Modelling Intermolecular Structures and Defining Ambiguity in Gene Sequences using Matrix Insertion-Deletion Systems
Lakshmanan Kuppusamy, Anand Mahendran, Éric Villemonte de La Clergerie

To cite this version:

HAL Id: hal-00659487
https://hal.inria.fr/hal-00659487
Submitted on 12 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Modelling Intermolecular Structures and Defining Ambiguity in Gene Sequences using Matrix Insertion-Deletion Systems

Lakshmanan Kuppusamy\textsuperscript{a,2}, Anand Mahendran\textsuperscript{a} and Eric Villemonte de la Clergerie\textsuperscript{b}

\textsuperscript{a}School of Computing Science and Engineering
VIT University
Vellore-632 014, India.
Email: \{klakshma, manand\}@vit.ac.in

\textsuperscript{b}INRIA-Rocquencourt
B.P. 105, Le Chesnay - 78153, France.
Email: Eric.De_La_Clergerie@inria.fr

Abstract. Gene insertion and deletion are considered as the basic operations in DNA processing and RNA editing. Based on these evolutionary transformations, a computing model has been formulated in formal language theory known as insertion-deletion systems. Recently, in \cite{6}, a new computing model named Matrix insertion-deletion system has been introduced to model various bio-molecular structures such as hairpin, stem and loop, pseudoknot, attenuator, cloverleaf, dumbbell that occur at intramolecular level. In this paper, we model some of the intermolecular structures such as double strand languages, nick languages, hybrid molecules (with R-loops), Holliday structure, replication fork and linear hybridization (ligated) languages using Matrix insertion-deletion system. In \cite{2}, the ambiguity in gene sequence was defined as deriving more than one structure for a single gene sequence. Here, we propose a different view of understanding the ambiguity in gene sequences: A gene sequence is obtained by more than one way such that their intermediate sequences are different. We further classify the ambiguity into many levels based on the components axiom, string (order of deletion/insertion) and contexts (order of the used contexts). We notice that some of the inter and intramolecular structures obey the newly defined ambiguity levels.

Keywords. insertion-deletion systems, intermolecular structures, ambiguity, matrix grammars, gene sequences

Introduction

Insertion-deletion systems are introduced to theoretically analyze the insertion and deletions operations that take place in gene sequences. Informally, the insertion and deletion

\textsuperscript{1}\textsuperscript{1}This work was partially supported by the project SR/S3/EECE/054/2010, Department of Science and Technology (DST), New Delhi, India.

\textsuperscript{2}\textsuperscript{2}Corresponding Author: School of Computing Science and Engineering, VIT University, Vellore 632 014, India.E-mail: klakshma@vit.ac.in
operations of an insertion-deletion system is defined as follows: If a string $\beta$ is inserted between two parts $w_1$ and $w_2$ of a string $w_1w_2$ to get $w_1\beta w_2$, we call the operation as insertion, whereas if a substring $\alpha$ is deleted from a string $w_1\alpha w_2$ to get $w_1w_2$, we call the operation as deletion. These systems opened a particular attention in the field of formal languages as the system is not exactly based on rewriting systems.

As DNA and RNA molecules can be considered as a sequence of symbols (i.e., strings) over $\{a, t, g, c\}$ and $\{a, u, g, c\}$ respectively, problems that exist in such molecules can be studied using formal grammars. The following example witnesses the correlation between formal grammars and bio-molecular structures. Consider a context-free language $\{ww^R \mid w \in \{a, b\}^*\}$ where $w^R$ is the reverse of $w$. Consider the following gene sequence $gcatgcgcatgc$. As the complementary pairs $\bar{a} = t$, $\bar{t} = a$, $\bar{g} = c$ and $\bar{c} = g$, the above gene sequence resembles a word in the palindrome language $\{ww^R \mid w \in \{a, t, g, c\}^*\}$.

There exist some relations between intramolecular gene sequences and natural language constructions such as triple agreements: $\{a^n b^n c^n \mid n \geq 1\}$, crossed dependencies: $\{a^n b^m c^n d^m \mid n, m \geq 1\}$ and copy language: $\{ww \mid w \in \{a, b\}^*\}$ [1], [2]. In below, we discuss briefly some of the important intra and intermolecular structures that are predominantly available in bio-molecules such as protein, DNA and RNA. Fig.1 shows the intramolecular structures (a) stem and loop, (b) cloverleaf and (c) dumbbell. Note that these structures can be represented by context-free grammars. However, there are some more structures that occur frequently in bio-molecules which cannot be modelled by context-free grammars. Fig.2 represents such intramolecular structures (a) pseudoknot and (b) attenuator. In Fig. 1(a) and Fig. 2, the strings are obtained by reading the symbols as per directed dotted lines. The string $ageucaug$ (refer Fig. 2a) represents the pseudoknot structure, the string $agaucuaga$ (refer Fig. 2b) represents the attenuator structure and they resemble the crossed dependency language and the copy language respectively.

