Measuring the clustering effect of BWT via RLE[☆]

Sabrina Mantaci[a], Antonio Restivo[a], Giovanna Rosone[b], Marinella Sciortino[a,*], Luca Versari[c]

[a] *University of Palermo, Dipartimento di Matematica e Informatica, ITALY.*
[b] *University of Pisa, Dipartimento di Informatica, ITALY.*
[c] *Scuola Normale Superiore, Pisa, ITALY.*

Abstract

The Burrows-Wheeler Transform (BWT) is a reversible transformation on which are based several text compressors and many other tools used in Bioinformatics and Computational Biology. The BWT is not actually a compressor, but a transformation that performs a context-dependent permutation of the letters of the input text that often create runs of equal letters (clusters) longer than the ones in the original text, usually referred to as the "clustering effect" of BWT. In particular, from a combinatorial point of view, great attention has been given to the case in which the BWT produces the fewest number of clusters (cf. [5, 16, 21, 23]). In this paper we are concerned about the cases when the clustering effect of the BWT is not achieved. For this purpose we introduce a complexity measure that counts the number of equal-letter runs of a word. This measure highlights that there exist many

 Final publication available at `https://doi.org/10.1016/j.tcs.2017.07.015`. Please, cite the publisher version: Sabrina Mantaci, Antonio Restivo, Giovanna Rosone, Marinella Sciortino, Luca Versari, Measuring the clustering effect of BWT via RLE, Theoretical Computer Science, 2017, DOI: `https://doi.org/10.1016/j.tcs.2017.07.015`. S. Mantaci, G. Rosone and M. Sciortino are partially supported by the project MIUR-SIR CMACBioSeq ("Combinatorial methods for analysis and compression of biological sequences") grant n. RBSI146R5L and by the Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS Project 2016).

*Corresponding Author.

Email addresses: `sabrina.mantaci@unipa.it` (Sabrina Mantaci), `antonio.restivo@unipa.it` (Antonio Restivo), `giovanna.rosone@unipi.it` (Giovanna Rosone), `marinella.sciortino@unipa.it` (Marinella Sciortino), `luca.versari@sns.it` (Luca Versari)

Postprint version; to appear on Theoretical Computer Science

words for which BWT gives an "un-clustering effect", that is BWT produce a great number of short clusters.

More in general we show that the application of BWT to any word at worst doubles the number of equal-letter runs. Moreover, we prove that this bound is tight by exhibiting some families of words where such upper bound is always reached. We also prove that for binary words the case in which the BWT produces the maximal number of clusters is related to the very well known Artin's conjecture on primitive roots. The study of some combinatorial properties underlying this transformation could be useful for improving indexing and compression strategies.

Keywords: BWT, permutation, run-length encoding

---

1. Introduction

The Burrows-Wheeler Transform (BWT) is a reversible transformation on the characters of a word on which are based several text compressors available today and also many tools used in different research fields, such as Bioinformatics and Computational Biology. It is known that the Burrows-Wheeler Transform performs a context-dependent permutation of all of the letters of the input data (cf. [4]). Since similar contexts usually adjoin similar sets of few letters, the permuted data has extensive groupings of similar letters and especially runs of equal letters (also called clusters).

In literature, several papers are related to the study of the compressibility of the BWT. Many of these studies use a natural statistical metric of the compressibility of a sequence, the empirical entropy[1].

Several papers (cf. [6, 11, 12, 17]) prove analytical upper bounds on the compression ratio of BWT-based compressors in terms of the k-th order empirical entropy H_k of the input word. Recall that, under the hypothesis of the Markovian nature of the input word w, $H_k(w)$ gives a lower bound on the compression ratio of any encoder that is allowed to use only the k-length context preceding letter in order to encode it. In [11, 12], the authors report some empirical results which seem to indicate that achieving good bounds with respect to H_k does not necessarily guarantee good compression results in practice.

[1]Empirical entropy states roughly that, given k consecutive letters, how much uncertainty there is on the average over the next letter in the sequence.

The empirical entropy, as has been observed in [15, 24], it does not reflect well the large-scale repetitiveness of the text[2], indeed when a text is repetitive, the letters preceding lexicographically adjacent suffixes are identical with high probability. Hence, the number of runs should be small when the text is repetitive. It has also been analyzed how various edit operations affect the number of runs when the input is a highly repetitive collection. Moreover, in [14] the relationship between the equal-letter runs in the Burrows-Wheeler transformed word and the k-th order entropy of the text is studied.

