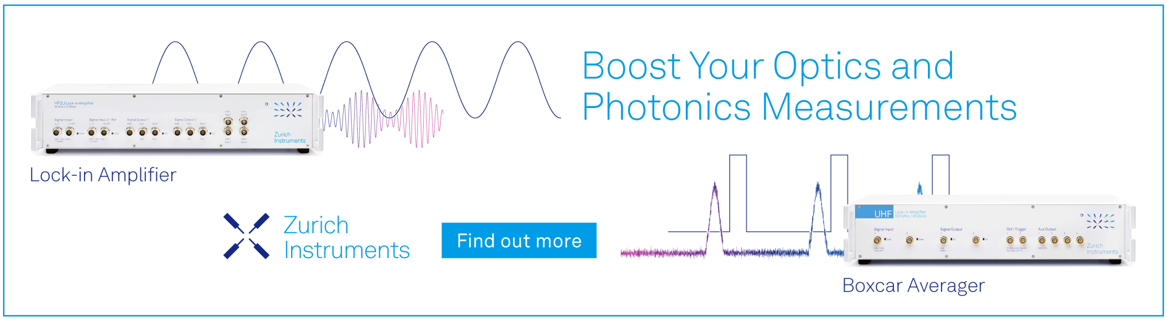
AIP Conference Proceedings

RESEARCH ARTICLE | MARCH 07 2024

The representation of *n*-cutting site splicing languages for a single string with palindromic rule via de Bruijn graph

Nooradelena Mohd Ruslim ✉; Yuhani Yusof; Mohd Sham Mohamad

AIP Conf. Proc. 2895, 080008 (2024)
https://doi.org/10.1063/5.0192296



AIP Publishing

The Representation of *n*-cutting Site Splicing Languages for a Single String with Palindromic Rule via de Bruijn Graph

Nooradelena Mohd Ruslim[a)], Yuhani Yusof[b)] and Mohd Sham Mohamad[c)]

Centre for Mathematical Sciences,
College of Computing and Applied Sciences,
Universiti Malaysia Pahang, Lebuhraya Persiaran Tun Khalil Yaakob,
26300 Kuantan, Pahang, Malaysia.

Corresponding author: [a)]PSE20003@student.ump.edu.my, [b)]yuhani@ump.edu.my
[c)]mohdsham@ump.edu.my

Abstract. Yusof-Goode (Y-G) splicing system proposed by Yusof illustrates the deoxyribonucleic acid (DNA) splicing process in a luminous way, where the rule is written in a double triple notation. In the process of generating splicing languages, DNA molecules are cut and pasted, based on the reaction of the molecules with restriction enzyme and ligase. The resulting molecules, either in original or new molecules can be classified mathematically in the form of limit, adult or transient languages. Conducting a wet lab experiment to produce splicing languages is a huge investment, thus mathematical approach is one of the best options in predicting the outcome on splicing languages when *n*-cutting site exist in a splicing process. Besides the generation of splicing languages through splicing system, a meaningful representation of the languages can be portrayed in a graph. In this paper, the *n*-cutting site splicing languages generated from Y-G splicing system containing single string with 2-cutting site, with palindromic rule and palindromic crossing site is investigated. The features of left and right contexts of the rule are considered in the two cases discussed in this paper. Besides, two new definitions namely connectors and sub-connectors are given and are adopted in the generation of de Bruijn graph. With the constructed de Bruijn graph for the resulting splicing languages from both cases, a theorem is proposed.

INTRODUCTION

Deoxyribonucleic acid (DNA) is genetic material that carries information necessary for an organism to develop and function. Its' double helix form consists of nucleotides with the ladder of the curving sides representing the sugar phosphate that ties the two DNA strands. The nitrogenous bases are classified into two classes namely purines and pyrimidines. The strands contain purines namely adenine (*A*) and guanine (*G*), and pyrimidines namely cytosine (*C*) and thymine (*T*). These purines and pyrimidines are paired together, following specific rules which are [*A*/*T*] or [*T*/*A*] and [*C*/*G*] or [*G*/*C*] [1]. The hydrogen bonds that tie these nitrogenous bases can be broken due to the reaction of restriction enzymes in certain DNA molecules. A specific sequence in restriction enzyme called recognition site will cleave DNA strands when the sequence is detected in the strand. A cut by restriction enzyme at the specific places will produce either sticky ends or blunt ends to the original DNA strand. For example, enzyme *EcoR*I which is in the sequence of $(g; aatt, c)$ will cleave as follows, producing $5'$ overhang string:

$$\begin{array}{c} G^{\blacktriangledown}AATTC \\ CTTAA_{\blacktriangle}G \end{array}$$