In RNA, the gene sequences are mostly intermolecular rather than intramolecular. Such intermolecular structures are modelled by cutting the intramolecular secondary structures randomly by leaving some open points such that base pairing can be done at these open points. Fig.3 represents some of the intermolecular structures (a) double strand languages and (b) nick languages where the cut takes place at arbitrary position is represented by a $\bullet$. In Fig.3 $S$, stands for the non-terminal in their corresponding context-free grammar.

In the last two decades, attempts have been used to model these intra and intermolecular structures by defining new grammar formalisms like crossed-interaction grammar.
simple linear tree adjoining grammars and extended simple linear tree adjoining grammars [11]. In [3], cut grammars, ligation grammars were defined specifically to model intermolecular structures. However, there is no unique grammar system that encapsulates all essential and important (intra/inter) bio-molecular structures. For example double copy language cannot be modelled by a simple linear tree adjoining grammar [11].

Very recently, in [6] a new simple and powerful biologically inspired computing model namely Matrix insertion-deletion system has been introduced by combining insertion-deletion system and matrix grammar with a strong motivation to model the above mentioned intramolecular structures in terms of languages. Though, insertion-deletion system itself is computationally complete, and therefore possible to generate any recursive enumerable language, constructing insertion-deletion grammars for some languages (that form bio-molecular structures) is not easy and the procedure to simulate the grammar is very tedious. For example, consider the triple agreement language \( \{ a^n b^n c^n \mid n \geq 1 \} \) that forms the triple stranded DNA. Defining the insertion-deletion grammar is very difficult since we need to make sure that whenever one \( a \) is introduced, simultaneously, one \( b \) and \( c \) must be introduced at the appropriate places, thus the synchronization between \( a \), \( b \) and \( c \) is taken care. To do this in insertion-deletion system, we need to introduce a new marker near the inserted \( a \) and move the marker towards right by passing all \( a \) until a \( b \) is encountered (for inserting \( b \)) and then move further to encounter a \( c \) (to introduce a \( c \)). Besides, the marker need to be deleted on crossing each symbol. If not constructing this way, we need to identify the Chomsky grammar which generate this language, and convert it to a Kuroda (or suitable) normal form and from that nor-
Ambiguity may be useful while considering inheritance properties and phylogenetic trees. Study of this concept of ambiguity in gene sequences is also possible (i.e., starting from one sequence we are able to get another sequence in more than one way such that the intermediate sequences are different). With this view of ambiguity, in this paper, we formally define the various levels of ambiguity for Matrix insertion-deletion systems. Study of this concept of ambiguity may be useful while considering inheritance properties and phylogenetic trees.

1. Preliminaries

Here, we recall the basic notions which are used in the paper. A finite non-empty set $V$ or $\Sigma$ is called an alphabet. We denote by $V^*$ or $\Sigma^*$, the free monoid generated by $V$ or $\Sigma$, by $\lambda$ its identity or the empty string, and by $V^+$ or $\Sigma^+$ the set $V^* - \{\lambda\}$ or $\Sigma^* - \{\lambda\}$. The elements of $V^*$ or $\Sigma^*$ are called words or strings. For a string $w$, $|w|_t$ denotes number of $t$ in $w$. The language of DNA (RNA) can be considered over $\Sigma^*_{DNA(RNA)} = \{a, t(u), g, c\}$. For more details on formal language theory, we refer to [9].

Next, we look into the basic definitions of insertion-deletion systems. Given an insertion-deletion system $\gamma = (V, T, A, R)$, where $V$ is an alphabet, $T \subseteq V$, $A$ is a finite language over $V$, $R$ is a finite triples of the form $(u, \alpha/\beta, v)$, where $(u, v) \in V^+ \times V^+$, $(\beta, \alpha) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$. The pair $(u, v)$ is called contexts. Insertion rule will be of the form $(u, \lambda/\beta, v)$ which means that $\beta$ is inserted between $u$ and $v$. Deletion rule will be of the form $(u, \alpha/\lambda, v)$, which means that $\alpha$ is deleted between $u$ and $v$. In
other words, \((u, \lambda/\beta, v)\) corresponds to the rewriting rule \(uv \to u\beta v\), and \((u, \alpha/\lambda, v)\) corresponds to the rewriting rule \(uvv \to uv\).

Consequently, for \(x, y \in V^\ast\) we can write \(x \Longrightarrow^\ast y\), if \(y\) can be obtained from \(x\) by using either an insertion rule or a deletion rule which is given as follows: (the down arrow \(\downarrow\) indicates the position where the string is deleted and the underlined string indicates the string deleted/inserted)

1. \(x = x_1u^i\hat{v}x_2, y = x_1u^i\beta\hat{v}x_2\), for some \(x_1, x_2 \in V^\ast\) and \((u, \lambda/\beta, v) \in R\).
2. \(x = x_1u^i\varpi\hat{v}x_2, y = x_1\hat{w}^i\varpi\hat{v}x_2\), for some \(x_1, x_2 \in V^\ast\) and \((u, \alpha/\lambda, v) \in R\).

