

Block inversion on finite rich $\wp$ words

R. Krishna Kumari *

Career Development Centre

College of Engineering and Technology

SRM Institute of Science and Technology

Kattankulathur 603203

Tamil Nadu

India

K. Janaki [†]

Department of Mathematics

Saveetha Engineering College

Saveetha Nagar, Thandalam

Chennai 602105

Tamil Nadu

India

R. Arulprakasam [§]

Department of Mathematics

College of Engineering and Technology

SRM Institute of Science and Technology

Kattankulathur 603203

Tamil Nadu

India

Abstract

A unified study of inversions on $\wp$ words resulted from the importance of words and inversions in molecular biology. In this study, we generalize the idea of inversion by using a technique called block inversion on finite rich $\wp$ words. A comparison of block inversion on finite total words and on finite $\wp$ words is made. We conclude that the total number of $\wp$ words in the block inversion set of a finite rich $\wp$ word of length n is strictly less than 2^{n-1} .

Subject Classification: 68Q45, 68R15.

Keywords: Conjugates, $\wp$ words, Palindromes, Primitivity.

* E-mail: krishrengan@gmail.com (Corresponding author)

[†] E-mail: janu89lava@gmail.com

[§] E-mail: r.aruljeeva@gmail.com (Corresponding author)

1. Introduction

Word combinations and formal language research are related since both fields investigate different aspects of words [10, 16]. Recent molecular genetic research have suggested that inversion (or reversal) operation is one of the vital operations on DNA sequences [12]. Inversion is a concept of rearrangement of a word. If a word x exists as the mirror image of another word y then x is the inversion of y . From the formal language theory point of view, many biological operations such as string matching and alignment problems consider hairpin inversion, pseudo-inversion and non-overlapping inversion [4, 11, 19]. The authors in [7] proved that context-free as well as regular languages are closed under the inversion but not closed under the iterated inversion. The decidability and closure properties of some language classes with respect to hairpin inversion was scrutinized in [5, 6]. In the literature of combinatorics of words, block inversion operation is used for many combinatorial manipulations [9, 17]. Block inversion of a word is a process of partitioning the word into blocks instead of letters and writing it in the reverse order. Block inversion can be part of algorithms for sorting or rearranging strings. For instance, it's used in algorithms like the Burrows-Wheeler Transform (BWT), which is a crucial step in data compression techniques like the Bzip2 compression algorithm. In computational biology, block inversion is used to measure the reversal distance between two permutations. This is relevant in evolutionary genomics when trying to understand the evolutionary relationships between species based on the rearrangements of genes or sequences.

In gene comparisons, partial words with gaps are taken into account [8, 14]. For instance, the building of two compatible *wp* words can be viewed as the alignment of two DNA sequences. DNA strands are used to encode information in DNA computing and are regarded as finite words. Some information components may be hidden or absent during encoding. Using ϕ words which stand for the locations of the missing symbols in a word, these components are made clear. The gaps in an incomplete word over an alphabet are not part of that alphabet but instead serve as a backup sign for an unidentified letter. The investigation of ϕ words was started in [1] and was expanded later by Blanchet Sadri [2, 3]. Here, we use a generalisation of the concept of inversion termed as block inversion on finite ϕ words. We study the significant differences between block inversion on finite total words and block inversion on finite ϕ words. We show that the total count of words in the block inversion set of a finite total

word of length n is less than or equal to 2^{n-1} but the total count of words in the block inversion set of a finite rich $\wp$ word of length n is strictly less than 2^{n-1} . In Section 2 the fundamental definitions pertaining to $\wp$ words and inversions are recalled. The block inversion operation on rich $\wp$ words is analyzed in Section 3 and finally in Section 5, conclusion and future work follows.

2. Preliminaries

In this section, we introduce standard notations and definitions related to $\wp$ words.

Let $\mathbb{A}$ denote a non-empty finite set of symbols, referred to as the “alphabet.” A consecutive arrangement of symbols from $\mathbb{A}$ is termed a “total word” or simply a “string.” The collection of all total words from $\mathbb{A}$ is represented as $\mathbb{A}^*$. We define $\mathbb{A}^+$ as $\mathbb{A}^*$ excluding the empty word, denoted as $\{\lambda\}$. A “language” L is a subset of $\mathbb{A}^*$. For a word $w \in \mathbb{A}^*$, its “length” is denoted as $|w|$. The set $Alph(w)$ represents the set of letters occurring in word w .