3rd International Conference on Applied & Industrial Mathematics and Statistics 2022 (ICoAIMS2022)
AIP Conf. Proc. 2895, 080008-1–080008-9; https://doi.org/10.1063/5.0192296
Published by AIP Publishing. 978-0-7354-4820-9/$30.00

The cutting and pasting with the existence of restriction enzymes and ligase will then produce new DNA molecules and the initial strand itself. Other than investigating this process biologically, the splicing process can be depicted in a splicing system. This began in the pioneer work by Head [2] called Head splicing system and recently by Yusof [3], namely Yusof-Goode (Y-G) splicing system. The Y-G splicing system is motivated by the Head splicing system. It was developed with a modification in rule notation. Thereafter, motivated by the Y-G splicing system, many researchers had adopted this splicing system in presenting the language generated by adopting various mathematical methods, for example, Lim [4] used limit adjacency matrix and Mudaber [5] used de Bruijn graph in presenting the splicing languages. The purpose of presenting the language in graph is based on the finding that the double stranded DNA (dsDNA) which are in three-dimensional structure [6] is found to be better presented in graphical form [7]. This is conspicuously seen from several research that applied various mathematical approach in presenting the splicing languages. Yusof [8,9] adopted limit graph, to show the existence of limit languages and new type of splicing languages namely active persistent languages. In the same limit graph, the existence of transient languages was clearly shown. Additionally, Razak in 2021 applied cycle graph and its complement, to study the behavior of cuts at various edges and DNA recombination [10].

Hence, this paper aims to show the produced languages in a graphical representation by using de Bruijn graph. Mudaber [11] in 2016 applied the de Bruijn graph to show two stages splicing languages with persistent and permanent characteristics that resulted in the languages. Furthermore, de Bruijn graph is also applied in metagenomic data classification [12], genome assembly [13] and indexing data structures [14]. Whereas in this research, the exploration of de Bruijn graph in representing the splicing languages is meant to illustrate the infinitely long molecules produced as the effect of *n*-cutting in a string. The features of left and right contexts of the rule, where Case I will consider a different left and right context of rule, while in Case II, a same left and right context of rule is considered. These two cases are considered to investigate the outcome of the features of the rule on the splicing languages.

This paper includes four sections, starts with the introduction. In the second section, some important definitions are stated, followed by the results and discussion which are disclosed in the third section, with the proposition of theorem and its proving by considering two cases namely Case I and Case II. In addition, this section also includes de Bruijn graph representation of the generated languages. The final section of the report summarizes the findings.

PRELIMINARIES

A few definitions that are used throughout this paper are included in this section. In this paper, the splicing process on a string with palindromic rule and palindromic crossing site are considered. In addition, the effect of left and right contexts towards the generated languages is studied. The splicing languages are then represented in the de Bruijn graph. In this paper, the splicing process is conducted based on Y-G splicing system; thus, its definition is first given as follows:

Definition 1 [3]: Yusof-Goode (Y-G) Splicing System
Let $S = (A, I, R)$ be a Y-G splicing system, containing a set of alphabets A, a set of initial strings $I \in A^*$ and a set of rules, $r \in R$ that is applied on DNA strings and can be either in left pattern, $r = (a; x, b: c; x, d)$, in right pattern, $r = (a, x; b: c, x; d)$ or in both patterns, $r = (a, x, b: c, x, d)$. For $s_1 = \alpha a x b \beta$ and $s_2 = \gamma c x d \delta$, were s_1 and s_2 are elements in I, then the languages generated will be in the form of $\alpha a x d \delta$ and $\gamma c x b \beta$, together with its initial string I, where $\alpha, \beta, \gamma, \delta, a, b, c, d$ and $x \in A^*$ are the free monoids generated by A with the concatenation operation and 1 as the identity element. Then L is a splicing language if there exists a splicing system S for which $L = L(S)$.

Then, definition of single stage splicing language is stated.

Definition 2 [15]: Single Stage Splicing Language
If $S = (A, I, R)$ is a Y-G splicing system, then the set of all molecule types that will appear when all restriction enzymes, double stranded deoxyribonucleic acid (dsDNA) strings and ligases act simultaneously in a single buffer are known as the set of single stage splicing language, $L_1 = L_1(S)$.

Next, the definition of palindromic is provided.

Definition 3 [8]: Palindromic
Any initial string I of dsDNA is in a palindromic sequence if the reading of the upper string from left to right is equal to the reading of the sequence of the lower string from right to left.