The language generated by \(\gamma\) is defined by

\[L(\gamma) = \{w \in T^\ast \mid x \Longrightarrow^\ast w, \text{ for some } x \in A\}\]

where \(\Longrightarrow^\ast\) is the reflexive and transitive closure of the relation \(\Longrightarrow\).

Next, we will look into the definition of cut grammars. A cut grammar \(G = (N, \Sigma, S, P)\) where \(N\) is a finite set of non-terminals, \(\Sigma\) is a finite set of terminals, \(S\) is a start symbol and \(P\) is a finite set of productions in \((N \cup \Sigma)^{\ast}(N \cup \Sigma)^{\ast}(N \cup \Sigma \cup \{\hat{\cdot}\})\)^{\ast}. Where \(\hat{\cdot}\) is a new symbol called cut symbol not in \(N\) or \(\Sigma\). The language generated by cut grammar is defined as \(L(G) = \{w \in (\Sigma \cup \{\hat{\cdot}\})^{\ast} \mid S \Longrightarrow^\ast w\}\). Given any string \(w = w_1\hat{w}_2\hat{w}_3\ldots\hat{w}_n\) where \(w_i \in \Sigma^{\ast}\) for \(1 \leq i \leq n\), the cut function \(\hat{w} = \{w_1, w_2, \ldots, w_n\}\) and the uncut function \(\hat{\bar{w}} = w_1 w_2 \ldots w_n\). For a given cut grammar \(G\) and start symbol \(S\), the cut language is defined as \(\tilde{L}(G) = \{\hat{w} \in 2^{\Sigma^{\ast}} \mid S \Longrightarrow^\ast w\}\) and the uncut language \(L(G) = \{\hat{\bar{w}} \in \Sigma^{\ast} \mid S \Longrightarrow^\ast w\}\). A ligation grammar is a cut grammar with an additional new symbol \(\hat{\cdot}\). For any \(w = w_1\hat{w}_2\hat{w}_3\ldots\hat{w}_n\), where \(w_i \in (\Sigma \cup \{\hat{\cdot}\})^{\ast}\) for each \(1 \leq i \leq n\), the ligate function is defined as \(\hat{\bar{w}} = \{\hat{\bar{w}}_1, \hat{\bar{w}}_2, \ldots, \hat{\bar{w}}_n\}\). The ligated language \(\tilde{L}(G) = \{\hat{\bar{w}} \in 2^{\Sigma^{\ast}} \mid S \Longrightarrow^\ast w\}\).

2. Matrix Insertion-Deletion Systems

In this section, we describe Matrix insertion-deletion (in short Matrix ins-del) systems. A Matrix ins-del system is a construct \(\Upsilon = (V, T, A, R)\) where \(V\) is an alphabet, \(T \subseteq V\), \(A\) is a finite language over \(V\), \(R\) is a finite triples of the form \(\{(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n, v_n)\}\), where \((u_k, v_k) \in V^{\ast} \times V^{\ast}\), and \((\beta_k, \alpha_k) \in (V^{\ast} \times \{\lambda\}) \cup \{(\lambda) \times V^{\ast}\}\), with \((u_k, \alpha_k/\beta_k, v_k) \in R_{I_k} \cup R_{D_k} \cup R_{I_k/D_k}\), for \(1 \leq k \leq n\). Here \(R_{I_k}\) denotes the matrix which consists of only insertion rules, \(R_{D_k}\) denotes the matrix which consists of only deletion rules and \(R_{I_k/D_k}\) denotes the matrix which consists of both insertion and deletion rules.

Consequently, for \(x, y \in V^\ast\) we can write \(x \Longrightarrow y\), if \(y\) can be obtained from \(x\) by using a matrix consisting of insertion or deletion or insertion and deletion rules as follows: In a derivation step the rules in a matrix are applied sequentially one after other in order and no rule is in appearance checking (note that the rules in a matrix are not applied in parallel). The language generated by \(\Upsilon\) is defined by \(L(\Upsilon) = \{w \in T^\ast \mid x \Longrightarrow^\ast w, \text{ for some } x \in A\}\), where \(\chi \in \{I_k, D_k, I_k/D_k\}\) where \(\Longrightarrow^\ast\) is the reflexive and transitive closure of the relation \(\Longrightarrow\). Note that the string \(w\) is collected after applying all the rules in a matrix and also \(w \in T^\ast\) only. Note that the above mentioned cut, uncut
3. Modelling Intermolecular structures

In this section, we model some of the intermolecular structures using Matrix insertion-deletion systems.

**Lemma 1** The double strand language \( L_{ds} = \{ u\triangle u^R \mid u \in \Sigma_{DNA}^* \} \) can be modelled by Matrix ins-del system.