A “palindrome” is a word that reads the same forwards and backwards. Formally, if a word w satisfies $w = w^R$, where w^R is the “reversal” or “mirror image” of w , then w is considered a palindrome.

A “partial word” is a word that contains one or more holes or wildcard symbols, often represented by the symbol $\diamond$, which can appear anywhere within the word’s letter sequence. It’s important to note that the symbol $\diamond$ is not part of the alphabet $\mathbb{A}$ and serves as a placeholder for an unidentified letter. For example, a partial word u could be represented as $aa\diamond b$ with a length of $|u| = 4$.

Formally, a partial word u of length n over alphabet $\mathbb{A}$ is defined as a partial function $u: \{0, 1, 2, \dots, n-1\} \rightarrow \mathbb{A}$. In this definition, the function u assigns letters from the alphabet to specific positions in the word, and it is defined only for certain positions while leaving others (the holes) unspecified. The domain of u , denoted as $d(u)$, includes positions for which u is defined, while the set of holes, denoted as $h(u)$, includes positions where u is unspecified. A finite partial word p is considered “primitive” or “non-periodic” if there exists no finite partial word q such that p can be expressed as a power of q for all integers $m \geq 2$.

3. Inversion of a Finite $\wp$ word with Defect Zero

The inversion of a non-empty finite total word of length k , denoted as $u_1u_2\dots u_k$, where each u_m belongs to the set $\mathbb{A}_\diamond$ for all m , involves dividing the word u into k non-empty segments and arranging them in reverse order as $u_k\dots u_2u_1$. For example, the inversion of the total word aab results in baa . It's important to note that when dealing with finite rich $\wp$ words, the inversion operation differs from the inversion of finite words. This difference arises from the presence of $\diamond$ in a rich $\wp$ word, which cannot exist as a non-empty segment without a corresponding companion during the partitioning process. To address this distinction, we introduce the concept of block inversion operations on finite rich $\wp$ words. This approach deviates from the traditional definition of block inversion applied to finite total words. In our study, we explore the palindromic properties in conjunction with the block inversion operation on rich $\wp$ words. Additionally, it's worth noting that we consider the empty word, denoted as λ , as a valid palindrome.

Definition 3.1: Whenever a factor $p_\diamond$ of a $\wp$ word $u_\diamond$ over $\mathbb{A}_\diamond$ is compatible with its reversal (denoted by $p_\diamond \uparrow p_\diamond^R$), $p_\diamond$ is termed as a partial palindromic proper factor. The non-empty set of partial palindromic proper factors of $u_\diamond$ is denoted by $PPPF(u_\diamond)$.

Definition 3.2: Any $\wp$ word over $\mathbb{A}_\diamond$ with length n and having at least n distinct partial palindromic proper factors is a rich $\wp$ word.

Example 3.3: Assume a $\wp$ word $u_\diamond = ba\diamond bb$ over $\mathbb{A}_\diamond = \{a, b\} \cup \{\diamond\}$ with $|u_\diamond| = 5$.

The set of all distinct palindromic proper factors of $u_\diamond$ are

$$\{\lambda, a, b, bb, a\diamond, \diamond b, ba\diamond, \diamond bb, ba\diamond b\}.$$

In the midst of the above set, the set of all distinct partial palindromic factors of $u_\diamond$ are

$$\{a\diamond, \diamond b, ba\diamond, \diamond bb, ba\diamond b\}.$$

Here the count of well defined partial palindromic proper factors is equal to $|u_\diamond|$. Hence $u_\diamond$ is a rich partial word.

Definition 3.4: Consider a finite alphabet $\mathbb{A}_\diamond = \mathbb{A} \cup \{\diamond\}$. For any integer m , let $u_\diamond, I_m \in \mathbb{A}_\diamond^+$ where $u_\diamond$ represent the rich $\wp$ word over $\mathbb{A}_\diamond$; I_m denotes non-empty blocks of a rich $\wp$ word in $\mathbb{A}_\diamond^+$ such that

$$|I_m| \geq \begin{cases} 1 & \text{if } \diamond \notin I_m \\ 2 & \text{if } \diamond \in I_m. \end{cases}$$

The block inversion of $u_\diamond$ represented by $I_B(u_\diamond)$ is the set

$$I_B(u_\diamond) = \{I_p I_{p-1} \dots I_2 I_1 : u_\diamond = I_1 I_2 \dots I_{p-1} I_p, p \geq 1\}.$$

