of innovative mechanisms used for modifying nucleotide cycles pertaining to RNA transcripts present in various organisms. The main proposal is to develop indexed insertion deletion system in bio-inspired computing.

Methods: Two generic models of nucleotide modification are included in the phenomenon, insertion/deletion and replacement, described according to whether the cycle of modified RNA happens to be collinear with DNA cycle which encodes that. RNA editing may be mediated through a range of pathways which are evolutionarily and mechanistically not related. We are considering two basic types in string rewriting known as assisted insertion/deletion and next, the assisted rewriting. Original strings might be edited in accordance with the particular match with any given group of supplementary strings, knows as guides. Assisted insertion/deletion will consider pairing some string and guide in relation to a particular string correspondence. Assisted rewriting will consider pairing a guide and string in relation to some equivalence connection on an alphabet. Assisted insertion/deletion has been inspired by the RNA-editing, which is one biological process through which the authentic genetic data that is saved in DNA is changed prior to its ultimate expression.

Findings: A correspondence related to slice cycles is established and we assist rewrite cycles within a Grammar that is Context Free (CFG). Due to their left-to-right arrangement, slice cycles happen to be very convenient in dealing when compared with assisted rewrite cycles pertaining to construction of finite automata which we will encounter in proofs for regularity. Our proposed method CNF and GNF having indexed grammar produces improved and satisfactory result in rewriting (insertion/deletion) procedure.

Keywords: CNF, GNF, CFG, RNA, DNA, Insertion, Deletion, String Rewriting, Regular Languages, Finite Automata

1. INTRODUCTION

RNA editing happens to be the biological mechanism which changes the original or raw text related to the genetic data about some living organism when it has been copied (or transcribed) from DNA. Our study involves analysis of two basic formalisms pertaining to string conversion that have been inspired through RNA-editing. We have considered assisted insertion/deletion that is similar to a modification mechanism that is encountered in a living cell, along with assisted rewriting, with base on some adjacent relationship that supports easy formal analysis. A substring pertaining to the raw or original string has been adapted in both the types of string rewriting, when it is matching a string that pertains to a particular group, known as group of guides or escorts [1]. The group G of escorts happens to be set and is finite. In assisted insertion/deletion process, the escort or guide and the particular portion of string which is rewritten need not necessarily have same length; nonetheless, they must be equal till the occurrences of some distinct dummy sign. In the assisted rewriting, the substring and the guide are equal sign-by-sign in accordance with the relation of adjustment, a selected and confirmed equivalence relationship [2]. Both the favors related to rewriting foster finiteness of initial group of strings. Presuming some finite pair of escorts G,

* Research Scholar, Bharath University, Chennai-600073, India. Email: kottees@yahoo.co.in
** Professor, Department of CSE, C.Abdul Hakeem College of Engg & Tech, Vellore-632509, India. Email: amrajesh2@rediffmail.com,
*** Professor & Dean, Department of IT, Bharath University,Chennai-600073, India. Email: drvkannan62@yahoo.com

in the two cases, just one finite group of strings may be obtained through repeated rewriting of any given string. We show in our study that for both the cases, also the regularity of first string group is fostered. Beginning with language L, here, we are considering an expansion Li/d of that language related to all rewrites attained through assisted insertion/deletion and expansion LG of that language attained through adding up every adjustment-oriented assisted rewrites. The primary outcome of the study describes the fact regularity of L indicates the consistency of LG [3], [4]. The general belief was that DNA-coded, exonic cycle data kept in genome will directly predict amino acid constitution of resulting products of genes. Anyhow, this particular view needed to be altered after discovery of RNA-editing process which empowers a cell in recoding genomic data in some regulated and systematic way, while selectively modifying genes readout at some single nucleotide position inside the initial RNA transcript [5]. RNA-editing type discovered first works using insertion and/or deletion of the bases, while involving small assisting RNAs that are complementary to an objective RNA [6], [7]. This particular process that is seen in mitochondria pertaining to primitive eukaryotes helps creating open readable frames with no existence. This happens to be a basically different type of mechanism compared to other kinds of RNA-editing that is characterized through base replacement, in the pre-mRNAs pertaining to higher eukaryotes. In this case, open readable frame is changed, thus generating a protein presented with single or more modifications in amino acid cycle. Replacement RNA-editing has been detected in several mitochondrial RNAs pertaining to higher plants having mostly cvtidine-to-uridine i.e., C-to-U or U-to C modifications [8]. It has been revealed by one recent systematic assessment of Arabidopsis mitochondrial genes that around 8% of all the C containing codons get modified, documenting widespread instance of C-to-U modification in the higher plants. RNA-editing is primarily represented in mammals using C-to-U and also adenosine-to-inosine (that performs as guanosine), i.e., (A-to-I) alterations creating sole amino acid modifications in resultant protein. Often, this is seen to have consequences on the protein [9] function. Amid the mRNAs which were seen to undergo modification, best possible examples happen to be apolipoprotein B (apoB) transcripts (that is, C-to-U modification) and the messages pertaining to serotonin and glutamate receiving subunits (A-to-I modifications).