Actually, compression algorithms based on the BWT take advantage of the fact that the output of BWT shows a local similarity (occurrences of a given letter tend to occur in clusters) and so it turns out to be highly compressible. In literature, this property is referred to as the "clustering effect" of BWT. For instance, BWT applied to the word *mathematics* outputs the word *mmihttsecaa*, where one can see that equal letters are consecutive.

In order to investigate this "clustering effect" from a combinatorial viewpoint, it is interesting to consider which are the structural properties of the words for which the BWT produces the maximal or the minimal amount of clusters and whether the increase of the number of clusters can be maximal. Such a behavior may have some effects in terms of input compressibility. In fact, a perfect clustering produced by the BWT corresponds to optimal performances of compression techniques such as run-length encoding (RLE) and move-to-front (MTF)[2]. In past, several authors have considered the set $\mathcal{S}$ of the words v over a totally ordered alphabet $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$, with $a_1 < a_2 < \ldots < a_\sigma$, for which the words produced by BWT is $a_\sigma^{n_\sigma} a_{\sigma-1}^{n_{\sigma-1}} \cdots a_2^{n_2} a_1^{n_1}$ for some non-negative integers $n_1, n_2, \ldots, n_\sigma$. A complete description of the set $\mathcal{S}$ in the case of a binary alphabet has been given in [16], where it is proved that a word is in $\mathcal{S}$ if and only if it is a power of a conjugate of a standard sturmian word (cf. [13]). In the case of three letters alphabet a constructive characterization of the elements of $\mathcal{S}$ has been given by Simpson and Puglisi in [23]. In [21], the authors show that the elements of $\mathcal{S}$ are "rich" in palindromes, in the sense that they contain the maximum number of different palindromic factors. Finally, in [5] it is proved that perfectly clustering words are intrinsically related to k-discrete interval

[2]Informally, a text can be considered highly repetitive, if the number of equal letter runs in the Burrows-Wheeler Transform is much less than the length of the sequence.

exchange transformations. Such transformations can be intuitively defined by partitioning the interval $\{1, \ldots, n\}$ into k distinct sub-intervals. If each position of the interval is labeled by a letter of the alphabet Σ, the transformation produces a trajectory starting from a given position and is the infinite sequences of letters obtained by following the transformation. Formal definitions can be found in [5]. In [22], the authors propose an experimental study in order to analyze the clustering effect on "real" texts. In this study, the authors compare the rate of compression of a BWT-based compressor and of the LZ-based compressor [25] in relation to the clustering of the input.

In this paper we make a combinatorial analysis of the "clustering effect" of the BWT. In particular we use a measure that counts the number of equal-letter runs produced by the BWT. We are interested in exploring how the number of equal-letter runs (clusters) of the BWT output varies depending on the number equal-letter runs of the input. This measure highlights that there exist many words for which BWT gives an "un-clustering effect", that is BWT spreads equal letters far away from one another, producing many runs. For instance, for the infinite family of binary de Bruijn words, the BWT always determines an increase of the number of equal-letter runs. More in general we prove that the application of BWT to a word at most doubles the number of equal-letter runs with respect to the ones in the input word. Moreover we prove that this bound is tight by showing some families of words where such upper bound is always reached. We also prove that for binary words the case in which the BWT produces the maximal number of clusters is related to the very well known Artin's conjecture on primitive roots.

2. Preliminaries

Let $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$ be a finite ordered alphabet with $a_1 < a_2 < \ldots < a_\sigma$, where $<$ denotes the standard lexicographic order. We denote by Σ^* the set of words over Σ. Given a finite word $w = w_1 w_2 \cdots w_n \in \Sigma^*$ with each $w_i \in \Sigma$, the length of w, denoted $|w|$, is equal to n. We denote by $alph(w)$ the subset of Σ containing all the letters that appear in w. Given a finite word $w = w_1 w_2 \cdots w_n$ with each $w_i \in \Sigma$, a *factor* of a word w is written as $w[i, j] = w_i \cdots w_j$ with $1 \leq i \leq j \leq n$. A factor of type $w[1, j]$ is called a *prefix*, while a factor of type $w[i, n]$ is called a *suffix*. A *subsequence* of w is a word obtained by deleting some (not necessarily contiguous) letters from w. We also denote by $w[i]$ the i-th letter in w for any $1 \leq i \leq n$.

The *concatenation* of two words w and v, written wv, is simply the word consisting of the letters of w followed by the letters of v.