The palindromic characteristics can also be found in the sequence of restriction enzyme and crossing site of restriction enzyme. For example, enzyme *Acc65*I is a palindromic enzyme with palindromic crossing site, $(g; gtac, c)$ while enzyme *Aci*I is a non-palindromic enzyme with palindromic crossing site, $(c; cg, c)$.

In addition, the definition of k-mers, de Bruijn graph, prefix and suffix that will be adopted in the corresponding graphs in this paper are given.

Definition 4 [16]: k-mers
A string of length k over the alphabet $A = \{a, c, g, t\}$ is called a k-mers where $k \in \mathrm{N}$.

Definition 5 [16]: de Bruijn Graph
The de Bruijn graph is a directed graph, such that, for a set of k-mers, the nodes are exactly k-mers and the edges are given by a direct relation.

Definition 6 [17]: Prefix and Suffix
A substring of w refers to any string of continuous symbols in some word w. Given $w = vu$, then the substrings v and u are said to be a prefix and a suffix of w, respectively. For example, if $w = xyyxy$, then $\{\lambda, x, xy, xyy, xyyx, xyyxy\}$ is the set of all prefixes of w, while $\{y, xy, yxy, yyxy\}$ are some of its suffixes.

The new graph that contains all vertices and edges of two or more graphs that are combined is called a union of graphs [18]. In this paper, the de Bruijn graph developed for each case will then be represented in the form of union graphs.

Finally, the definition of initial and terminal vertices that will be used in the development of graph in this paper is given as follows:

Definition 7 [18]: Initial and Terminal Vertices
In a directed graph or digraph G, there is a set of vertices V, together with a set of edges E or can be written as $G = (V, E)$. For the edge (u, v), the vertex u is called the initial vertex while the vertex v is called the terminal vertex of this edge.

RESULTS AND DISCUSSION

This section will illustrate the generated splicing languages in the form of de Bruijn graph. This paper aims to show the effect of 2-cutting site in a string on the generated languages, with distinct characteristics set on the rule, concerning its left and right contexts. Prior to the discussion on the cases, two new definitions are given.

Definition 8: **Connectors**
Connectors are vertices that link by edge containing the sequence of restriction enzyme that connect the initial and terminal vertices.

Definition 9: **Sub-connectors**
Sub-connectors is a set of vertices that link the connectors, where the number of sub-connectors will increase corresponds to the increases in the length of the restriction enzyme and will be resulted in the composition of the string (k-mers), its prefix and suffix.

Next, to develop the de Bruijn graph that corresponds to the cases in this paper, Definition 7, 8, and 9 are visualized in Figure 1.

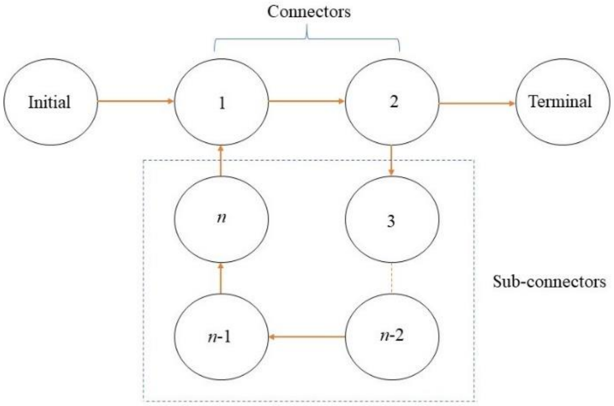


FIGURE 1. de Bruijn graph containing initial, terminal, connectors and sub-connectors

Based on Fig. 1, the number of initial and terminal vertices, connectors and sub-connectors are based on the recombinant DNA string. The composition of the string (k-mers) is reflected in the formation of prefixes and suffixes of the string. As stated in Definition 8, the connectors can be identified by the edge that contains the sequence of the rule while the number of sub-connectors will depend on the length of the respective rule that eventually will be resulted in the number of prefixes and suffixes. The difference between the length of rule applied in the splicing system will be shown in the two cases discussed in this section namely Case I and Case II. In these two cases, its recombinant DNA strands are presented in de Bruijn graph. Hence, for each case, de Bruijn graphs will be constructed based on the generated splicing languages, showing repetition of middle segment of the languages for $k = 1, 2$ and 3. Additionally, the graph will show the existence of infinitely long molecules, as k increases. Beforehand, a theorem that is related to the cases discussed in this paper is stated below.