2. RELATED WORK

Popularity of formalisms that are grammar-based has been increasingly growing in research about vision-oriented action recognition in the last two decades [10]. Breaking down complicated behaviors into primary and simple actions (signs), and certain means of performing some action or behavior can both be defined as strings related to the symbols. Grammar model rules determine which of the combination of elementary activities includes a valid function of some behavior. Due to several reasons, grammar model proves appealing to representing complex action patterns. It is possible to represent Grammar elegantly, its structure may be interpretable, and also it is possible to be utilized for formulating concise description about patterns of actions. Several researches have followed syntactic methods for recognizing various kinds of non-oral behavior such as Finite State Mechanism (FSM) regarding recognition of hand gesture [11] or Context less Grammar while being used for representing and recognizing interactions and actions of humans [12], [13]. For addressing uncertainty produced by computer vision or noisy sensors, the syntactic methods are being extended to comprise of the probabilities from early stages. Stochastic Context-less Grammar [14] (SCFG) has been applied on various vision-oriented programs like surveillance in parking or recognition of simple hand movement [15]. Many of such methods have employed the parsing algorithm of Earley-Stolcke [14] toward effective and probabilistic parsing. This concerned author [16] make use of similar methods toward action identification during the task of Hanoi Tower [17], and for identifying multi-user actions in card games like blackjack. All these systems describe manually the syntax of applied grammar and they have exerted that only regarding identification during levels of comparatively complicated actions with clear compositionality in relation to both structures and

units. None of these met with the confrontation of having to learn grammar from the samples of the action performances. Two exceptions were found. These work on [12] very easy action identification and through [11] who has learnt SCFGs of programs like identification of the traffic events, multiple user interactions, and gymnastic exercises.

(a) *Presynaptic filament and Recombinases:* HR reaction can be mediated through recombinases which catalyze formation of the hybrid joints related to homologous molecules of DNA. There are a couple of recombinases that exist in the eukaryotes, namely, Dmc1 and Rd51. Rad51 is vital for both meiotic and mitotic events, while Dmc1 may be expressed in only meiosis and the performance is restricted in that zone. The active form ofRad51 in its catalyst mode includes one right-handed protein filament that is helical and assembled on a ssDNa. Such recombinase-ssDNA of nucleoprotein filament has been frequently known as presynaptic type filament (San Filippo et al. 2008). Having been assembled once, this presynaptic filament will run a search regarding homologous chromatid; it catalyzes invasion of a donor chromatid for forming one DNA connection molecule known as displacement (D)-loop. The following steps will consist of DNA litigation, DNA synthesis, and resolution of the DNA intermediaries along any one of the many pathways, in order for completing the process of repair [18], [19].