A rich $\wp$ word $u_\diamond$ can be partitioned at most of $|u_\diamond|$ blocks. The block inversion in $L_\diamond \subseteq \mathbb{A}_\diamond^+$ which is denoted by $I_B(L_\diamond)$ is

$$I_B(L_\diamond) = \bigcup_{u \in L_\diamond} I_B(u_\diamond).$$

Example 3.5: Assume $u_\diamond = ab\diamond b$ over $\mathbb{A}_\diamond = \{a, b\} \in \{\diamond\}$. Then

$$I_B(u_\diamond) = \{ab\diamond b, \diamond bba, bb\diamond a, b\diamond ba, bab\diamond, \diamond bab\}.$$

Block inversion on rich $\wp$ words can be used to explore and manipulate the structure of these $\wp$ words while preserving certain properties. The following algorithm takes a rich $\wp$ word, identifies the specified block, inverts it, and then reconstructs the rich word with the inverted block.

Algorithm 1 Block Inversion on finite rich $\wp$ words

Input : A rich word represented as a sequence of characters

Two indices (start_index, end_index) indicating the block to be inverted

Output: The rich word after performing block inversion

- 1 Initialize an empty result string as 'result' Initialize an empty string 'block' to store the block to be inverted
 - 2 **for** each character in the rich word **do**
 - 3 **if** start_index $\leq$ current index $\leq$ end_index **then**
 - 4 Append the current character to 'block'
 - 5 **else**
 - 6 Append the current character to 'result'
 - 7 Reverse the 'block' string to perform the block inversion
 - 8 Append the inverted 'block' to 'result'
 - 9 **return** 'result' as the rich word after block inversion
-

Theorem 3.6: [17] Assume a total word $x \in \mathbb{A}^+$ with $|x| = n$. Then $|I_B(x)| = 2^{n-1}$ iff $|\text{Alph}(x)| = n$.

Theorem 3.7: Assume a rich partial word $u_\diamond \in \mathbb{A}_\diamond^+$ with $|u_\diamond| = n$. Then $|I_B(u_\diamond)| < 2^{n-1}$ if and only if $|\text{Alph}(u_\diamond)| \leq n$.

Proof: Consider a rich partial word $u_\diamond \in \mathbb{A}_\diamond^+$ with $|u_\diamond| = n$, such that $|I_B(u_\diamond)| = 2^{n-1}$. If a is a letter in the alphabet, the number of occurrences of a in $u_\diamond$ is denoted as $|u_\diamond|_a$. Let $|u_\diamond|_a \geq 1$ and $|u_\diamond|_\diamond \geq 1$. Also, let $a \in \text{Alph}(u_\diamond)$ and $\diamond \in \text{Alph}(u_\diamond)$ with $\diamond \notin \mathbb{A}$. We consider four cases:

1. Assume that $|u_\diamond|_a > 1$ and $|u_\diamond|_\diamond > 1$. Then, for some $x, y, z \in \mathbb{A}^*$, $u_\diamond = axay\diamond z\diamond$. If we take $I_1 = axa$, $I_2 = y$, and $I_3 = \diamond z\diamond$, we have $I_3 I_2 I_1 = \diamond z\diamond y a x a \in I_B(u_\diamond)$. However, we cannot find $J_7 J_6 J_5 J_4 J_3 J_2 J_1 = \diamond z\diamond y a x a \in I_B(u_\diamond)$ for $J_1 = a$, $J_2 = x$, $J_3 = a$, $J_4 = y$, $J_5 = \diamond$, $J_6 = z$, and $J_7 = \diamond$, since $\diamond$ without a companion does not exist as a block. This contradicts $|I_B(u_\diamond)| = 2^{n-1}$.
2. Assume that $|u_\diamond|_a = 1$ and $|u_\diamond|_\diamond > 1$. Then, for some $x, y, z \in \mathbb{A}^*$, $u_\diamond = xay\diamond z\diamond$. Taking $I_1 = x$, $I_2 = a$, $I_3 = y$, and $I_4 = \diamond z\diamond$, we have $I_4 I_3 I_2 I_1 = \diamond z\diamond y a x \in I_B(u_\diamond)$. However, we cannot find $J_6 J_5 J_4 J_3 J_2 J_1 = \diamond z\diamond y a x \in I_B(u_\diamond)$ for $J_1 = x$, $J_2 = a$, $J_3 = y$, $J_4 = \diamond$, $J_5 = z$, and $J_6 = \diamond$. Thus, $I_4 I_3 I_2 I_1 \neq J_6 J_5 J_4 J_3 J_2 J_1$, implying $|I_B(u_\diamond)| < 2^{n-1}$.
3. Assume that $|u_\diamond|_a > 1$ and $|u_\diamond|_\diamond = 1$. Then, $u_\diamond = axay\diamond z$ for some $x, y, z \in \mathbb{A}^*$. Taking $I_1 = axa$, $I_2 = y$, and $I_3 = \diamond z$, we have $I_3 I_2 I_1 = \diamond z y a x a \in I_B(u_\diamond)$. However, we cannot find $J_7 J_6 J_5 J_4 J_3 J_2 J_1 = \diamond z y a x a \in I_B(u_\diamond)$ for $J_1 = a$, $J_2 = x$, $J_3 = a$, $J_4 = y$, $J_5 = z$, $J_6 = \diamond$, and $J_7 = \diamond$, as $\diamond$ without a companion does not exist as a block. This contradicts $|I_B(u_\diamond)| = 2^{n-1}$.
4. Assume that $|u_\diamond|_a = 1$ and $|u_\diamond|_\diamond = 1$. Then, for some $x, y, z \in \mathbb{A}^*$, $u_\diamond = xay\diamond z$. Taking $I_1 = x$, $I_2 = a$, $I_3 = y$, and $I_4 = \diamond z$, we have $I_4 I_3 I_2 I_1 = \diamond z y a x \in I_B(u_\diamond)$. However, we cannot find $J_5 J_4 J_3 J_2 J_1 = \diamond z y a x \in I_B(u_\diamond)$ for $J_1 = x$, $J_2 = a$, $J_3 = y$, $J_4 = z$, $J_5 = \diamond$, as $\diamond$ without a companion does not exist as a block. Thus, $I_4 I_3 I_2 I_1 \neq J_5 J_4 J_3 J_2 J_1$, implying $|I_B(u_\diamond)| < 2^{n-1}$.

For each $a \in \text{Alph}(u_\diamond)$ and $\diamond \in \text{Alph}(u_\diamond)$ with $\diamond \notin \mathbb{A}$, $|u_\diamond|_a \geq 1$ and $|u_\diamond|_\diamond \geq 1$. Thus, if $|I_B(u_\diamond)| < 2^{n-1}$, then $|\text{Alph}(u_\diamond)| \leq n$.

Conversely, consider $u_\diamond$ such that for each $a \in \text{Alph}(u_\diamond)$ and $\diamond \in \text{Alph}(u_\diamond)$ with $\diamond \notin \mathbb{A}$, $|u_\diamond|_a \geq 1$ and $|u_\diamond|_\diamond \geq 1$. For $1 \leq k < |u_\diamond|$, the rich partial word $u_\diamond$ can be split into k non-empty blocks in $\binom{|u_\diamond|-1}{k-1}$ well-defined ways. Since all the letters in $u_\diamond$ are well-defined, the block inversion of such a rich partial word contains $2^{|u_\diamond|-1}$ elements. Hence, the proof is complete.

A subset of the set $I_B(u_\diamond)$ that considers rich $\wp$ words with the inversion of minimum two non-empty blocks is termed as a proper block inversion.

Definition 3.8: The proper block inversion of $u_\diamond$ denoted by $PI_B(u_\diamond)$ is

$$PI_B(u_\diamond) = \{I_p I_{p-1} \dots I_2 I_1 : u_\diamond = I_1 I_2 \dots I_{p-1} I_p, p \geq 2\}.$$

The proper block inversion of $(u_\diamond)_\diamond$ denoted by $PI_B((u_\diamond)_\diamond)$ is

$$PI_B((u_\diamond)_\diamond) = \{I_B(PI_B(u_\diamond)) : u_\diamond = I_1 I_2 \dots I_{p-1} I_p, p \geq 2\}.$$

