Modeling of DNA Single Stage Splicing Language via Yusof-Goode (Y-G) Approach: One String with Two Rules

Wen Li, Lim\textsuperscript{a}, Yuhani Yusof\textsuperscript{a} and Mohammad Hassan Mudaber\textsuperscript{a}

\textsuperscript{a}Faculty of Industrial Sciences and Technology
Universiti Malaysia Pahang, Tun Razak Highway, 26300 Kuantan, Pahang, MALAYSIA

Abstract. Splicing system plays a pivotal role in attempts to recombine sets of double-stranded DNA molecules when acted by restriction enzymes and ligase. Traditional method of finding the result of DNA recombination through experiment is both time and money consuming. Hence, finding the number of patterns of DNA single stage splicing language through formalism of splicing system is a way to optimize the searching process. From the biological perspective, it predicts the number of types of molecules that will exist in the system under existence of restriction enzymes and ligase. In this paper, some theorems, corollaries and examples that lead to the predictions of single stage splicing languages involving one pattern string and two rules are presented via Yusof-Goode approach.

Keywords: DNA Single Stage Splicing Language, Yusof-Goode Splicing System


INTRODUCTION

Cloning and recombinant DNA technologies are rapidly growth in Molecular Biology. In 1987, Head [1] has introduced splicing system, a formal model of recombinant double-stranded DNA molecules (dsDNA) in the existence of restriction enzymes and ligase. Yusof then introduced Y-G splicing system in order to presents the transparent behavior of the DNA biological process.[2] Y-G splicing system will be used throughout the paper to model the behavior of splicing language from one string with two rules. Only sticky ends DNA restriction enzymes will be considered. This is due to any blunt ended DNA can be ligated to any other blunt ended DNA disregard to the sequence of the two molecules, therefore producing infinite splicing languages. In fact, sticky ends which assist the joining of DNA segments with matching protrusions can be ligated much more readily than blunt ended fragments, apparently because of hybridization between the single-stranded regions holds the fragments together in the proper position for ligation [3]. In this paper, the concept of single stage splicing language [8] is introduced and modeled using Y-G rule. This research will be extended more and more towards real biological situation in order to save scientist’s time and money. Theorems and examples will be given in order to give a precise prediction of splicing languages resulting from one string, two palindromic rules based on the rules and strings factor.

PRELIMINARIES

In this section, some fundamental definitions used in this paper are reviewed. The definitions of Y-G splicing system, splicing language and palindromic are stated in the following section.

Definition 1 [4]: (Y-G splicing system, splicing language) A Y-G splicing system \( S = (A, I, R) \) consists of a set of alphabet \( A \) (\( a, g, c \) and \( t \)), an initial set \( I \) of double stranded DNA over \( A \) and a set \( R \) of rules that represents the existing restriction enzymes. If \( r \in R \), where \( r = (u, x, v : y, x, z) \) and \( s_1 = \alpha uxv\beta \) and \( s_2 = \gamma yxv\delta \) are elements of \( I \), then splicing \( s_1 \) and \( s_2 \) using \( r \) produces the initial string \( I \) together with \( \alpha uxz\delta \) and \( \gamma yxv\beta \), presented in either order where \( \alpha, \beta, \gamma, \delta, \alpha, x, v, y \) and \( z \in A^* \) are the free monoids generated by \( A \) with the concatenation...
operation and 1 as the identity element. A language $L$ is a **splicing language** if there exists a splicing system $S$ for which $L=L(S)$.

The palindrome phenomenon is a common occurrence in DNA splicing as well, where the definition is given as below:

**Definition 2** [4]: Palindromic. A string $I$ of dsDNA is said to be **palindromic** if the sequence from the left side of the upper single strand is equal with the sequence from the right side of the lower single strand.

The next definition of single stage splicing language models the set of all molecule types which appear in a test-tube environment with restriction enzymes and ligases that act simultaneously.