(b) Education from Studies of Bacteria: There happen to be two known routes in Escherichia coli, which will contribute to the DNA final resection. RecBCD intricate will function in a major resection route, while a minor pathway happens to be dependent over RecQ-RecJ set [20]. RecB subunit pertaining to RecBCD intricate shelters both endonuclease and helicase activities, and RecD subunits will also possess some helicase action. The RecC will identify certain particular cysle in the DNA (5-GCTGGTGG-3) that is known as X, and it will serve as scaffolding operation in the protein intricate assembly. The RecBCD will engage a DNA end; it will separate rapidly two DNA srands, through a combined action related to RecD and RecB helicase, towards preparation for cleavage of strands via RecB. The strand scission and DNA unwinding characteristics of RecBCD intricate will be modulated via the X cycle in order for favoring a generation of the 3 ssDNA toward nucleation of bacterial recombinase of RecA [22]. On RecQ-RecJ route of DNA terminal resection, RecQ helicase will separate strands of DNA from one end, and subsequent digestion of 5 DNA strand through the 5-to-3 exonuclease action related to RecJ will lead to accumulation of 3 ssDNA [21].

3. PROPOSED WORK

3.1 Overview

The proposed technique CNF and GNF with indexed grammar is classifying the family membrane of gene based on the RNA and DNA by inserting and deleting the unmatched gene from particular family. The GNF technique has been used to identify the spoof and for constructing PDA. CNF is eliminating the production of empty constraint and finding the equivalent grammar in the gene properties. Rewriting process is also being done by the CNF after the insertion and deletion process over gene through RNA and DNA classification within negative and positive matching.

The below mentioned figure 1 is describing the proposed framework within every step; the initial step is accessing the dataset of RNA and DNA, then it processing the dataset to calculate the equivalence bond of every gene and applying indexed grammar for insertion and deletion process. The insertion and deletion process has been done on the positive and negative criteria of the gene similarity. Then CNF and GNF have been applied to find out the family membrane of every particular gene based on the classified RNA and GNA.

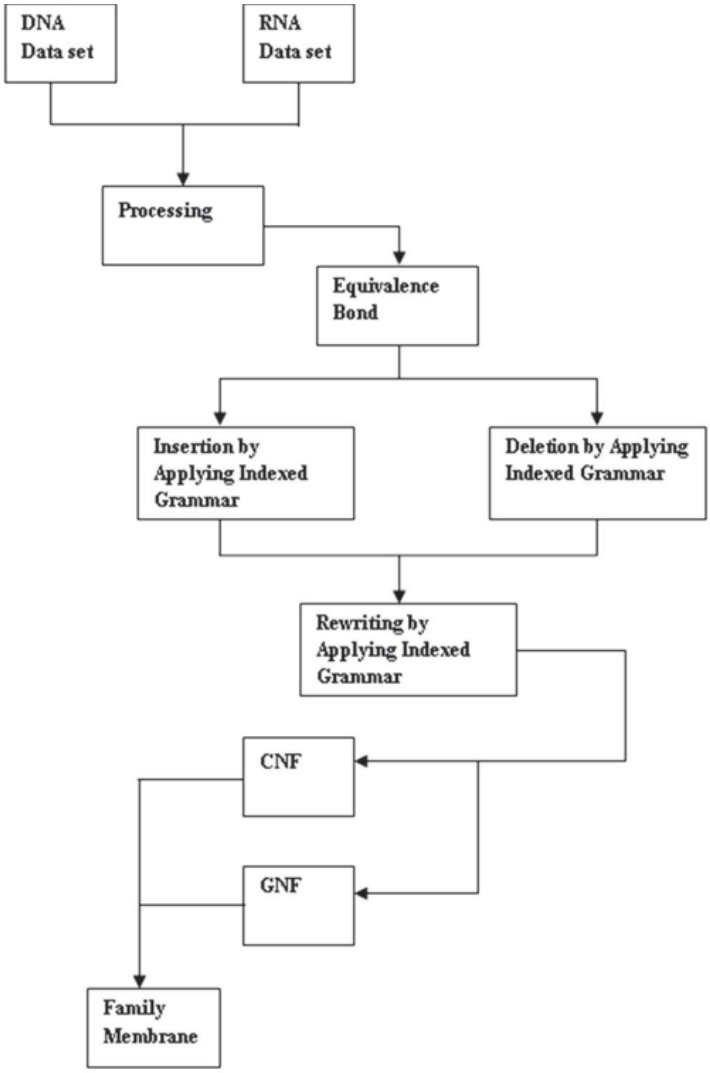

Figure 1. Proposed Framework

3.2 Greibach Normal Form (GNF)

Constraints are placed on length of right sides related to a creation in Chomsky Normal Form (CNF), While Greibach Normal Form (GNF) puts restrictions on positions wherein variables and terminals may appear. GNF proves useful for simplifying certain proofs and for making constructions like Push Down Automaton (PDA) agreeing with CFG.