**Proof.** The language \( L_{ds} \) can be modelled by Matrix ins-del system \( \Upsilon_{ds} = (\{b, \bar{b}, \checkmark\}, \{b, b, \checkmark\}, \{\checkmark\}, R) \) where \( b \in \{a, t, g, c\}, \checkmark \) is complement of \( b \) and \( R \) is given as: \( R_{I_1} = [\{\lambda, \lambda/b, \checkmark\}, \{\checkmark, \lambda/b, \lambda\}] \). A sample derivation is given as follows:

\[
\downarrow \checkmark \downarrow \Rightarrow R_{I_1} a_{\checkmark} \downarrow \downarrow \Rightarrow R_{I_1} a_{\checkmark} \downarrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \downarrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \downarrow \downarrow \\
\checkmark \checkmark \downarrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \downarrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \downarrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \downarrow \downarrow \\
\end{align*}

In the next lemma, we give a cut grammar which generates nick language and we model such cut grammar using Matrix ins-del system. From Fig.3(b) the nick language can be informally described as \( \{ w_1 \checkmark w_2 \mid \tilde{w}_2 = \tilde{w}_1^R \} \), where \( w_1 \in \Sigma^* \) and \( w_2 \in (\Sigma \cup \{\checkmark\})^* \) (i.e., \( w_2 \) is a string which contains many number of \( \checkmark \)).

**Lemma 2** The nick language \( L_{nl} \) can be generated by Matrix ins-del system.

**Proof.** The language \( L_{nl} \) can be generated by the cut grammar \( G_{nl} = S \rightarrow bSb \mid S\checkmark \mid \checkmark \) for each \( b \in \Sigma_{DNA}^* \). The grammar \( G_{nl} \) can be modelled by Matrix ins-del system \( \Upsilon_{nl} = (\{b, b, \checkmark\}, \{\checkmark\}, R) \) where \( b \in \{a, t, g, c\}, \checkmark \) is complement of \( b \) and \( R \) is given as: \( R_{I_1} = [\{\lambda, \lambda/b, \checkmark\}, \{\checkmark, \lambda/b, \lambda\}], R_{I_2} = [\{\checkmark, \checkmark, \lambda\}], R_{D_1} = [\{\checkmark, \checkmark, \lambda\}] \). A sample derivation is given as follows:

\[
\begin{align*}
\uparrow \checkmark \downarrow \downarrow \Rightarrow & \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \\
\downarrow \downarrow \downarrow \downarrow \Rightarrow & \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \\
\uparrow \checkmark \checkmark \downarrow \downarrow \Rightarrow & \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \Rightarrow \quad R_{I_1} a_{\checkmark} \uparrow \downarrow \\
\end{align*}
\]

In the next two lemmas we model R-loops and Holliday structure as mentioned in Fig. 4 (in Fig 4, \( \checkmark \) denotes \( \diamond \), \( \ast \) denotes \( \lambda \) and \( S, A, R \) represents the non-terminals in their corresponding context-free grammars).

**Lemma 3** The hybrid molecule language (with any number of R-loops) \( L_{rl} = \{ u\checkmark v\checkmark u^R \mid u, v \in \Sigma_{DNA}^* \} \) can be generated by Matrix ins-del system.

**Proof.** The language \( L_{rl} \) can be generated by Matrix ins-del system \( \Upsilon_{rl} = (\{b, b, \checkmark, \checkmark, \checkmark, \checkmark\}, \{b, b, \checkmark\}, \{\checkmark, \checkmark, \checkmark, \checkmark, \checkmark\}, R) \) where \( b \in \{a, t, g, c\}, \checkmark \) is complement of \( b \) and \( R \) is given as: \( R_{I_1} = [\{\lambda, \lambda/b, \checkmark, \checkmark, \checkmark, \checkmark\}, \{\checkmark, \lambda/b, \lambda\}] \), \( R_{I_2} = [\{\lambda, \lambda/b, \checkmark, \checkmark, \checkmark, \checkmark\}] \), \( R_{D_1} = [\{\lambda, \checkmark, \lambda\}] \), \( R_{D_2} = [\{\lambda, \checkmark, \lambda\}] \). A sample derivation is given as follows:
Lemma 4 The Holliday structure $L_{hs} = \{ u_1 \diamond \bar{u}_1 R u_2 \diamond \bar{u}_2 R \ldots u_n \diamond \bar{u}_n R \mid n \geq 0, u_1, \ldots, u_n \in \Sigma_{DNA}^* \}$ can be generated by Matrix ins-del system.

Proof. The language $L_{hs}$ (for n=3) can be generated by Matrix ins-del system $\Sigma_{hs} = \{ \{b, b, \dag t_1, \dag t_2, \dag t_3, \}, \{b, b, \}, \{\dag t_1, \dag t_2, \dag t_3, \}, R \}$ where $b \in \{a, t, g, c\}, b$ is complement of $b$ and $R$ is given as: $R_{t_1} = [\{\lambda, \lambda/b, \dag t_1\}, \{\lambda, \lambda/b, \dag t_2\}, \{\lambda, \lambda/b, \dag t_3\}], R_{t_2} = [\{\lambda, \lambda/b, \dag t_1\}, \{\lambda, \lambda/b, \dag t_2\}, \{\lambda, \lambda/b, \dag t_3\}], R_{t_3} = [\{\lambda, \lambda/b, \dag t_1\}, \{\lambda, \lambda/b, \dag t_2\}, \{\lambda, \lambda/b, \dag t_3\}], R_D_1 = [\{\lambda, \dag t_1, \lambda\}], R_D_2 = [\{\lambda, \lambda/b, \lambda\}]$

$$\Rightarrow_{R_{t_1}} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_{t_2}} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_{t_3}} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_{D_1}} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_{D_2}} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_D_3} a \otimes_1 a \otimes_2 a \otimes_3 a \Rightarrow_{R_{D_4}} a \otimes_1 a \otimes_2 a \otimes_3 a$$