Example 3.9: Assume $u_\diamond = ac\Diamond ca$ over $\mathbb{A}_\diamond$. Then

$$\begin{aligned} PI_B(u_\diamond) &= \{c\Diamond caa, \Diamond caac, caac\Diamond, aac\Diamond c, cac\Diamond a, a\Diamond cca, a\Diamond cac, \\ &\quad acac\Diamond, ac\Diamond ca, acc\Diamond a, \Diamond caca\}, \\ PI_B((u_\diamond)_\diamond) &= \{c\Diamond caa, \Diamond caac, caac\Diamond, aac\Diamond c, cac\Diamond a, a\Diamond cca, a\Diamond cac, \\ &\quad acac\Diamond, ac\Diamond ca, acc\Diamond a, \Diamond caca, aacc\Diamond, aa\Diamond cc, acac\Diamond, \\ &\quad aca\Diamond c, ca\Diamond ca, c\Diamond aca, ac\Diamond ac, \Diamond ccaa\}. \end{aligned}$$

Definition 3.10: A rich $\wp$ word $u_\diamond$ over $\mathbb{A}_\diamond$ is called a block palindrome (denoted as b_{pal}) if $u_\diamond$ exists as

$$u_\diamond = I_{-p} I_{-(p-1)} \dots I_{-1} I_0 I_1 \dots I_{(p-1)} I_p,$$

where $I_m, I_{-m} \in \mathbb{A}_\diamond^+$ and $I_m = I_{-m}$ for $1 \leq m \leq p$. Here $I_0 \in \mathbb{A}^*$ is the mid-block and it may or may not be a palindrome.

Theorem 3.11: For any rich $\wp$ word $u_\diamond \in \mathbb{A}_\diamond^+$, $u_\diamond \in PI_B(u_\diamond)$ iff $u_\diamond$ is a b_{pal} .

Proof: Let the rich $\wp$ word $u_\diamond \in PI_B(u_\diamond)$. Then by Definition 3.10, there exists $I_m, I_{-m} \in \mathbb{A}_\diamond^+$ for $2 \leq m \leq p$ such that $u_\diamond = I_{-p} I_{-(p-1)} \dots I_{-1} = I_1 \dots I_{(p-1)} I_p$. Let $|I_1| \leq |I_p|$, then we have the following cases:

1. If $|I_1| = |I_p|$, it is trivial that $u_\diamond \in PI_B(u)$ is a b_{pal} .
2. Let $|I_1| < |I_p|$. Then $I_p = I_1x$ for some $x \in \mathbb{A}^*$. We have

$$\begin{aligned} u_\diamond &= I_p I_{p-1} \dots I_2 I_1 \\ &= I_1 x I_{p-1} \dots I_2 I_1 \\ &= I_1 I_0 I_1 \text{ where } I_0 = x I_{p-1} \dots I_2. \end{aligned}$$

Thus $u_\diamond$ is a b_{pal} .

Conversely, let $u_\diamond$ be a b_{pal} . Then $u_\diamond \in PI_B(u_\diamond)$ since $I_m = I_{-m}$ for $1 \leq m \leq p$.

Lemma 3.12: [3] *For any non-empty rich $\wp$ words $u_\diamond$ and $v_\diamond$, the equality $u_\diamond v_\diamond = v_\diamond u_\diamond$ holds iff $u_\diamond = w_\diamond^i$ and $v_\diamond = w_\diamond^j$ for some positive integers i, j and rich $\wp$ word $w_\diamond$.*

Remark 3.13: For all integers $i, j \geq 1$, the rich $\wp$ word $u_\diamond^i$ is a palindrome iff $u_\diamond^j$ is a palindrome.

The total count of total words in the block inversion set of a total word of length n is equal to 2^{n-1} . But it is not so, in the case of rich $\wp$ words. The following theorem proves that the total count of rich $\wp$ words in the block inversion set of a rich $\wp$ word of length n is strictly less than 2^{n-1} .

Lemma 3.14: [10] *The count of palindromic segments of an integer n is $2^{\lfloor \frac{n}{2} \rfloor}$. Also,*

$$\text{Count of palindromic segments of } n \text{ with } \begin{cases} 2k+1 \text{ parts equals } \binom{\lfloor \frac{n}{2} \rfloor - 1}{2} \\ 2k \text{ parts equals } \binom{\lfloor \frac{n}{2} \rfloor - 1}{k-1}. \end{cases}$$

Theorem 3.15: *Assume $u_\diamond$ to be a palindromic $\wp$ word with $|u_\diamond| = n$. Let $Pal(I_B(u_\diamond))$ denote the total number of palindromic words in $I_B(u_\diamond)$. Then*

$$Pal(I_B(u_\diamond)) = \begin{cases} < 2^{\frac{n}{2}-1} & \text{if } n \text{ is even, } |Alph(u_\diamond)| = \frac{n}{2} \\ < 2^{\frac{n-1}{2}-1} & \text{if } n \text{ is odd, } |Alph(u_\diamond)| = \frac{n+1}{2}. \end{cases}_9$$

Proof: Consider a palindromic rich $\wp$ word $u_\diamond$ with $|u_\diamond| = n$. The following two cases arises with respect to n .