Definition: Context-less grammar is known to be present in Greibach Normal Form (GNF) when each production has the said form.

$A \rightarrow ax$,

where $a \in T$ and $x \in V^*$

For grammar in the GNF, RHS of each production will have single terminal that is followed by some string of the variables. Each context-less language L having no $\in$ may be developed by some grammar wherein all productions will be of a form $A \rightarrow ax$, in which a happens to be a variable, A is terminal, and α will be some string of –less variables (probably empty).

Define: A-production to be a production with variable A on the left.

Let G = (V,T,P,S) be a CFG.

Let A $\rightarrow \alpha 1 B \alpha 2$ be a production in P and B $\rightarrow \beta 1 |\beta 2| \ldots | \beta r$ be the set of B-productions.

Let G1 = (V,T,P,S) be obtained from G by deleting the production A $\rightarrow \alpha 1 B \alpha 2$ from P and adding the productions A $\rightarrow \alpha 1 \beta 1 \alpha 2 |\alpha 1 \beta 2 \alpha 2| \alpha 1 \beta r \alpha 2$.

Then L(G) = L(G1)

Lemma

Let G = (V,T,P,S) be a CFG.

Let A $\rightarrow A\alpha 1 | A\alpha 2 | \ldots | A\alpha r$ be the set of A-productions for which A is the leftmost symbol for the right hand side.

Let A $\rightarrow \beta 1 |\beta 2| \ldots | \beta s$ be the remaining A-productions.

Let G1 = (V$\cup${B}, T, P1, S) be the CFG formed by adding the variable B to V and replacing all the A-productions by the productions:

1. A $\rightarrow \beta I$ A $\rightarrow \beta i B$ $1 \leq i \leq s$
2. B $\rightarrow \alpha I$ B $\rightarrow \alpha i B$ $1 \leq i \leq r$

Then L(G1) = L(G)

3.3 Chomsky Normal Form (CNF)

A context language L having no λ-production is created through a grammar wherein creations are found to be of form $A \rightarrow BC$ or $A \rightarrow a$, in which A, B $\in$ VN, and a $\in$ V T.

Procedure for finding Equivalent Grammar on the CNF:

(i) unit productions, if they are present

(ii) Remove terminals on RHS having length two and more than two

(iii) Limit the count of variants on RHS pertaining to productions as two.

3.4 Rule of Deletion

A production of A $\rightarrow x1$ B $x2$ may be removed from grammar, in case we are able to replace it with a group of productions wherein B will be substituted by all the strings that it derives in a single step. In the stated result, it is essential that B and A happen to be different variables.

Elimination of Unuseful Productions

In the grammar G with P, S $\rightarrow aSb|\lambda|$A, A $\rightarrow a$A.

In grammar P with G, production S $\rightarrow$ A will not be playing any role as it is not possible to transform A as some terminal string. As ‘A’ will occur in the extracted string S, this never leads any form of sentence. The production rule can be removed which will not affect language of the grammar.

3.5 Removal of Empty Production

Productions pertaining to context-less grammars may be cajoled into a wide range of formats without impacting the grammars’ power of expression. When empty string is not found to belong to some language,

there will be a way of eliminating productions having form $A \rightarrow \lambda$ from that grammar. In case that empty string happens to belong to some language, it is possible for us to remove λ from out of all productions except for one single production of $S \rightarrow \lambda$.

In such a case, we may remove any number of occurrences of S from out of the productions RHS.