**Definition 3**: Single stage splicing language is defined as

$[L_1 = L_1(S)] \equiv \sum_{r=1}^{n} (R_r + I_r + l)$

$R_r =$ set of rules, $1 \leq r \leq n$

$I_r =$ set of initial strings, $1 \leq r \leq n$

$l =$ ligases

Let $S = (A, I, R)$ be the Y-G splicing system. The set of single stage splicing language, $L_1 = L_1(S)$, models the set of all molecule types which appear when all the restriction enzymes, double stranded deoxyribonucleic acid strings and ligases act simultaneously in a single buffer.

**THEOREMS ON ONE STRING WITH TWO RULES**

The followings are the theorems that are significant to the behavior of DNA splicing for one string with two rules. In the first theorem, prediction of single stage splicing languages based on the palindromic and non palindromic properties is presented.

**Theorem 1** In a splicing system that contains one initial string of two rules with each palindromic and non palindromic crossing site respectively; there will be three pattern of single stage splicing language.

**Proof:** Assume that $S = (A, I, R)$ is a Y-G splicing system consisting only one initial string and two rules of palindromic and non palindromic crossing site $r_1, r_2 \in R$ respectively. Let $v = ax_1a_2yxb_1b_2y$ , the rules $r_1, r_2 \in R$ are presented as $R = \{x, a_1a_2, y : x, b_1b_2, y\}$ where $a_1$ with $a_2$ are complement to each other, but $x$ with $y$, $b_1$ with $b_2$ are not complement to each other. $a_1, a_2, x, y, b_1, b_2 \in A^*$. “′” denotes the complement of the string. Therefore, by rotating the sequences of the rules by 180 degree and applying both rules on $I$ will result in $L(S) = \{ax_1a_2yxhb_1b_2y, ax_1a_2x′y′b′b_1b_2y′, b′b_1b_2y′a_2yxhb_1b_2y\}$ (assuming both rules are either 5’ sticky overhang or 3’ sticky overhang). Thus, $L(S) = \{3\}$, three number pattern of splicing languages exist.

**Theorem 2** A one string Y-G splicing system $S = (A, I, R)$ with two palindromic rules generates infinite set of transient languages and three number patterns of infinitely long splicing languages.

**Proof:** Assume $S = (A, I, R)$ is a Y-G splicing system consisting one initial string $I$ and two palindromic rules $r_1, r_2 \in R$. Let $a_1a_2b_1b_2c_1c_2d_1d_2b_1a_2d_2\beta$ be the string in $I \in A^*$. Two cases need to be considered.

i) Two rules with same crossing sites

ii) Two rules with different crossing sites
Case 1: Assume that the crossings of the rules \( r_1, r_2 \in R \) are the same. Thus the rules are presented as \( r_1 = (a_1, b_1, a_2, b_2) \) and \( r_2 = (c_1, b_1, a_2, c_2) \), such that \( a_1 \) with \( b_2 \), \( a_2 \) with \( b_1 \), \( c_1 \) with \( d_1 \) and \( c_2 \) with \( d_2 \) are complement and \( a_1, b_1, a_2, c_2, c_2, d_1, d_2 \in A^* \). Since the crossings of both rules are the same, by applying \( r_1 \) and \( r_2 \) on \( I \), three distinct splicing languages will be obtained as follows: 
\[ \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \alpha^k, \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \alpha^k \beta^k c_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \beta^k \]
for \( k \geq 0 \). The splicing languages will eventually get very long by consuming word \( b_1 a_2 b_2 c_1 d_1 c_2 d_2 \). Thus, three number pattern of infinitely long splicing languages are generated. However, as \( k \) goes to infinity, these three number patterns of infinitely long splicing languages will disappear [3], which resulted in infinite set of transient languages. Let \( P(n) \) equals to the number of recognition sites. This is true for one recognition site from each rules in a string, \( P(1) \) is three infinitely long splicing languages with infinite set of transient languages. Let us consider two recognition sites from each rules in a string, from calculation, the three splicing language is 
\[ \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \alpha^k, \beta^k c_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \beta^k \]
for \( k \geq 0 \). Since splicing \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \alpha^k \) and \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \beta^k \) produce \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^k b_1 a_2 b_2 \alpha^k \beta^k \) [5], all molecules of \( (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \) will eventually be used up thus produce infinite set of transient languages. Hence, \( P(2) = 3 \) three infinitely long splicing languages with infinite set of transient languages. Assuming \( P(n) \) is true, we shall show that \( P(n+1) \) is also true. Let us consider \( n+1 \) recognition sites from each rules, from calculation, the splicing language is 
\[ \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \alpha^{n+1}, \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \beta^{n+1} \]
and \( \beta^k c_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \beta^{n+1} \)
for \( k \geq 0 \). Since splicing \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \alpha^{n+1} \) and \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \beta^{n+1} \) produce \( \alpha a_1 (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \uparrow^{n+1} b_1 a_2 b_2 \alpha^{n+1} \beta^{n+1} \) [2], all molecules of \( (b_1 a_2 b_2 c_1 d_1 c_2 d_2 c_3) \) will eventually use up, producing infinite set of transient languages. \( P(n+1) \) is three infinitely long splicing languages and infinite set of transient languages. According to the principle of mathematical induction, \( P(n) \) is true for \( n \) crossing sites. Therefore, an initial string that contains two palindromic rules with \( n \) crossing sites produces three infinitely long splicing languages and infinite set of transient languages.