Case 1: Let n be even and $|Alph(u_\diamond)| = \frac{n}{2}$. By Lemma 3.16, all elements of $I_B(u_\diamond)$ with palindromic blocks are palindromes. Let $v_\diamond \in I_B(u_\diamond)$ be a palindrome with palindromic blocks $n_1 + n_2 + \dots + n_m + \dots + n_2 + n_1$. Since n is even, n_m is also even. This implies that palindromic blocks with an even number of parts forms $v_\diamond$. Then followed by the Theorem 3.11 and the Lemma 3.14, we get the total number of palindromic elements in $I_B(u_\diamond)$ less than $2^{\frac{n}{2}-1}$.

Case 2: Let n be odd and $|Alph(u_\diamond)| = \frac{n+1}{2}$. This shows that a letter occurs once in the rich $\wp$ word and also in the mid position of $u_\diamond$. Let $v \in I_B(u_\diamond)$ be a palindrome. The palindromic blocks of $v_\diamond$ are of the form $n_1 + n_2 + \dots + n_m + n_m + \dots + n_2 + n_1$. Then by Theorem 3.11, the total number of palindromic elements in $I_B(u_\diamond)$ is less than $2^{\frac{n+1}{2}-1}$.

Remark 3.16: If $u_\diamond$ over $\mathbb{A}_\diamond^+$ is a palindromic rich $\wp$ word then all elements of $I_B(u_\diamond)$ with palindromic partitions are palindromes.

4. Block Inversion of Rich $\wp$ words in DNA Sequencing

Block inversion plays a crucial role in DNA sequence alignment. It helps identify regions of similarity and dissimilarity between DNA sequences. By comparing rich partial words with their block-inverted counterparts, researchers can align sequences to find matching subsequences, which is fundamental for studying genetic relationships and evolutionary history. Detecting mutations and variations in DNA sequences is vital for understanding genetic diseases and evolutionary processes. Block inversion enables the identification of differences between a given sequence and a reference sequence. Researchers can pinpoint mutations, insertions, deletions, or inversions within the DNA, contributing to mutation analysis and disease diagnostics. Genome assembly involves piecing together short DNA sequences into longer, contiguous sequences. Block inversion aids in this process by aligning and merging overlapping rich partial words. It helps reconstruct complete genomes, making it essential for studying an organism's genetic makeup. Algorithm 2 for block inversion of rich $\wp$ words is a versatile tool that can be applied in various aspects of DNA sequencing and analysis, contributing to our understanding of genetic information and evolution. Block Inversion of Rich $\wp$ Words in DNA Sequencing

Algorithm 2 Block Inversion of Rich $\wp$ Words in DNA Sequencing

```

1: procedure BLOCK INVERSION(rich  $\wp$ Word)
2:   inverted Word  $\leftarrow$  Empty string
3:   segment List  $\leftarrow$  List of segments in rich  $\wp$ Word
4:   for  $i$  from |segment List| down to 1 do
5:     current Segment  $\leftarrow$  segment List[ $i$ ]
6:     if current Segment is a regular letter then
7:       inverted Word  $\leftarrow$  inverted Word + current Segment else
8:         companion  $\leftarrow$  Find companion letter for current Segment
9:         inverted Word  $\leftarrow$  inverted Word + companion
10:  return inverted Word
11: end procedure

```

Structural variations in DNA, such as large insertions, deletions, or inversions, are crucial factors in understanding genetic diversity and disease susceptibility. Block inversion assists in detecting these structural variants by comparing rich partial words with their inverted forms. This contributes to characterizing genomic rearrangements. Many genomes contain repetitive DNA elements, which can be challenging to study. Block inversion allows researchers to analyze and identify repetitive sequences within DNA strands. Understanding repetitive regions is vital for investigating genomic stability and evolution. Phylogenetic analysis aims to establish evolutionary relationships between species. Block inversion contributes by aligning DNA sequences and identifying shared and unique regions. This information helps build phylogenetic trees, enabling researchers to trace the evolutionary history of organisms. In metagenomic studies, researchers analyze mixed DNA samples from diverse organisms. Block inversion assists in identifying common genetic elements among different species within a sample. This aids in characterizing the genetic diversity of microbial communities and ecosystems.