Lemma 5 The replication fork language $\bar{L}_{rf} = \{ u \bar{u} R u v \bar{u} R u R \mid u, v \in \Sigma_{DNA}^* \}$ can be generated by Matrix ins-del system.

Proof. The language $\bar{L}_{rf}$ can be generated by $\Sigma_{rf} = \{ \{\dag t_1, \dag t_2, \dag t_3, \dag t_4, u, t, g, c\}, \{a, t, g, c\}, \{\dag t_1, \dag t_2, \dag t_3, \}, R \}$ where $R$ is given as follows:

$$R_{t_1} = [\{\lambda, \lambda/t, \lambda\}, \{\lambda, \lambda/t, \lambda\}, \{\lambda, \lambda/t, \lambda\}], R_{t_2} = [\{\lambda, \lambda/g, \}, \{\lambda, \lambda/g, \}, \{\lambda, \lambda/g, \}], R_{t_3} = [\{\lambda, \lambda/b, \}, \{\lambda, \lambda/b, \}, \{\lambda, \lambda/b, \}], R_{t_4} = [\{\lambda, \lambda/g, \}, \{\lambda, \lambda/g, \}, \{\lambda, \lambda/g, \}], R_D_1 = [\{\lambda, \lambda/t, \lambda\}, \{\lambda, \lambda/t, \lambda\}, \{\lambda, \lambda/t, \lambda\}], R_D_2 = [\{\lambda, \lambda/b, \lambda\}]. A sample derivation is given as follows:
The generalized linear hybridization $L_{gh}$ can be modelled by Matrix ins-del system.

Proof. The generalized linear hybridization language $L_{gh}$ can be generated by the grammar $G_{gh} = S \rightarrow bSb \mid \diamond A \mid B\diamond \mid \# A \rightarrow \#Ab \mid bSb \mid \# B \rightarrow bB\# \mid bSb \mid \#$ for each $b \in \Sigma_{DNA}$. The grammar $G_{gh}$ can be modelled by Matrix ins-del system $\mathcal{Y}_{gh} = (\{b, \bar{b}, \dagger, \ddagger, \circ, \#\}, \{\{b, \bar{b}, \dagger, \ddagger, \circ, \#\}, \{b \dagger b, \circ b, \# b\}, \{b \dagger b, \circ \ddagger, \# \ddagger\}, R\}$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of $b$ and $R$ is given as:

$$R_{1/D_1} = [(\lambda, \lambda/b \dagger \bar{b}, \ddagger), (\dagger \bar{b}, \dagger t, \circ, \#)], R_{2/D_2} = [(\lambda, \lambda/\dagger t, \ddagger), (\dagger t, \dagger t, \circ, \#)],$$

$$R_{1/D_1} = [(\lambda, \lambda/\ddagger t, , \dagger t), (\ddagger t, \dagger t, \circ, \#)], R_{1/D_2} = [(\lambda, \lambda/\circ t, \ddagger), (\circ t, \dagger t, \circ, \#)],$$

$$R_{2/D_2} = [(\lambda, \lambda/\circ \ddagger, \dagger t), (\circ \ddagger, \dagger t, \circ, \#)], R_{3/D_3} = [(\lambda, \lambda/\circ \circ, \dagger t), (\circ \circ, \dagger t, \circ, \#)],$$

$$R_{4/D_4} = [(\lambda, \lambda/\circ \circ, \circ \ddagger), (\circ \circ, \circ \ddagger, \circ, \#)]$$

A sample derivation is given as follows:

$$\dagger t \equiv R_{1/D_1} \circ a \equiv R_{1/D_1} \circ a \equiv R_{2/D_2} \circ a \equiv R_{3/D_3} \circ a \equiv R_{4/D_4} \circ a
$$

4. New Levels of Ambiguity

In this section, we formally define various ambiguity levels for Matrix ins-del system based on the components used in the derivation such as axiom, contexts, (for deletion/insertion).