We say that two words $x, y \in \Sigma^*$ are *conjugate*, if $x = uv$ and $y = vu$, where $u, v \in \Sigma^*$. Conjugacy between words is an equivalence relation over Σ^*. The *conjugacy class* (w) of $w \in \Sigma^*$ is the set of all words $w_i w_{i+1} \cdots w_n w_1 \cdots w_{i-1}$, for any $1 \leq i \leq n$ and $w_i \in \Sigma$.

A nonempty word $w \in \Sigma^*$ is *primitive* if $w = u^h$ implies $w = u$ and $h = 1$. Recall that every nonempty word $u \in \Sigma^*$ can be written in a unique way as a power of a primitive word, i.e., there exists a unique primitive word w, called the *root* of u, and a unique integer k such that $u = w^k$.

A *Lyndon word* is a primitive word which is the minimum in its conjugacy class, with respect to the lexicographic order relation.

The *run-length encoding* of a word w, denoted by $\texttt{rle}(w)$, is a sequence of pairs (w_i, l_i) such that $w_i w_{i+1} \cdots w_{i+l_i-1}$ is a maximal run of a letter w_i (i.e., $w_i = w_{i+1} = \cdots = w_{i+l_i-1}$, $w_{i-1} \neq w_i$ and $w_{i+l_i} \neq w_i$), and all such maximal runs are listed in $\texttt{rle}(w)$ in the order they appear in w. We denote by $\rho(w) = |\texttt{rle}(w)|$ i.e., is the number of pairs in w, or equivalently the number of equal-letter runs in w. Moreover we denote by $\rho(w)_{a_i}$ the number of pairs (w_j, l_j) in $\texttt{rle}(w)$ where $w_j = a_i$.

Notice that $\rho(w) \leq \rho(w_1) + \rho(w_2) + \cdots + \rho(w_p)$, where $w_1 w_2 \cdots w_p = w$ is any partition of w.

The *Burrows-Wheeler Transform* (BWT) can be described as follows: given a word $w \in \Sigma^*$, the output of BWT is the pair $(\texttt{bwt}(w), I)$, where:

- $\texttt{bwt}(w)$ is the permutation of the letters in the input word w obtained by considering the matrix M (also called BWT *matrix*) containing the lexicographically sorted list of the conjugates of w, and by concatenating the letters of the last column L of matrix M.

- I is the position where the original word w appears in M.

Note also that the first column F of the BWT matrix M is the sequence of lexicographically sorted symbols of w.

The Burrows-Wheeler Transform is reversible by using the properties (cf. [4]) described in the following proposition.

Proposition 2.1. *Let (L, I) be a pair produced by the BWT applied to a word w. Let F be the sequence of the lexicographically sorted letters of $L = \texttt{bwt}(w)$. The following properties hold:*

1. *For all $i = 1, \ldots, n$, $i \neq I$, the letter $F[i]$ cyclically follows $L[i]$ in the original word w;*
2. *for each letter c, the r-th occurrence of c in F corresponds to the r-th occurrence of c in L;*
3. *The first letter of w is $F[I]$.*

From the above properties it follows that the BWT is reversible in the sense that, given L and I, it is possible to recover the original word w. These properties are the basis of a very well known indexing data structure called *FM-index* (cf. [7]).

Actually, according to Property 2 of Proposition 2.1, we can define a permutation $\mu\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$ where μ gives the correspondence between the positions of letters of F and L. The permutation μ is also called *FL-mapping*.

The permutation μ also represents the order in which we have to rearrange the elements of F to reconstruct the original word w. Hence, starting from the position I, we can recover the word w as follows:

$$w[i] = F[\mu^{i-1}(I)]\ , \text{ where } \mu^0(x) = x \text{ and } \mu^i(x) = \mu(\mu^{i-1}(x)), \text{with } 1 \leq i \leq n.$$

Remark 2.2. Note that given a pair (L, I) where $L \in \Sigma^*$ and $1 \leq I \leq n$, the word $w = \mathrm{BWT}^{-1}(L, I)$ (and then $\mathtt{bwt}(w) = L$) exists if and only if the permutation μ consists of a single cycle.

Remark 2.3. It follows by the definition that a Lyndon word is located at the first row of the correspondent BWT matrix, so in order to recover a Lyndon conjugate of a word w it is sufficient to apply $|w|$ times the FL-mapping by starting from $I = 1$. In this paper we are mainly interested to Lyndon words, so we can omit the index I.

3. How many equal-letter runs when the BWT is applied?

Several papers dealing with BWT focus on its "clustering effect", meaning that $\mathtt{bwt}(w)$ usually contains longer (and therefore less) equal-letters runs than the ones in w. This is the main reason for using BWT as a preprocessing for text compression.