5. Conclusion and Future Work

We explored block inversion operations on finite rich $\wp$ words and also talked about the categorization of palindromic words in the block

inversion of a finite rich $\emptyset$ word. It would be intriguing in the future to examine the combinatorial characteristics of finite words with non-overlapped inversion and pseudo-inversion procedures.

References

- [1] J. Berstel and L. Boasson. Partial words and a Theorem of Fine and Wilf., *Theor. Comput. Sci.*, 218(1),135-141 (1999).
- [2] F. Blanchet-Sadri, Primitive Partial Words,*Discrete Applied Mathematics*, 148, 195-213 (2005).
- [3] F. Blanchet-Sadri.Algorithmic Combinatorics on Partial Words, *Discrete Mathematics and its Applications. CRC Press* (2008).
- [4] D.J. Cho, Y.-S. Han, S.D. Kang, H. Kim, S.K. Ko and K. Salomaa, Pseudo-Inversion on Formal Languages, *Proceedings of the Unconventional Computation and Natural Computation*, 93-104 (2014).
- [5] M.Daley, O.H.Ibarra and L.Kari, Closure and Decidability Properties of some Language Classes with respect to Ciliate Bio-operations, *Theor. Comput. Sci.*, 306 (1-3), 19-38 (2003).
- [6] M.Daley, L.Kari and I.McQuillan, Families of Languages defined by Ciliate Bio-operations, *Theor. Comput. Sci.*, 320(1), 51-69 (2004).
- [7] J.Dassow, V.Mitrana and A.Salomaa, Operations and Language generating Devices suggested by the Genome Evolution, *Theor. Comput. Sci.*, 270(1), 701-738 (2002).
- [8] M.J.Fischer and M.S.Paterson, String Matching and other Products, *In: Karp, R.M. (ed.) Complexity of Computation. SIAM-AMS Proceedings*, 113-125 (1974).
- [9] K. Goto, I. Tomohiro, H. Bannai and S. Inenaga, Block Palindromes: a New Generalization of Palindromes, *String Processing and Information Retrieval, Springer International Publishing, Cham*, 183-190 (2018).
- [10] S. Heubach and T. Mansour, Combinatorics of Compositions and Words, *Chapman; Hall/CRC*, (2009).
- [11] O.H. Ibarra, On Decidability and Closure Properties of Language Classes with respect to Bio-operations, *Proceedings of the 20th International Conference on DNA Computing and Molecular Programming*,148-160 (2014).
- [12] H. Kim and Y.-S. Han, Non-overlapping Inversion on Strings and Languages, *Theor. Comput. Sci.*, 592, 9-22 (2015).

- [13] R.Krishna Kumari, R. Arulprakasam, Meenakshi Paramasivan and V.R.Dare, Circular Partial Words and Arrays, *Proceedings of the International Symposium on Artificial Intelligence and Mathematics 2022 (ISAIM 2022), Fort Lauderdale, Florida, USA (2022)*.
- [14] R.Krishna Kumari, R. Arulprakasam and V.R.Dare, Combinatorial Properties of Fibonacci Partial Words and Arrays, *Journal of Discrete Mathematical Sciences and Cryptography*, 24(4), 1007-1020 (2021).
- [15] László Hegedüs and Benedek Nagy, Representations of Circular Words. *Proceedings 14th International Conference on Automata and Formal Languages, AFL 2014, Szeged, Hungary*, 151, 261-270 (2014).
- [16] M. Lothaire, *Combinatorics on Words*, Cambridge University Press (1983).
- [17] K.Mahalingam, A.Maity, P.Pandoh and R.Raghavan, Block Reversal on Finite Words, *Theoret. Comput.Sci.*, Preprint, (2021).
- [18] B. Rittaud, L. Vivier: Circular Words and Applications, *Proceedings of Words 2011, Electronic Proceedings in Theoret. Comput.Sci.*, 63, 31-36 (2011).
- [19] M. Schöniger, M. Waterman, A Local Algorithm for DNA Sequence Alignment with Inversions, *Bull. Math. Biol.*, 54, 521-536 (1992).

Received April, 2023