Consider the following derivation step in a Matrix ins-del system $\mathcal{Y}$, $\delta : w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_m, m \geq 1$, such that $w_1 \in A$ and the following scenarios can happen (1): $w_k \Rightarrow w_{k+1}$ can be obtained by using a matrix which consists of only insertion rules ($R_{1/D_1}$) (2): $w_k \Rightarrow w_{k+1}$ can be obtained by using a matrix which consists of only deletion rules ($R_{1/D_2}$) (3): $w_k \Rightarrow w_{k+1}$, such that $1 \leq k \leq m$ can be obtained by using a matrix which consists of both insertion and deletion rules ($R_{1/D_3}$). The sequence which consists of used axiom, strings $\alpha_j/\beta_j$ to be deleted/inserted is called as Control Sequence which is given as follows:

$$w_1, \ldots, (\alpha_n/\beta_n), [((\alpha/\beta_1), \ldots, (\alpha/\beta_n)), [(\alpha/\beta_1), \ldots, (\alpha/\beta_n)], \ldots, [\alpha_{m-1}/\beta_{m-1}, \ldots, (\alpha/\beta_n)]$$

More precisely the control sequence means the order in which the strings are deleted/inserted. The sequence which consists of used axiom, strings $\beta_j/\alpha_j$ to be deleted/inserted and the used contexts $(u_j, v_j)$ is called Complete Control Sequence which is given as follows:

$$w_1, [(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n), v_n]\)$$

More precisely the complete control sequence means the order of the contexts used in deletion/insertion rules. Note that one of $\beta_j$ or $\alpha_j$ is empty for all $j$ in the derivation. The position where insertion $\beta_j$ or deletion $\alpha_j$ takes place can be given by the description of $\delta$. 
Definition 1:

1. A Matrix ins-del system \( \Upsilon \), is said to be 0-ambiguous, if there exist at least two different axioms, \( w_1, w_2 \in A \), \( w_1 \neq w_2 \), such that they both derive the same word \( z \), i.e., \( w_1 \implies +z \), \( w_2 \implies +z \).

2. A Matrix ins-del system \( \Upsilon \), is said to be 1-ambiguous, if there are two different ordered control sequences which derive the same word.

3. A Matrix ins-del system \( \Upsilon \), is said to be 2-ambiguous, if there are two different ordered complete control sequences which derive the same word.

4. A Matrix ins-del system \( \Upsilon \), is said to be 3-ambiguous, if there are two different descriptions which derive the same word.

5. Interpretation of Ambiguity in Gene Sequences

In this section, our aim is not to show that there exists inherently ambiguous languages, rather our objective is to present a new interpretation of ambiguity in gene sequences. As an application to the new introduced ambiguity levels, we present some intra and intermolecular structures in gene sequences which exhibits the above introduced new levels of ambiguity for Matrix ins-del systems.

Level 0: The Matrix ins-del system is said to be 0-ambiguous if the same string can be derived from two different axioms.

Definition 2: A string \( w \) over a complementary alphabet \( \Sigma \) is called ideal iff \( |w|_b = |w|_{\bar{b}} \) for all \( b \in \Sigma \). A language is ideal iff it contains only ideal strings.

Lemma 7 The ideal language \( L_{id} \) generated by Matrix ins-del system exhibits level 0 ambiguity.

Proof. The ideal language \( L_{id} \) can be generated by the Matrix ins-del system \( \Upsilon_{id} = (\{b, \bar{b}\}, \{b, \bar{b}\}, \{\lambda\}, R) \), where \( b \in \{a, t, g, c\} \), \( b \) is complement of \( b \) and \( R \) is given as \( R_{I_1} = [(\lambda, \lambda/b, \lambda), (\lambda, \lambda/\bar{b}, \lambda)] \). Consider the gene sequence tactgagcta in ideal language. This sequence can be generated by the Matrix ins-del system \( \Upsilon_{id} \) from two different axioms at and gc such that the same string is obtained at the end of derivation. The two different derivations which differ by axioms are given as follows:

\[
\text{Derivation 1: } at^4 \implies a^4t^4gc \implies actg^4_{\bar{b}} gc \implies actgagc^4_{\bar{b}} \implies tactgagcta
\]

\[
\text{Derivation 2: } gc^4 \implies gc^4_{\bar{b}} gta \implies ta^4g^4cta \implies ta^4tgag^4cta \implies tactgagcata
\]

Level 1: The Matrix ins-del system is said to be 1-ambiguous if there are two different derivations for the same string which differs by the order of string deleted/inserted.

In the next lemma, we show that level 1 ambiguity exists in both intramolecular structure (stem and loop structure) and intermolecular structure (hybrid molecule structure).

Lemma 8 The stem and loop structure \( L_{sl} = \{uvu^R \mid u, v \in \Sigma_{DNA}\} \) represented by Matrix ins-del system obeys level 1 ambiguity.
Proof. The stem and loop structure \( L_{st} \) can be generated by the Matrix ins-del system \( \Upsilon_{sl} = ([\{\bar{b}, b, \hat{t}_1, \hat{t}_2, \hat{t}_3\}, \{b, \bar{b}\}, \lambda, b \hat{t}_1 \hat{t}_3 \hat{t}_2 \hat{b}], R) \), where \( b \in \{a, t, g, c\} \), \( b \) is complement of \( b \) and \( R \) is given as: \( R_{I_1} = [(\lambda, \lambda/b, \hat{t}_1), (\hat{t}_2, \lambda/b, \lambda)], R_{I_2} = [(\lambda, \lambda/b, \hat{t}_3)], R_{D_1} = [(\lambda, \hat{t}_1/\lambda, \lambda), (\hat{t}_2/\lambda, \lambda)], R_{D_2} = [(\lambda, \hat{t}_3/\lambda, \lambda)] \).