3.6 Rewriting

The concept behind assisted rewriting happens to be that a sign may be substituted by some equivalent sign, while equivalence is taken related to some particular adjustment relationship ~. Let Σ be some finite alphabet, ~ be an equivalence relationship over Σ, known as the relation of adjustment. When $a \sim b$, then we could presume '*a*' may be adjusted with *b*. Consider some string of $u \sim \Sigma^*$. *We may* write *#u* as its length, and use *u*[i] for denoting its ith element, i.e., I = 1,…., *#u*, and then let *u*[*p, q*] denote substring *u*[*p*] *u*[*p+1*] … *u*[*q*]. The relationship of ~ may be elevated to Σ by keeping $u \sim v$ *iff* $\#u \wedge Ai = 1,\ldots,\#u$: $u[i] \sim v[i]$. After that, we can describe the idea of assisted rewriting which will involve some adjustment relationship.

3.7 Slice Cycles and Rewrite Cycles

Here, we introduce one auxiliary idea viz. the idea of slice cycle which may be considered to be one 'vertical' variant of a 'horizontal' idea pertaining to some rewrite cycle. A pair of guide-position (*g, p*) denotes one intentional assisted rewrite having *g* of any string u located at *p*. In order for thus rewrite to match, it is essential that we have $p + \#g \leq \#u$.

The sequence Q induces a sequence of strings by putting $u0 = u$ and *uk* such that $uk - 1 \Rightarrow gk, pkuk$ for $k = 1,\ldots, r$. To conclude that $uk - 1 \sim gk, pkuk$ is indeed a proper guided rewrite step, in particular that we have $uk - 1[pk + 1, pk + \#gk] \sim gk$, we use the assumption $u[pk + 1, pk + \#gk] \sim gk$ and the fact that if $u \Rightarrow g, p\ v$ then $u[p + 1, p + \#g] \sim v[p + 1, p + \#g]$ and $u \sim v$. Therefore, by induction $u \Rightarrow * uk - 1$ and $u[pk + 1, pk + \#gk] \sim uk - 1[pk + 1, pk + \#gk]$.

3.8 Algorithm

Function Best Discontinuous Parser (π)

$n = |\pi|$

for $i \leftarrow 2$ to n do

for $A \subset \{1\ldots n\}$ s.t. $|A| = i$ do

best [A] $\leftarrow \infty$

for B, C s.t. $A = B \cup C \wedge B \cap C = \varnothing$ do

let *a*, *b* and *c* denote the number of

(A, π(A)), (B, π(B)), and (C, π(C))'s spans

Compl [$A \rightarrow BC$] = max {$a + b + c$, best[B], best[C]}

If compl [$A \rightarrow BC$] < best[A] then

Best[A] $\leftarrow$ compl [$A \rightarrow BC$]

Rule[A] $\leftarrow A \rightarrow BC$

Return best [{1…*n*}]

4. RESULT AND DISCUSSION

In order to measure the performance of our proposed approach, a sequence of experiments on extracted dataset were conducted. Based on the following configuration our proposed method should be implemented

1. Windows 7,
2. Intel Pentium(R),
3. CPU G2020 and
4. Processer speed 2.90 GHz.

Table 1. Performance Analysis

	Precision	*Recall*	*Match*	*Problematic*
Overall	84.20	84.56	42.23	25.26
One Error	90.1	90.80	59.89	15.72
Two or More	79.59	79.71	28.48	33.33
Incorrect Word	96.11	96.12	78.64	6.80
Extra Word	84.25	88.60	39.29	23.21
Missing Word	88.22	86.00	43.90	19.51
Level 1	85.71	80.96	50.00	31.25
Level 2	82.95	82.74	45.45	29.43
Level 3	84.59	85.11	40.12	24.24
Level 4	84.95	86.54	43.81	22.86

The above mentioned table 1 is describing about the performance analysis within several factors such as one error, two error, incorrect word, missing word, extra word, and level 1 to level 4 within precision, recall, match and problematic factors.