Case 2: Assume that the crossings of the rules \( r_1, r_2 \in R \) are the different. Thus the rules are presented as \( r_1 = (a_1, b_1, a_2, b_2) \) and \( r_2 = (c_1, d_1, c_2, d_2) \), such that \( a_1 \) with \( b_2 \), \( a_2 \) with \( b_1 \), \( c_1 \) with \( d_1 \) and \( c_2 \) with \( d_2 \) are complement and \( a_1, b_1, a_2, c_2, c_2, d_1, d_2 \in A^* \). Although the crossings of both rules are different, three distinct splicing languages will still be obtained after applying \( r_1 \) and \( r_2 \) on \( I \) as the word \( b_1 a_2 b_2 \) can relegate with the 180 degree rotation of itself to generate splicing languages as follows: 
\[ \alpha a_1 (b_1 a_2 b_2) \uparrow^k d_1 c_2 d_3 b_1 a_2 b_2 \beta^k, \alpha a_1 (b_1 a_2 b_2) \uparrow^k d_1 c_2 d_3 b_1 a_2 b_2 \beta^k \]
and \( \alpha a_1 (b_1 a_2 b_2) \uparrow^k d_1 c_2 d_3 b_1 a_2 b_2 \beta^k \)
for \( k \geq 0 \). The splicing languages will eventually get very long by consuming word \( b_1 a_2 b_2 c_1 \). Thus, three number pattern of infinitely long splicing languages are generated. However, as \( k \) goes to infinity, these three number patterns of infinitely long splicing languages will disappear [3], which resulted in infinite set of transient languages. This is true for one recognition site from each rules in one string. Let \( P(n) \) equals to the number of recognition sites from each rules in one string. By principle of mathematical induction in Case 1, \( P(n) = 3 \) three infinite long splicing languages and infinite set of transient languages is true for \( n \) crossing sites.

The above two cases lead to the desired result.

**SOME MOLECULAR EXAMPLES OF ONE PATTERN WITH TWO RULES IN Y-G SPICING SYSTEM**

A splicing model given in [4] has shown that one string with two same palindromic rules will produce 3 infinitely long splicing languages, which is parallel to Theorem 2 above. Besides that, two examples are given below by using the theorems proven above.
Example 1: One pattern initial language with two same palindromic crossing sites of palindromic rules

By Theorem 2, infinitely transient languages and three pattern of infinitely long splicing language should be produced. Let Y-G splicing system consists of one initial molecule \( I = \alpha \text{gggcccgggccc} \beta \) where \( \alpha, \beta \in A^* \), and two restriction enzymes NotI and PspOMI with the left cleavage pattern on 5’ overhang, \( r = (gc;ggcc;gc;ggcc,c) \) where \( r \in R, \ c, g \in A \). With the existence of both restriction enzymes, the initial string will split as follows:

\[
A_0 = \alpha GC^\uparrow GGCC \ GCG^\uparrow GGCC \ C\beta \\
\quad \quad \quad \quad \alpha' CG CCGG\uparrow CGC \ CCGG\uparrow CG \beta'
\]