Consider the gene sequence \( aag\_tt \) in stem and loop structure. This sequence can be generated in two different ordered control sequences by the Matrix ins-del system \( \Upsilon_{sl} \). Note that the axiom for both sequence is the same. The two sequences are given as follows:

\[
\begin{align*}
\text{Sequence 1} : & a^+ \hat{t}_1 \hat{t}_3 \hat{t}_2 t \implies R_{I_1} a a \hat{t}_1 \hat{t}_3 \hat{t}_2 t t \implies R_{I_2} a a \hat{t}_1 g \hat{t}_3 \hat{t}_2 t t \implies R_{D_1} \\
& a a^g g \hat{t}_3 \hat{t}_2 t t \implies R_{D_2} a a g^g t t \\
\text{Sequence 2} : & a \hat{t}_1 \hat{t}_3 \hat{t}_2 t \implies R_{I_2} a^+ \hat{t}_1 g \hat{t}_3 \hat{t}_2 t \implies R_{I_1} a a \hat{t}_1 g \hat{t}_3 \hat{t}_2 t t \implies R_{D_2} \\
& a a \hat{t}_1 g^g \hat{t}_3 \hat{t}_2 t t \implies R_{D_1} a a^g g^g t t \\
\end{align*}
\]

In sequence 1, the order of strings used for deletion/insertion is \((a, t), g, (\hat{t}_1, \hat{t}_2), \hat{t}_1\). In sequence 2, the order of strings used for deletion/insertion is \(g, (a, t), \hat{t}_3, (\hat{t}_1, \hat{t}_2)\). Thus, we obtain two different ordered control sequences which derive the same gene sequence. Therefore, the Matrix ins-del system \( \Upsilon_{sl} \) obeys level 1 ambiguity. The Level 1 ambiguity can be pictorially represented as shown in Fig.5. Fig. 5(a) corresponds to sequence 1 and Fig. 5(b) corresponds to sequence 2.

![Figure 5](image)

**Figure 5.** Ambiguity in stem and loop structure

**Level 1 ambiguity in Intermolecular structure:** Consider the following string \( g t \otimes g c a c \) in the hybrid molecule structure. This string can be generated in two different ordered control sequences by the Matrix ins-del system \( \Upsilon_{rl} \). The two sequences are given below:

\[
\begin{align*}
\text{Sequence 1} : & \hat{t}_1 \hat{t}_2 \implies R_{I_1} g^t \hat{t}_1 \hat{t}_2 c \implies R_{I_1} g^t \hat{t}_1 \hat{t}_2 c \implies R_{I_2} g t \hat{t}_1 \\
& \otimes g \hat{t}_2 a c \implies R_{I_2} g t \hat{t}_1 \otimes g \hat{t}_2 a c \implies R_{D_1} g t \otimes g c \hat{t}_2 a c \implies R_{D_2} g t \otimes g c \hat{t}_2 a c \\
\text{Sequence 2} : & \hat{t}_1 \hat{t}_2 \implies R_{I_2} g^t \hat{t}_1 \hat{t}_2 \implies R_{I_2} g^t \hat{t}_1 \hat{t}_2 \implies R_{I_1} g^t \hat{t}_1 \hat{t}_2 c \implies R_{I_1} g^t \hat{t}_1 \hat{t}_2 g c \\
& \hat{t}_2 c \implies R_{I_1} g^t \hat{t}_1 \hat{t}_2 g c \hat{t}_2 a c \implies R_{D_2} g t \hat{t}_1 \otimes g c \hat{t}_2 a c \implies R_{D_1} g t \otimes g c a c
\end{align*}
\]

In sequence 1, the order of strings used for deletion/insertion is \((g, c), (t, a), g, c, \hat{t}_1, \hat{t}_2\). In sequence 2, the order of strings used for deletion/insertion is \(g, c, (g, c), (t, a), \hat{t}_1, \hat{t}_2, \hat{t}_1\). Thus, we obtain two different ordered control sequences which derive the same gene
sequence $gt^\ddagger gcac$. Therefore, the Matrix ins-del system $\Sigma_{rl}$ exhibits level 1 ambiguity.

**Level 2**: The Matrix ins-del system is said to be 2-ambiguous if there are two different derivations for the same string which differs by the order of contexts used for deletion/insertion.

**Lemma 9** The Holliday structure $L_{hs} = \{ u_1 \triangleleft u_1^R u_2 \triangleleft u_2^R \ldots u_n \triangleleft u_n^R \mid n \geq 0, u_1, \ldots, u_n \in \Sigma_{DNA} \}$ represented by Matrix ins-del system obeys level 2 ambiguity.