On the other side one can find several examples where this clustering effect is not achieved at all, since BWT "un-clusters" the word as, for instance, in the following example.

Example 3.1. Consider the word $w = aacbbcccc$ where $\rho(w) = 4$. Then one can verify that $\texttt{bwt}(w) = cacbcaccb$ and $\rho(\texttt{bwt}(w)) = 8$.

The above mentioned "un-clustering effect" also happens for some important infinite classes, such as de Bruijn words, as we are going to prove in Section 5.

Here we are interested in exploring how the number of equal-letter runs of the BWT output varies depending on the number equal-letter runs of the input. It is clear that $\rho(\texttt{bwt}(w)) \geq |alph(w)|$.

To this concern, by considering that much literature studied the case when the maximum clustering effect is achieved, a natural question is to find the cases where the maximum un-clustering is achieved, formalized in the following:

Problem 3.2. *How much can $\rho(\texttt{bwt}(w))$ grow compared to $\rho(w)$?*

First, note that if v and w are conjugate words, then $|\rho(v) - \rho(w)| \leq 1$. On the other hand, for two conjugate words BWT produces the same word as output but with a different index. Since we are interested in the number of equal-letter runs of such a word, then we can assume that the input is a Lyndon word because it is one of the conjugates having least number of equal-letter runs.

The following theorem gives the answer to Problem 3.2 and the result can be somehow unexpected: the number of equal-letter runs can at most be doubled by the BWT.

Theorem 3.3. *Let $w \in \Sigma^*$ be a Lyndon word over a finite alphabet Σ. Then:*

$$\rho(\texttt{bwt}(w)) \leq 2\rho(w).$$

Proof. Let $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$ with $a_1 < a_2 < \cdots < a_\sigma$ and let $\texttt{rle}(w) = (b_1, l_1), (b_2, l_2), \ldots, (b_k, l_k)$, where $b_1, b_2, \ldots b_k \in \Sigma$.

When computing $\texttt{bwt}(w)$, one can split the BWT matrix into groups of rows according to their first letter a_i $(i = 1, 2, \ldots, \sigma)$. This splitting induces a parsing on $\texttt{bwt}(w)$. We denote by u_{a_i} the factor in $\texttt{bwt}(w)$ associated to the letter a_i, i.e., all the letters that in the original words precede an occurrence of the letter a_i. Then we can write $\texttt{bwt}(w) = u_{a_1} u_{a_2} \cdots u_{a_\sigma}$.

Consider any block u_{a_j}. In this block there are at most as many letters different from a_j as the number of different runs of a_j in w. In fact, in w, a_j is preceded by a letter different from a_j itself only in the beginning of each of

its runs. So the greatest possible number of runs contained in u_{a_j} is achieved when all the letters different from a_j never appear in the block one next to another, producing on u_{a_j} a number of runs $\texttt{rle}(u_{a_j})$ equal to $2 \cdot \rho(w)_{a_j}$. This happens for each block, then

$$\rho(\texttt{bwt}(w)) \leq \sum_{i=1}^{\sigma} \rho(u_{a_i}) \leq \sum_{i=1}^{\sigma} 2 \cdot \rho(w)_{a_i} = 2 \sum_{i=1}^{\sigma} \rho(w)_{a_i} = 2\ \rho(w)$$

□

Note that Example 3.1 shows that the upper bound given by Theorem 3.3 is tight.

In order to evaluate whether the application of the Burrows-Wheeler Transform increases the number of equal-letter runs of the input text we consider the measure defined as the ratio $\frac{\rho(\texttt{bwt}(w))}{\rho(w)}$.

The next goal of this paper is to look for a solution to the following problem:

Problem 3.4. *Find an infinite family of Lyndon words w such that*

$$\frac{\rho(\texttt{bwt}(w))}{\rho(w)} > 1.$$

Such a problem is faced in Section 5. Furthermore, since we proved in Theorem 3.3 that the ratio $\frac{\rho(\texttt{bwt}(w))}{\rho(w)}$ is at most 2, we are also interested in the following problem:

Problem 3.5. *Find an infinite family of Lyndon words w such that*

$$\frac{\rho(\texttt{bwt}(w))}{\rho(w)} = 2.$$

In the remaining part of this section we show some properties of this measure.