Theorem 1. [19] Let $S = (A, I, R)$ be a Y-G splicing system. If $I = \{s\}$ is a set of non-palindromic initial string and the element of set $R = \{r\}$ is a palindromic rule which contains palindromic crossing site of more than one cutting site, then $n\big(L_1(S)\big) > 3$, in the sequence of $\alpha - \beta, \alpha - \alpha', \beta' - \beta$ with infinitely long molecules.

To make it clear, more than one cutting site in Theorem 1 [19] refers to the number of cutting sites that existed in a string. In this paper, the generated splicing languages obtained from Theorem 1 [19] are then presented via de Bruijn graph. Before complete proof is given, some cases need to be discussed first. Suppose $S = (A, I, R)$ is a Y-G splicing system, then a splicing process for a single string $I \in A^*$ containing 2-cutting sites, a palindromic rule, $r \in R$ with palindromic crossing sites is considered. Hence, two cases are discussed, intending to present the splicing languages in the form of de Bruijn graph. For Case I, a different left and right context of rule is regarded. While in Case II, the features of same left and right context of rule are observed. These cases are intended to see if the features of the rule will affect the generated languages, in terms of the sequences and its type.

Case I: Modelling of de Bruijn graph for a Single String with a Palindromic Rule and Palindromic Crossing Site with Different Left and Right Contexts

In this case, single stage splicing languages produced by Y-G splicing system consisting of an initial string and a palindromic rule with palindromic crossing site is presented via de Bruijn graph. This case assumes different left and right contexts in a rule.

Let $s = \alpha axybaxyb\beta$ with $r = (a, xy, b{:}\, a, xy, b)$. With a different left and right contexts of the rule and x is complement with y and a is complement with b, $\forall \alpha, \beta, a, b, x, y \in A^*$, generated splicing languages of this splicing process is in the form of:

$$L_1(S) = \{\alpha axyb\beta, \alpha axybaxyb\beta, \alpha axybaxybaxyb\beta, \alpha axyb\alpha', \alpha axybaxyb\alpha', \alpha axybaxybaxyb\alpha', \\ \beta' axyb\beta, \beta' axybaxyb\beta, \beta' axybaxybaxyb\beta, \dots\}$$

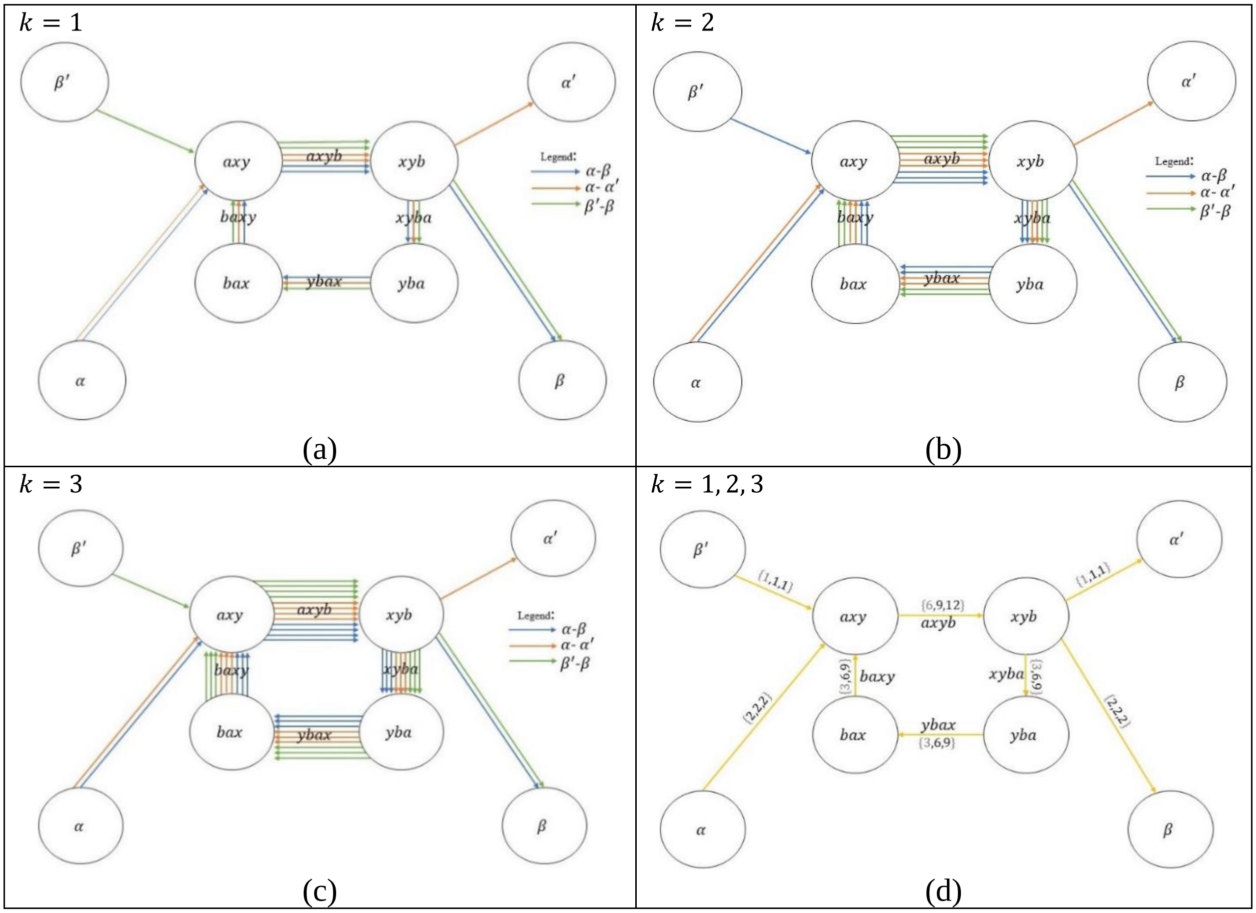
Thus, the generated splicing languages with $k \geq 0$ and $k \in Z^+$ can be generalized as follows:

$$L_1(S) = \{\alpha a(xyba)^k xyb\beta, \alpha a(xyba)^k xyb\alpha', \beta' a(xyba)^k xyb\beta\}.$$

Next, for the purpose of representing graph, splicing languages in the sequence of $\alpha - \beta$, $\alpha - \alpha'$ and $\beta' - \beta$ with $k = 1, 2, 3$ will be considered, that is:

$$L_1(S) = \{\alpha axybaxyb\beta, \alpha axybaxybaxyb\beta, \alpha axybaxybaxybaxyb\beta, \alpha axybaxyb\alpha', \alpha axybaxybaxyb\alpha', \\ \alpha axybaxybaxybaxyb\alpha', \beta' axybaxyb\beta, \beta' axybaxybaxyb\beta, \beta' axybaxybaxybaxyb\beta\}$$

Following that, to model the splicing languages on de Bruijn graph, the length of restriction enzyme is considered. For rule $r = (a, xy, b: a, xy, b)$, the corresponding composition DNA splicing languages is 4-mers. Based on 4-mers compositions that reflect the edges of de Bruijn graph, the vertices of the de Bruijn graph can be obtained by separating the compositions to its prefix and suffix. For graph representation, the graphs are constructed for $k = 1$, $k = 2$ and $k = 3$ only and the union of the graphs is developed for $k = 1, 2$ and 3. Based on Definitions 7–9, the above splicing languages can be represented in the de Bruijn graph as follows:



(a)

(b)

(c)

(d)

FIGURE 2. de Bruijn graphs for (a) $k = 1$, (b) $k = 2$, (c) $k = 3$ and (d) the union of de Bruijn graphs, $k = 1, 2, 3$

Next, in Case II, the single stage splicing languages that are generated after reaction between a single string containing 2-cutting site, with a palindromic rule and palindromic crossing site is then presented in de Bruijn graph. Features of same left and right context of the rule will be observed.

Case II: Modelling of de Bruijn Graph for a Single String with a Palindromic Rule and Palindromic Crossing Site with Same Left and Right Contexts

In this case, single stage splicing languages produced by Y-G splicing system consisting of one initial string and one palindromic rule with palindromic crossing site is presented via de Bruijn graph. This case considers the same left and right contexts in a rule.

Assume $s = \alpha abxyababxyab\beta$ with $r = (ab, xy, ab: ab, xy, ab)$ where the same left and right context of the rule is considered. Given a is complement with a', b with b', x with x' and y with y', $\forall \alpha, \beta, a, b, x, y \in A^*$. The generated splicing languages of this splicing process is in the form of:

$$\begin{gathered} L_1(S) = \{\alpha abxyab\beta, \alpha abxyababxyab\beta, \alpha abxyababxyababxyab\beta, \alpha abxyab\alpha', \\ \alpha abxyababxyab\alpha', \alpha abxyababxyababxyab\alpha', \\ \beta' abxyab\beta, \beta' abxyababxyab\beta, \beta' abxyababxyababxyab\beta, \dots\} \end{gathered}$$

Thus, the generated splicing languages with $k \geq 0$ and $k \in Z^+$ can be generalized as follows:

$$L_1(S) = \{\alpha ab(xyabab)^k xyab\beta, \alpha ab(xyabab)^k xyab\alpha', \beta' ab(xyabab)^k xyab\beta\}.$$