**Proof.** Consider the following string $t^\ddagger ac^\ddagger gg^\ddagger c$ in Holliday structure. This string can be generated in two different ordered complete control sequences by the Matrix ins-del system $L_{hs}$. The two sequences are given below:

**Sequence 1** $\downarrow t_1^- \downarrow t_2^- \downarrow t_3^- \rightarrow R_{l_1} \downarrow t_1^- \downarrow u_2^\ddagger \downarrow t_2^- \downarrow t_3^- \rightarrow R_{l_2} \downarrow t^- \downarrow ac^- t_2^- \downarrow g^- g^- g^- t_3^- \rightarrow R_{D_1} t^\ddagger \downarrow ac^- t_2^- \downarrow g^- t_3^- \rightarrow R_{D_2} t^\ddagger \downarrow ac^- g^- g^- c^- c$

**Sequence 2** $\downarrow t_1^- \downarrow t_2^- \downarrow t_3^- \rightarrow R_{l_2} \downarrow t_1^- \downarrow u_2^\ddagger \downarrow g^- t_3^- \rightarrow R_{D_2} t^\ddagger \downarrow ac^- g^- t_3^- \rightarrow R_{D_1} t^\ddagger \downarrow ac^- g^- g^- g^- c^- c$

In sequence 1, the order of contexts used is $[(\lambda, t_1), (t_1, \lambda), (\lambda, t_3), (t_3, \lambda), (\lambda, \lambda), (\lambda, \lambda), (\lambda, \lambda), (\lambda, \lambda)]$. In sequence 2, the order of contexts used is $[(\lambda, t_2), (t_2, \lambda), (\lambda, \lambda), (\lambda, t_1), (t_1, \lambda), (\lambda, \lambda), (\lambda, t_3), (t_3, \lambda), (\lambda, \lambda), (\lambda, \lambda)]$. Thus, we obtain two different ordered complete control sequences which derive the same gene sequence $t^\ddagger ac^\ddagger gg^\ddagger c$. Therefore, the Matrix ins-del system $L_{hs}$ obeys level 2 ambiguity.

**Level 3**: The Matrix ins-del systems is said to be 3-ambiguous if there are two different descriptions for the same string which differs by the position where the string is deleted/inserted.

**Definition 3**: A string $w$ over a complementary alphabet $\Sigma$ is called orthodox iff it is (i) the empty string $\lambda$, or (2) the result of inserting two adjacent complementary element $bb$, for some $b \in \Sigma$, anywhere in an orthodox string. A language is orthodox iff it contains only orthodox strings.

**Lemma 10** The orthodox language $L_{od}$ generated by Matrix ins-del system obeys level 3 ambiguity.

**Proof.** The orthodox language $L_{od}$ can be generated by the Matrix ins-del system $\Sigma_{od} = \{(b, b), (b, b), (\lambda, R), b \in \{a, t, g, c\}, b$ is complement of $b$ and $R$ is given as $R_{l_1} = [(\lambda, \lambda/bb, \lambda)]$. Consider the string $tagcgcac$ in orthodox language. This string can be derived in two different descriptions by $Y_{od}$ which are given as follows:

**Description 1**: $ge \rightarrow R_{l_1} \tagcgcac \rightarrow R_{l_2} \tagcgcac \rightarrow R_{l_3} \tagcgcac$

**Description 2**: $ge \rightarrow R_{l_2} \tagcgcac \rightarrow R_{l_1} \tagcgcac \rightarrow R_{l_3} \tagcgcac \rightarrow R_{l_4} \tagcgcac$
Note that the axiom, order of insertion of strings, order of contexts (here ($\lambda, \lambda$)) all are same in both derivations, but the position of insertion is different in each derivation. Therefore, the system $\Upsilon_{od}$ is 3-ambiguous.

**Note:** Consider the orthodox string $gcgegcgec$. This string can be derived from axiom $\lambda$ by two different descriptions which are given as follows:

- **Description 1:** $\lambda \Rightarrow gc \Rightarrow gcgc \Rightarrow gcgcgc \Rightarrow gcgcgcgc$
- **Description 2:** $\lambda \Rightarrow gc \Rightarrow gcgc \Rightarrow gcgcgc \Rightarrow gcgcgcgc$

In one derivation the string $gc$ is used one time and $cg$ is used three times. In another derivation the string $gc$ is used four times. The above string obeys level 1 ambiguity but the intermediate gene sequences are same.

### 6. Conclusion

In this paper, we discussed the Matrix insertion-deletion grammar systems and using the system we have modelled some intermolecular structures like double strand languages, nick languages, hybrid molecules (with R-loops), Holliday structure, replication fork, linear hybridization (ligated) languages. We have formally defined various levels of $i$ ($i = 0, 1, 2, 3$)-ambiguity for Matrix insertion-deletion systems. We have given the application for the introduced levels of ambiguity with an interpretation in gene sequences. We witnessed that the many gene sequences in intra and intermolecular structures like ideal, stem and loop, hybrid molecule, Holliday structure, orthodox has a relevance with the introduced levels of ambiguity. Since, in this paper we have shown that ambiguity is possible in gene sequences this research will throw some ideas on how gene sequences in DNA, RNA, protein molecules can be predicted and processed.

### References