Proposition 3.6. *Let v be a Lyndon word (different from a single letter) and let k be a positive integer. Then*

$$\frac{\rho(\texttt{bwt}(v^k))}{\rho(v^k)} = \frac{1}{k} \frac{\rho(\texttt{bwt}(v))}{\rho(v)}.$$

Proof. Since v is a Lyndon word containing at least two different letters, it is straightforward that $\rho(v^k) = k\rho(v)$. Moreover, in [16] it has been proved that if $\texttt{bwt}(v) = v_1 v_2 \cdots v_n$ then $\texttt{bwt}(v^k) = v_1^k v_2^k \cdots v_n^k$. So, $\rho(\texttt{bwt}(v^k)) = \rho(\texttt{bwt}(v))$. Then the thesis follows. □

Corollary 3.7. *Let $w = v^k$, for some Lyndon word v and for some positive integer k. Then*

$$\frac{\rho(\texttt{bwt}(w))}{\rho(w)} = 2 \Rightarrow \frac{\rho(\texttt{bwt}(v))}{\rho(v)} = 2$$

Proof. From Proposition 3.6 we have that $\frac{\rho(\texttt{bwt}(w))}{\rho(w)} \leq \frac{\rho(\texttt{bwt}(v))}{\rho(v)}$. The thesis follows from the fact that, by Theorem 3.3, $\frac{\rho(\texttt{bwt}(v))}{\rho(v)}$ must be smaller than or equal to 2. □

Proposition 3.9 and Corollary 3.10 show that the Burrows-Wheeler Transform can lead to a maximum increase in the number of equal-letter runs even if the input word does not have a large number of runs, compared to its length.

Lemma 3.8. *Let $w = w_1 w_2 \cdots w_n \in \Sigma^*$ and let v be a subsequence of w, i.e., $v = w_{i_1} w_{i_2} \cdots w_{i_k}$ where $i_1 < i_2 < \ldots < i_k$. Then $\rho(v) \leq \rho(w)$.*

Proof. The thesis immediately follows from the fact that the presence of any letter between two consecutive letters of v or before the first letter or after the last letter of v, can not decrease the number of runs. □

We call *expansion* of order r of the letter $c \in \Sigma$ (denoted by $\theta_{r,c}$) the morphism that fixes all the letters except c, which is mapped to c^r, i.e., $\theta_{r,c}(b) = b$ if $b \neq c$, $\theta_{r,c}(c) = c^r$.

Proposition 3.9. *Let w be a Lyndon word over the alphabet Σ. Then, for each $c \in \Sigma$ and for each positive integer r,*

$$\frac{\rho(\texttt{bwt}(\theta_{r,c}(w)))}{\rho(\theta_{r,c}(w))} \geq \frac{\rho(\texttt{bwt}(w))}{\rho(w)}.$$

Proof. It is obvious that $\rho(\theta_{r,c}(w)) = \rho(w)$. Let us denote by w' and w'' two conjugates of w. One can verify that $\theta_{r,c}(w')$ and $\theta_{r,c}(w'')$ are conjugate of $\theta_{r,c}(w)$. Moreover, if w' is lexicographically smaller than w'', then $\theta_{r,c}(w')$ is lexicographically smaller than $\theta_{r,c}(w'')$. Consequently, $\texttt{bwt}(w)$ is a subsequence of $\texttt{bwt}(\theta_{r,c}(w))$. So, by using Lemma 3.8, $\rho(\texttt{bwt}(\theta_{r,c}(w))) \geq \rho(\texttt{bwt}(w))$. Then the thesis follows. □

Corollary 3.10. *Let w be a Lyndon word, over the alphabet Σ, such that*

$$\frac{\rho(\texttt{bwt}(w))}{\rho(w)} = 2.$$

Then, for each $c \in \Sigma$ and for each positive integer r,

$$\frac{\rho(\texttt{bwt}(\theta_{r,c}(w)))}{\rho(\theta_{r,c}(w))} = 2.$$

Note that Corollary 3.10 shows that the expansion of one or more letters could be a tool to construct infinite families of Lyndon words keeping the maximal increase of the number of equal-letter runs produced by BWT.

The following theorem shows that, for any alphabet, there exists an infinite family of Lyndon words where $\rho(\texttt{bwt}(v)) = 2\rho(v)$.