Consequently, the solutions also contain

\[
A_{180} = \beta' G^\uparrow GGCC \ CGC^\uparrow GGCC \ GC \ \alpha' \\
\quad \quad \quad \quad \beta C CCGG\uparrow CG \ CCGG\uparrow CG \ \alpha
\]

Since dsDNA are presented in multiple copies, with the existence of ligase, the above molecules namely \( A_0 \) and \( A_{180} \) will relegate and forming new molecules as follows:

\[
L_1 = \alpha \ G \ (GGCC \ CGC \ \cup \ \GGCC \ GCG)^\circ \ \GGCC \ GC\beta \\
\quad \quad \quad \quad \alpha' CG CCGG \ (GCG \ CCGG \ \cup \ CGC \ CCGG)^\circ \ CG \beta'
\]

\[
L_2 = \alpha \ G \ (GGCC \ CGC \ \cup \ \GGCC \ GCG)^\circ \ \GGCC \ Ca' \\
\quad \quad \quad \quad \alpha' CG CCGG \ (GCG \ CCGG \ \cup \ CGC \ CCGG)^\circ \ Ga
\]

\[
L_3 = \beta \ G \ (GGCC \ CGC \ \cup \ \GGCC \ GCG)^\circ \ \GGCC \ C\beta \\
\quad \quad \quad \quad \beta C CCGG \ (GCG \ CCGG \ \cup \ CGC \ CCGG)^\circ \ G\beta'
\]

Three types of infinitely long conceivable molecules, namely splicing languages are produced. Nevertheless, these molecules will disappear by growing too long, leaving infinitely transient languages, as stated in the theorem.

Example 2: One initial language with one palindromic and one non palindromic rules

By theorem 1, three pattern of splicing language should be produced. Let Y-G splicing system consists of one initial languages \( I = \alpha \text{ccggcacgag} \beta \) where \( \alpha, \beta \in A^* \), and same pattern different crossing site restriction enzyme with one palindromic rules and another non palindromic rules HpaII and BssSI, \( r = (c;cg,g;c;acga,g) \) where \( r \in R, \ c, g, a, t \in A \).

With the existence of both restriction enzymes, the initial string will split as follows:

\[
A_0 = \alpha \ C^\uparrow CG \ G \ C \ ACGA \ G\beta \\
\quad \quad \quad \quad \alpha' G \ GC\uparrow C \ G \ TGCT\uparrow C \beta'
\]

Consequently, the solutions also contain

\[
A_{180} = \beta' \ C^\uparrow TCGT \ GC^\uparrow CG \ G \ \alpha' \\
\quad \quad \quad \quad \beta \ G \ AGCA\uparrow CG \ GC\uparrow C \ \alpha
\]
With the existence of ligase, the above molecules namely $A_6$ and $A_{80}$ will relegate and forming molecules as follows:

\[
L_1 = \alpha' C G G C ACGA G \beta \\
\alpha' G G C G TGCT C \beta'
\]

\[
L_2 = \alpha CCG G \alpha' \\
\alpha' G GCC \alpha
\]

\[
L_3 = \beta' C TCGT GC G G C ACGA G \beta \\
\beta G AGCA CG GC G G TGCT C \beta'
\]

Three single stage splicing language are generated, as stated in the theorem.

CONCLUSIONS

In this paper, two theorems on predicting splicing languages involving one string with two rules are proven. The two examples given using restriction enzymes collected from New England Biolabs [6] do behave as theorems provided. Future work will include using methods of a limit adjacency matrix with computational model to predict the behavior of splicing languages in the equilibrium state of the system.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Ministry of Education (MOE) and Research and Innovation Department, Universiti Malaysia Pahang (UMP) for the financial funding through UMP Research Grant Vote No: RDU 130354 and RAGS Grant Vote No: RDU131404.

REFERENCES

3. E. Goode and D. Pixton, Splicing to the limit. In: Janoska, N., paun, Gh. And Rozenberg, G. eds. Lecture Notes in Computer Science: Springer-Verlag. 189-201; 2004