However, for graph representation, splicing languages in the sequence of $\alpha - \beta$, $\alpha - \alpha'$ and $\beta' - \beta$ with $k = 1, 2, 3$ will be considered that is:

$$\begin{aligned} L_1(S) = \{&\alpha abxyababxyab\beta, \alpha abxyababxyababxyab\beta, \alpha abxyababxyababxyababxyab\beta, \\ &\alpha abxyababxyab\alpha', \alpha abxyababxyababxyab\alpha', \alpha abxyababxyababxyababxyab\alpha', \\ &\beta' abxyababxyab\beta, \beta' abxyababxyababxyab\beta, \beta' abxyababxyababxyababxyab\beta\} \end{aligned}$$

To model these splicing languages on de Bruijn graph, the composition of DNA splicing languages is 6-mers which reflects the length of rule $r = (ab, xy, ab: ab, xy, ab)$. Based on 6-mers compositions, the vertices of de Bruijn graph can be obtained by separating the compositions to its prefix and suffix. For graph representation, the graphs are constructed for $k = 1$, $k = 2$ and $k = 3$ and its respective union of de Bruijn graph. Based on Definitions 7–9, the above three splicing languages can be represented in de Bruijn graph as given in Figure 3.

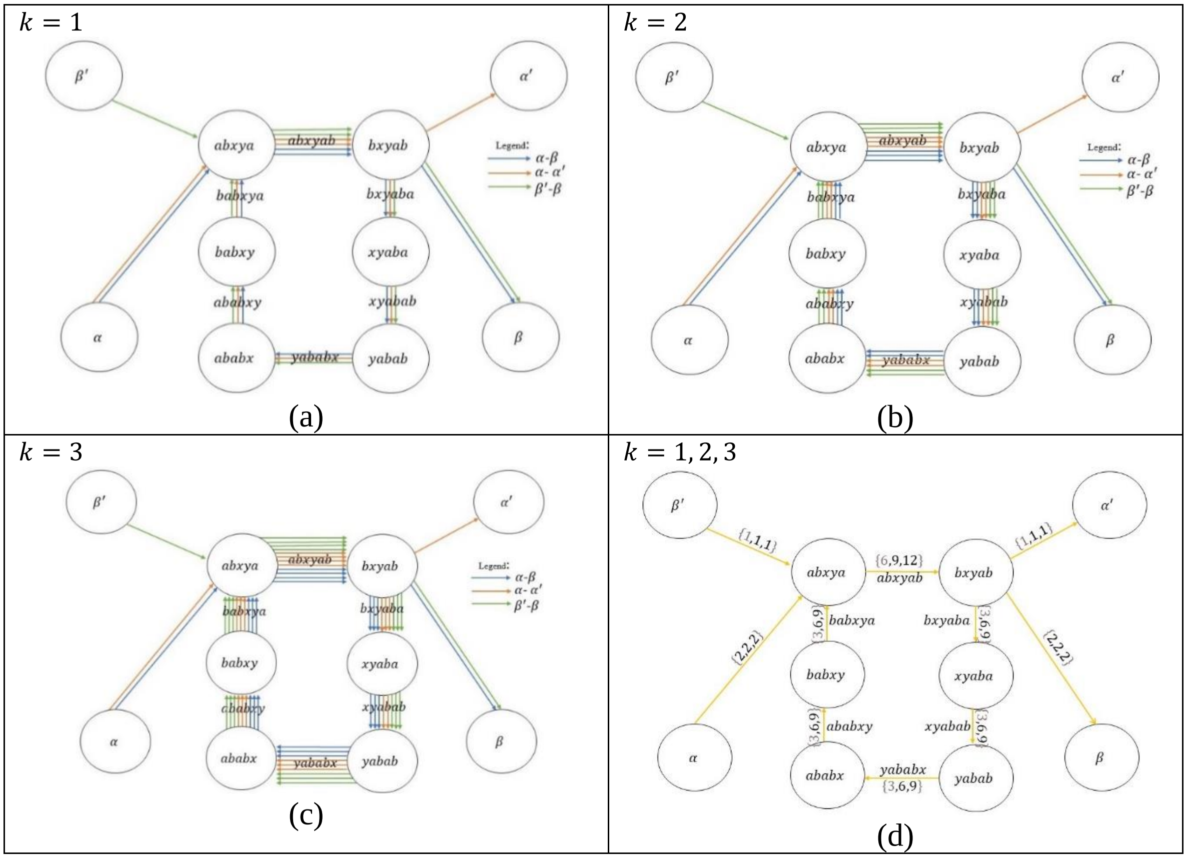


FIGURE 3. de Bruijn graphs for (a) $k = 1$, (b) $k = 2$, (c) $k = 3$ and (d) the union of de Bruijn graphs, $k = 1, 2, 3$

Both cases lead to the desired result, hence the model is shown.