Theorem 3.11. *For any alphabet Σ and any $k \geq \max\{|\Sigma|, 3\}$, there is a Lyndon word v on Σ such that $\rho(v) = 2k-2$ and $\rho(\texttt{bwt}(v)) = 4k-4$.*

Proof. Let σ be the smallest letter in Σ and let $\delta_1, \ldots, \delta_{k-1}$ be a non-decreasing sequence of letters covering all the letters of the alphabet different from σ. Let $f_i = \sigma^{2i-1}\delta_i^{2i-1}$ for $i = 2, \ldots, k-1$, $f_1 = \sigma\delta_1\delta_1$ and $v = f_{k-1}\cdots f_1$. Clearly, $\rho(v) = 2k-2$. One can note that $\texttt{bwt}(v)$ can be factored in two parts: the first corresponding to all the conjugates starting with σ, and the second corresponding to all other conjugates.

For the first part, we first have the only conjugate that starts with $2k-3$ σ letters, then the one with $2k-4$, then the two conjugates that start with $2k-5$ σ letters (from the rightmost to the leftmost), and so on. From this we can easily see that the first part of $\texttt{bwt}(v)$ is $\delta_1\sigma\delta_{k-1}\sigma^3\cdots\delta_3\sigma^{2k-5}\delta_2\sigma^{k-2}$.

For the second part, let us assume first that all letters δ_i are distinct. In this case, we have exactly all the conjugates starting in the second part of each f_i, from the rightmost to the leftmost. This means that the second part of $\texttt{bwt}(v)$ is $\delta_1\sigma\delta_2^2\sigma\delta_3^4\sigma\cdots\delta_{k-1}^{2k-4}\sigma$.

It follows that $\texttt{bwt}(v) = \delta_1\sigma\delta_{k-1}\sigma^3\cdots\delta_3\sigma^{2k-5}\delta_2\sigma^{k-2}\delta_1\sigma\delta_2^2\sigma\delta_3^4\sigma\cdots\delta_{k-1}^{2k-4}\sigma$, that clearly has $4k-4$ runs. So $\rho(\texttt{bwt}(v)) = 4k-4$, that is the thesis.

On the other hand, when some letters δ_i are not distinct, we can still prove that any two occurrences of σ in the second part of $\texttt{bwt}(v)$ have a non-σ character between them. This still generates $2k-2$ runs in the second part. We know that any conjugate that ends with σ starts with $\delta_i^{e_i}$, where $e_i = 2$ if $i = 1$, and $e_i = 2i-1$ otherwise.

Also in this case the first conjugate starts with $\delta_1\sigma^{2k-3}$ and ends with δ_1. If there are conjugates between the first one and the conjugate starting with δ_1^2 and ending with σ, they must start with $\delta_1\sigma$ and end with δ_1. Let us now consider $1 \leq i \leq k-2$ and the conjugates ending with σ and starting with δ_i^{2i-1} and δ_{i+1}^{2i+1}, respectively. If $\delta_i \neq \delta_{i+1}$, all the conjugates between them end with δ_{i+1}. If $\delta_i = \delta_{i+1}$, all the conjugates that appear between them start with δ_i^x (for some x such that $2i-1 \leq x \leq 2i+1$). Moreover, the conjugate starting with δ_i^{2i} must appear between them, and clearly ends with δ_i. This proves that two occurrences of σ cannot be consecutive and are interspersed by the same symbol (that is different by σ). Finally, the last conjugate starts with δ_{k-1}^{2k-3} and ends with σ. This completes the proof. □

Example 3.12. Let $\Sigma = \{a, b\}$. The word $v = a^3b^3abb$ has $\texttt{bwt}(v) = bababbaba$, so $\rho(v) = 4$ and $\rho(\texttt{bwt}(v)) = 8$. The word $w = a^5b^5a^3b^3abb$ has $\texttt{bwt}(w) = babaaabaabbbabbbaba$, so $\rho(w) = 6$ and $\rho(\texttt{bwt}(w)) = 12$. The word $u = a^7e^7a^5d^5a^3c^3abb$ over the alphabet $\{a, b, c, d, e\}$ has $\texttt{bwt}(u) = baeaaadaaaaacaaabaccaddddaeeeeeea$, so $\rho(u) = 8$ and $\rho(\texttt{bwt}(u)) = 16$.

4. Maximal number of equal-letter runs by BWT in binary alphabets

In this section, we face with Problem 3.5 by considering the additional constraint that the Burrows-Wheeler Transform produces the maximal number of equal-letter runs, i.e., $\rho(\texttt{bwt}(w)) = |w|$. This could have an independent interest because it represents a dual point of view compared to the analysis of the case in which the BWT produces the fewest number of clusters. In fact, it is quite clear that, given any word $w \in \Sigma^*$, the value $\rho(\texttt{bwt}(w))$ is in the range $[|alph(w)|, |w|]$. In [16] there is a characterization of the binary words for which this clustering effect is maximal, i.e., BWT produces a word with just 2 equal-letter runs (one for each letter). Such words are the so-called *standard sturmian words*.