Table 2. Error Rate Detection

	Detect %		*Ins Err*		*Sub Err*		*Del Err*	
S	*On*	*Off*	*On*	*Off*	*On*	*Off*	*On*	*Off*
a	98.8	92.5	0.0	0.0	0.0	0.0	1.2	7.5
b	97.8	90.8	0.0	0.0	0.0	0.0	2.2	9.2
c	100.0	80.0	0.0	0.0	0.0	20.0	0.0	0.0
d	100.0	91.7	0.0	0.0	0.0	0.0	0.0	8.3
e	94.0	74.9	1.2	5.0	1.2	7.5	3.6	12.5
f	95.6	70.3	0.0	2.3	0.0	9.2	4.4	18.3
g	92.9	88.9	0.0	0.0	0.0	0.0	7.1	11.1
h	79.0	12.5	10.5	36.5	10.5	43.8	0.0	7.3
i	90.6	55.8	4.7	17.2	2.4	9.8	2.4	17.2

The above mentioned table 2 is describing about the error rate detection and explaining about every error detection rate in i^{th} term. The Insertion, Subtraction, and Deletion error rate has been considered for calculating the error rate.

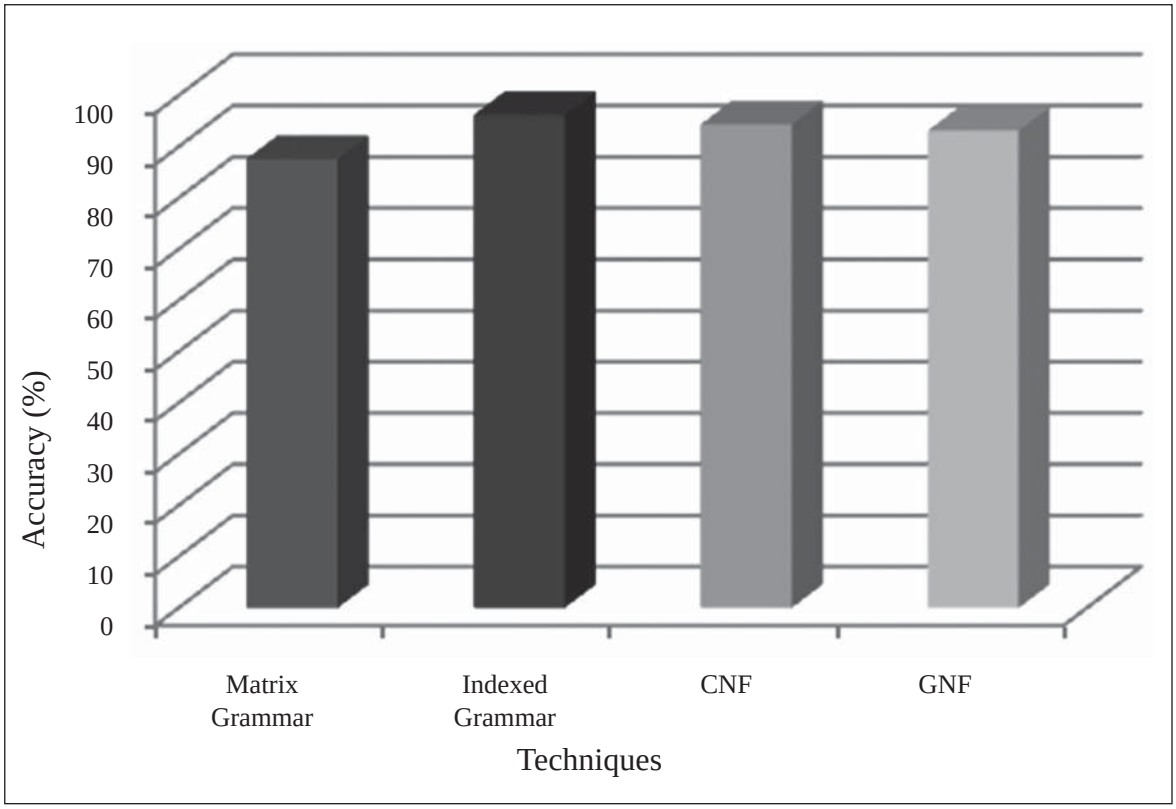

Figure 2. Performance Accuracy

The above mentioned figure 2 is describing about the performance measurement and accuracy of the proposed technique and doing comparison amid of existing techniques. The comparison result is showing that proposed technique producing better result.