Based on Fig. 2, the generated splicing languages with 4-mers compositions can be presented in de Bruijn graph, containing initial and terminal vertices, with 2 connectors and 2 sub-connectors. As given in Definitions 8 and 9, the middle segment of the de Bruijn graph consists of connectors and sub-connectors. It is observed that the repetition of each edge will increase as k increases. This can be seen by observing the set of edges for $\alpha - \beta$, $\alpha - \alpha'$ and $\beta' - \beta$ sequences. For example, for $k = 1$, the edge between the connectors containing the sequence of rules, $axyb$ repeat two times. Next, for $k = 2$, the repetition becomes three times, which increases the frequency of edge of the connectors. Also, for $k = 3$, the frequency of edge of the connectors increase to four. Similarly, based on Fig. 3, the splicing languages with composition of 6-mers then resulted in initial and terminal vertices, with 2 connectors and 4 sub-connectors. The repetition in the middle segment of de Bruijn graph follows the same trend as in Fig. 2, which it repeats two, three and four times for $k = 1, 2, 3$ respectively.

Next, by using the union of graphs [18], the de Bruijn graph for iteration $k = 1$, $k = 2$ and $k = 3$, can be performed. It is observed that the frequency of all edges increases accordingly. This shows that when k increases, the repetition of the middle segment will get longer, producing infinitely long molecules. Goode and Pixton [20] suggested that, a converging to infinitely long molecules will result in disappeared molecules or known as transient languages.

Additionally, from Fig. 2 and 3, it is observed that the n-cutting site exists in a string will affect the generated splicing languages especially at the middle segment of the languages. If the value of k is increased in the iteration process, then the middle segment of splicing languages will duplicate more and consequently the edges in de Bruijn graph will repeat more, forming infinitely long molecules. Then, from Fig. 2 and 3, it is perceived that the value of k-mers also affects the number of vertices in the middle segment, if the string only contains the rules' alphabet for the middle segment. The number of sub-connectors is found to be dependent on the value of k-mers. For example, in Fig. 2 that uses 4-mers, contains 2 sub-connectors, while in Fig. 3, for 6-mers, contains 4 sub-connectors. Also, with rise in the iteration of k, this leads to formation of longer molecules. This shows that when k increases, the repetition of the middle segment will get longer, producing infinitely long molecules which will disappear and known as transient languages [20]. By using the same approach, the following results are obtained:

TABLE 1. Relation between k-mers with connectors and sub-connectors

Length of rule	k-mers	Connectors	Sub-connectors
4	4-mers	2	2
6	6-mers	2	4
8	8-mers	2	6
10	10-mers	2	8

From Table 1, it is clearly shown that, if given n is the length of the rule and a_n is the number of sub-connectors, then by using recurrence relation:

$$\begin{gathered} n = 4, a_n = 4 - 2 \\ n = 6, a_n = 6 - 2 \\ n = 8, a_n = 8 - 2 \\ \vdots \\ n = k, a_k = k - 2 \end{gathered}$$

for $k \in \mathrm{N}$, $n \in \mathrm{N}$ and $n \geq 4$.

Hence, the following theorem is obtained:

Theorem 2. Assume n is the length of rule for $n \geq 4$. When a recombinant DNA string is represented in de Bruijn graph, there will exist n-mers and $(n - 2)$ sub-connectors.

Proof Let n is the length of rule for $n \geq 4$. Without loss of generality, let $n = 2k$ where $k = 2, 3, 4, \ldots$ since our consideration is only for n even. By induction, let the first case which $n = 4$ or equivalently when $k = 2$, then it is clear that there exists $n - 2 = 2k - 2 = 4 - 2 = 2$ sub-connectors exist as given in Table 1. Now, assume that for some n length of rule, then there exist $n - 2$ sub-connectors are true or similarly for some k, there exist $2k - 2$ sub-connectors is true. Now for $2(k + 1)$ length or rule, there exist $2(k + 1) - 2 = 2k + 2 - 2 = 2k$ number of sub-connectors which proves that for a string with length n represented in de Bruijn graph, then there exist $(n - 2)$ sub-connectors. ■

From the presented de Bruijn graphs in Case I and II, it is observed that there exist k-mers of the sequence of respective restriction enzymes. This implies that the splicing languages are in the type of transient languages, which corresponds to the edges that link the connectors. It is also observed that, in the case of palindromic rule and palindromic crossing site, the features of either using same or different left and right contexts, does not affect the generated languages. The DNA recombinant process is dependent on the crossing site and not the left or right context.