In [5, 21, 23] the authors show that the maximal clustering effect for k-letter alphabets can be achieved, since there exist families of words for which the BWT produces exactly k equal-letter runs.

We are interested in the following problem on the binary alphabet and we prove that it is related to the very well known Artin's conjecture on primitive roots.

Problem 4.1. *Characterize the family, denoted by $\mathcal{B}$, of Lyndon words $w \in \{a,b\}^*$ such that $\rho(\texttt{bwt}(w)) = |w|$, i.e., $\texttt{bwt}(w) = (ba)^{\frac{n}{2}}$ where $|w| = n$ and n is even.*

Note that one can verify that in case of a word w over a binary alphabet $\{a,b\}$, $\texttt{bwt}(w)$ must begin with the letter b and end with a. Therefore, if $|w|$ is odd then $\rho(\texttt{bwt}(w)) \leq |w| - 1$. In order to face with the Problem 4.1 we have to consider the FL-mapping from the words $a^{\frac{n}{2}}b^{\frac{n}{2}}$ to $(ba)^{\frac{n}{2}}$.

One can verify that in this case the FL-mapping $\mu : \{1,2,\ldots,n\} \mapsto \{1,2,\ldots,n\}$ is defined as $\mu(m) = 2m \pmod{n+1}$.

This means that Problem 4.1 of characterizing the words in $\mathcal{B}$ can be reduced to the following problem.

Problem 4.2. *Characterize the set, denoted by $\mathcal{P}$, of all the positive even integers n such that the permutation μ consists of a single cycle of length n.*

Note that the integer n belongs to the set $\mathcal{P}$ if and only $(ba)^{\frac{n}{2}}$ is a word produced by BWT and the word $w = \text{BWT}^{-1}((ba)^{\frac{n}{2}}, 1)$ belongs to the family $\mathcal{B}$.

Recall that given an integer $m > 1$, the congruence modulo m over $\mathbb{Z}$ is defined as follows: $\forall a, b \in \mathbb{Z}$, $a \equiv b \pmod{m}$ iff $a - b = km$ for some $k \in \mathbb{Z}$. We denote by $\mathbb{Z}_m$ the set of the congruence classes $[x]$ modulo m, $x = 0, 1, \ldots, m-1$.

The multiplicative group of integers modulo m, denoted by $\mathbb{Z}_m^*$, is the set of congruence classes $[x]$ $(x = 1, \ldots, m-1)$ such that x and m are coprime.

The following proposition gives the solution to Problem 4.2.

Proposition 4.3. *A positive integer n solves Problem 4.2 if and only if $n+1$ is an odd prime number and 2 generates the multiplicative group $\mathbb{Z}_{n+1}^*$.*

Proof. Since μ is defined as the permutation that sends m in $2m \pmod{n+1}$, we have that $\mu^i(1) \equiv 2^i \pmod{n+1}$. This implies that μ has a single cycle if and only if we can obtain any number from 1 to n as $\mu^i(1)$ for some i, i.e., if and only if $|\mathbb{Z}_{n+1}^*| = n$ and 2 is a generator of the multiplicative group. As $|\mathbb{Z}_{n+1}^*| = n$ if and only if $n+1$ is a prime number, the thesis follows. □

From previous proposition the set $\mathcal{P}$ of all numbers solving Problem 4.2 is the following:

$\mathcal{P} = \{2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, 100, 106, 130, 138, 148, 162, 172, 178,$

$180, 196, 210, 226, 268, 292, 316, 346, 348, 372, 378, 388, 418, 420, 442, 460, 466, 490,$

$508, 522, 540, 546, 556, 562, 586, 612, 618, 652, 658, 660, 676, 700, 708, 756, 772, \ldots\}$.

It is the sequence of integers n such that $n+1$ belongs to the integer sequence A001122 in [18] of primes with primitive root 2.

It is an open problem to establish whether $\mathcal{P}$ is an infinite set of integers. This problem is related to the very famous Artin's conjecture on primitive roots stating that a given integer c which is neither a perfect square nor -1 is a primitive root modulo infinitely many primes. The conjecture was formulated in 1927 and it is still open for each value of c. However, there exists a conditional proof for the conjecture proposed by Hooley in 1967 by assuming certain cases of the Generalized Riemann hypothesis (cf. [10]).

The following example shows two words such that their length belongs to $\mathcal{P}$.