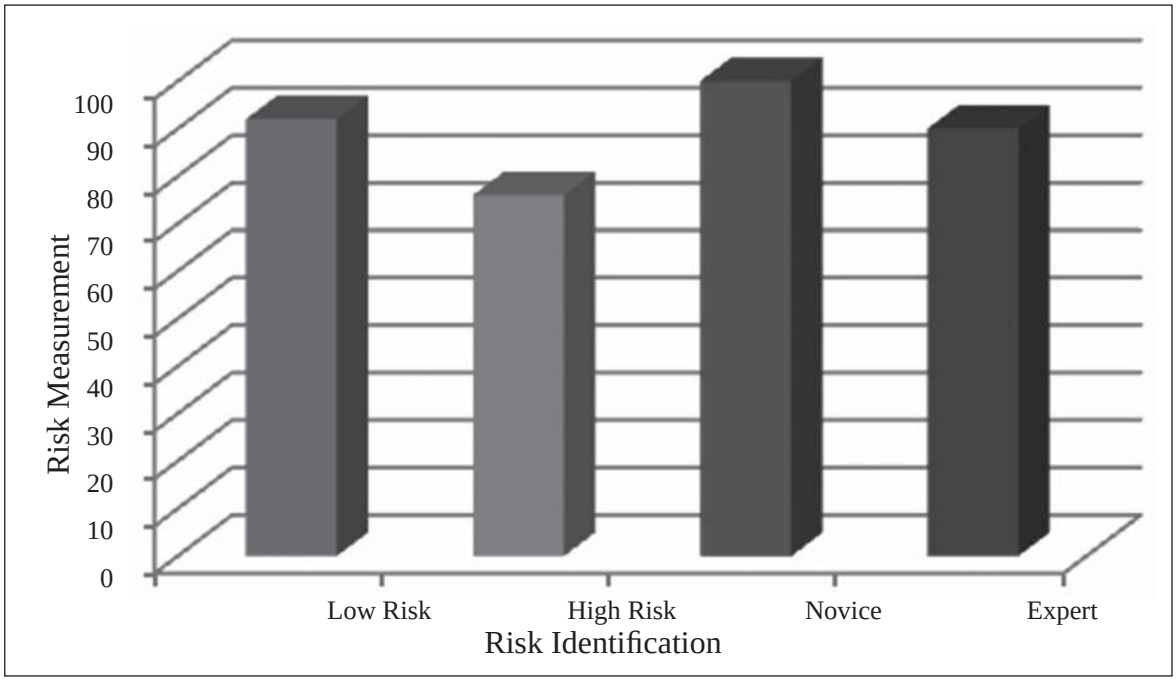

Figure 3. Risk Measurement

The above mentioned figure 3 is describing about the risk measurement within four distinct level of the condition, where low risk, high risk, novice and expert parameter low have been verified.

Table 3. Risk Identification

Behavior	*Detection Accuracy (%)*
Low - Risk	92
High – Risk	76
Novice	100
Expert	90

The above mentioned table 3 is describing about the risk analysis over the proposed system and producing the risk detection accuracy.

5. CONCLUSION

In this paper we discussed specific concepts of string rewriting: a more flexible notion focusing on insertions and deletions of a dummy symbol, another tougher notion based on an equivalence relation. Given a language L we considered the extended languages Li/d and LG comprising the closure of L for the two types of guided rewriting with guides from a finite set G. In particular, as our main results we proved that these closures preserve regularity. For doing so we investigated the local effect of guided rewriting on two consecutive string positions, leading to a novel notion of a slice sequence. Finally, the theorem for adjustment-based rewriting was proved by an automaton construction exploiting a slice sequence characterization of guided rewriting. Via a compression scheme for strings of dummy symbols, the theorem for guided insertion/deletion followed. Editing events can only be detected indirectly by comparing the genomic DNA sequences with that of the mature mRNA for the respective gene. In addition, the extent of RNA editing at a particular nucleotide position varies between fractions of a percent up to 100% in vivo.