CONCLUSION

In this paper, one theorem is given concerning the relation between the length of restriction enzyme and the number of sub-connectors generated in the de Bruijn graph. In the formation of de Bruijn graph in Case I and Case II, two new definitions namely connectors and sub-connectors are adopted in the representation of splicing languages. The pattern of the de Bruijn graph in Fig. 2 and 3, supports the suggestion by Goode and Pixton [20] that the generation of infinitely long molecules will disappear and will result in infinite set of transient languages. The outcome hence supports the objective of this paper which is to demonstrate the effect of 2-cutting site in a string that produce n-cutting site splicing languages and later can be translated in the form of de Bruijn graph. For future research, other features of restriction enzymes can be observed to see their effects on the generated splicing languages and their respective de Bruijn graph.

ACKNOWLEDGMENTS

All authors are grateful to Universiti Malaysia Pahang for the financial funding under UMP Fundamental Research Grants through RDU220344.

REFERENCES

1. P. J. Russell, *iGenetics A Molecular Approach*, 3rd. ed. (Benjamin Cummings, London, 2010), pp. 41–43.
2. T. Head, Bull. Math. Biol. **49**, 737–759 (1987).
3. Y. Yusof, N. H. Sarmin, M. Mahmud, T. E. Goode and W. H. Fong, "An Extension of DNA Splicing System" in 6th International Conference on Bio-Inspired Computing: Theories and Applications, Pulau Pinang, Malaysia, 2011, pp. 246- 248.
4. W. L. Lim, "Single stage DNA splicing system via Yusof-Goode approach," M.Sc. thesis, Universiti Malaysia Pahang, 2015.
5. M. H. Mudaber, "Persistency and permanency of two stages splicing languages based on DNA recombination process by using Yusof-Goode (Y-G) approach," M.Sc. thesis, Universiti Malaysia Pahang, 2015.
6. R. F. Weaver, *Molecular Biology*, 5th. ed. (McGraw-Hill, New York, 2012), pp. 41-45.
7. N. M. Ruslim, M. Elizabeth, Y. Yusof, M. S. Mohamad and N. Adzhar, J. Phys.: Conf. Ser. **1988**, 012081 (2021).
8. Y. Yusof, "DNA splicing system inspired by bio molecular operations," Ph.D. thesis, Universiti Teknologi Malaysia, 2012.
9. Y. Yusof, W. L. Lim, T. E. Goode, N. H. Sarmin, W. H. Fong and M. F. A. Wahab, in *AIP Conference Proceedings*, Vol. **1660** (AIP Publishing, 2015) 050045.
10. M. N. S. A. Razak, W. H. Fong, and N. H. Sarmin, J. Phys.: Conf. Ser. **1988**, 012067 (2021).
11. M. H. Mudaber, Y. Yusof, M. S. Mohamad, A. N. M. Ramli, and W. L. Lim, J. Tekn. **78**, 73-78 (2016).
12. M. S. Kamal, S. Parvin, A. S. Ashour, F. Shi, and N. Dey, Int. J. of Tech. **9**, 59-75 (2017).
13. Y. T. Huang and C. F. Liao, Bioinformatics **32**, 1301-1307 (2016).
14. B. Cazaux, T. Lecroq, and E. Rivals, J. Comp. Sys. **104**, 165-183 (2019).
15. W. L. Lim, Y. Yusof, and M. H. Mudaber, in *AIP Conference Proceedings*, Vol. **1643** (AIP Publishing, 2015) 695-699.
16. R. Chikhi, A. Limmaset, S. Jackman, J. T. Simpson, and P. Medvedev, J. Comp. Biol. **22**, 336-352 (2015).
17. P. Linz, *An introduction to formal languages and automata*, 6th. ed. (Jones and Barlett Publisher, USA, 2016), pp. 233-267.
18. K. H. Rosen, *Discrete mathematics and its applications*, 8th. ed. (McGraw Hill, New York, 2018), pp. 641-655.
19. N. M. Ruslim, Y. Yusof, and N. Adzhar, J. Phys.: Conf. Ser. **2157**, 012033 (2022).
20. E. Goode and D. Pixton, "Splicing to the Limit," in *Aspects of Molecular Computing, Lecture Notes Computer Science*, edited by Janoska, N. et al., (Springer-Verlag, Germany, 2004), pp. 189-201.