References

1. J.D. Alfonzo, O. Thiemann, and L. Simpson, “The mechanism of U insertion/deletion RNA editing in kinetoplastid mitochondria”, Nucleic Acids Research, 25(19):3751-3759, 1997.
2. G.J. Arts and R. Benne, “Mechanism and evolution of RNA editing in kinetoplastida”, Biochimica et Biophysica Acta, 1307(1):39-54, 1996.
3. R. Benne, J. van de Burg, J. Brakenhoff, P. Sloof, J. van Boom, and M. Tromp, “Major transcript of the frameshifted coxii gene from Try-panosome Mitochondria contains four nucleotides that are not encoded in the DNA Cell”, 46:819-826, 1986.
4. F. Biegler, M.J. Burrell, and M. Daley “Regulated RNA rewriting: Modelling RNA editing with guided insertion”, Theoretical Computer Science, 387(2):103-112, 2007.
5. Benne R, Van dBJ, Brakenhoff JP, Sloof P, Van BJ Tromp MC, “Major transcript of the frameshifted coxII gene from trypanosome mitochondria contains four nucleotides that are not encoded in the DNA”, Cell 1986; 46:819-826.
6. Simpson L. “RNA editing-An Evolutionary Perspective”, In: Atkins J, Gesteland R, editors. The RNA World. Cold Spring Harbor, New York: Cold Spring Harbor Laboratory Press, 1999 pp. 585-608.
7. Stuart K, Allen TE, Kable ML, Lawson S. Kinetoplastid, “RNA editing: complexes and catalysts”, Curr Opin Chem Biol 1997;1: 340-346.
8. Steinhauser S, Beckert S, Capesius I, Malek O, Knoop V, “Plant mitochondrial RNA editing”, J Mol Evol 1999;48:303-312.
9. Giege P, Brennicke A, “RNA editing in Arabidopsis mitochondria effects 441 C to U changes in ORFs”, Proc Natl Acad Sci USA 1999; 96: 15324-15329.
10. Chanda, G., and Dellaert, F. “Grammatical methods in computer vision: An overview”, Technical report, Georgia Institute of Technology, 2004.
11. Hong, P.; Turk, M.; and Huang, T. S. 2000, “Gesture modeling and recognition using finite state machines”, In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, 410–415. IEEE.

12. Kitani, K. M.; Sato, Y.; and Sugimoto, A. (2006), “An mdl approach to learning activity grammars”, Technical Report 376, IEICE - The Institute of Electronics, Information and Communication Engineers, Tokyo, Japan.

13. Ryoo, M. S., and Aggarwal, J. (2006), “Recognition of composite human activities through context-free grammar based representation”, In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, 1709–1718.

14. Stolcke, A. (1994) “Bayesian Learning of Probabilistic Language Models”, Ph.D. Dissertation, University of California at Berkeley, Berkeley, CA.

15. Ivanov, Y., and Bobick, A. (2000). Recognition of visual activities and interactions by stochastic parsing. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8):852–872.

16. Minnen, D.; Essa, I.; and Starner, T. (2003). “Expectation grammars: leveraging high-level expectations for activity recognition”, In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, 626–632.

17. Moore, D., and Essa, I. (2002), “Recognizing multitasked activities from video using stochastic context-free grammar” In Eighteenth national conference on Artificial intelligence, 770–776. Menlo Park, CA, USA: American Association for Artificial Intelligence.

18. San Filippo, J., Sung, P., and Klein, H. 2008. “Mechanism of eukaryotic homologous recombination”, Annu. Rev. Biochem. 77: 229–257.

19. Krogh, B.O. and Symington, L.S. 2004, “Recombination proteins in yeast”, Annu. Rev. Genet. 38: 233–271.

20. Spies, M. and Kowalczykowski, S.C. 2005, “Homologous recombination by RecBCD and RecF pathways”, In The bacterial chromosome (ed. N.P. Higgins), pp. 389–403. ASM Press, Washington, DC.

21. Shereda, R.D., Bernstein, D.A., and Keck, J.L. 2007, “A central role for SSB in Escherichia coli RecQ DNA helicase function”, J. Biol. Chem. 282: 19247–19258.

22. Anderson, D.G. and Kowalczykowski, S.C. 1997, “The translocating RecBCD enzyme stimulates recombination by directing RecA protein onto ssDNA in an irregulated manner”, Cell 90: 77–86.