Example 4.4. Let $\mathtt{bwt}(w) = (ba)^{26}$, then $\rho(\mathtt{bwt}(w)) = 52$ and

$$w = aaaaabaabbababaabaaaabbbaabbbbbabbaababababbabbbbaaabb$$

contains 26 equal-letter runs.

Let $\mathtt{bwt}(v) = (ba)^9$, then $\rho(\mathtt{bwt}(v)) = 18$ and

$$v = aaaabbababbbbaabab$$

contains 10 equal-letter runs.

The following two propositions show that the family $\mathcal{B}$ of Lyndon words, for which BWT outputs the maximal numbers of equal-letter runs, can be partitioned into two disjoint families $\mathcal{B}_1$ and $\mathcal{B}_2$, where $\mathcal{B}_1$ is the family of Lyndon words w such that $\rho(\mathtt{bwt}(w)) = 2\rho(w)$ and $\mathcal{B}_2$ is the family of Lyndon words v such that $\rho(\mathtt{bwt}(v)) = 2\rho(v)-2$. In Example 4.4, $w \in \mathcal{B}_1$ and $v \in \mathcal{B}_2$.

In particular, Proposition 4.5 describes a family of Lyndon words w such that $\frac{\rho(\mathtt{bwt}(w))}{\rho(w)} = 2$ and $\rho(\mathtt{bwt}(w)) = |w|$.

Proposition 4.5. *There exists a family $\mathcal{B}_1 \subset \mathcal{B}$ of binary Lyndon words such that for each word $w \in \mathcal{B}_1$, $\rho(\mathtt{bwt}(w)) = 2\rho(w) = |w|$.*

Proof. By Proposition 4.3 $w \in \mathcal{B}$ if and only if $n+1$ is an odd prime number with primitive root 2, where $n = |w|$. The set $\mathcal{P}$ can be partitioned into the sets $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_1$ is the set of odd primes of the form $4k+1$, for $k \geq 1$,

and $\mathcal{P}_2$ is the set of odd primes of the form $4k+3$, for $k \geq 0$. We consider the words w such that $n+1 \in \mathcal{P}_1$. We know that $\mathtt{bwt}(w) = (ba)^{\frac{n}{2}}$. Let h be such that $n = 2h$. Then $\mathtt{bwt}(w)$ contains exactly $2h$ equal-letter runs. We have to prove that if we recover the word w by using the FL-mapping μ we obtain h runs.

We know that, since $n+1 = 4k+1$, then $h = n/2 = 4k/2 = 2k$ is even. We can decompose $L = \mathtt{bwt}(w) = (ba)^{2k}$ in two parts, i.e., $L[1, 2k]$ and $L[2k+1, 4k]$. Such a decomposition induces a split of the column F of the BWT matrix into $F[1, 2k] = a^{2k}$ and $F[2k+1, 4k] = b^{2k}$. Recall that by Proposition 2.1 for each $i = 1, \ldots, 4k$ the letter $F[i]$ follows $L[i]$ in the word w. This means that all the letters in $L[1, 2k]$ are followed by a and all the letters in $L[2k+1, 4k]$ are followed by b in the word w. In particular, $L[1, 2k]$ contains k letters equal to b. This implies that w contains exactly k runs of b's. A similar argument can be used for $L[2k+1, 4k]$. In fact, $L[2k+1, 4k]$ contains k letters equal to a. Consequently, w contains exactly k runs of a's.

So, we have $2k$ equal-letter runs in v. □

We formulate the following

Conjecture 4.6. *The family $\mathcal{B}_1$ is infinite.*

Note that if the conjecture were true, $\mathcal{B}_1$ would solve Problem 3.5. Moreover, it is strictly related to a possible solution of Artin's conjecture.

The following proposition completes the evaluation of the increase of the equal-letter runs when the Burrows-Wheeler Transform determines the maximal value of ρ in the binary case.

Proposition 4.7. *There exists a family $\mathcal{B}_2 \subset \mathcal{B}$ of binary Lyndon words such that for each word $v \in \mathcal{B}_2$, $\rho(\mathtt{bwt}(v)) = 2\rho(v) - 2 = |v|$.*

Proof. By Proposition 4.3 $w \in \mathcal{B}$ if and only if $n+1$ is an odd prime number with primitive root 2, where $n = |w|$. Even in this case we can observe that the set $\mathcal{P}$ can be partitioned into the sets $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_1$ is the set of odd primes of the form $4k+1$, for $k \geq 1$, and $\mathcal{P}_2$ is the set of odd primes of the form $4k+3$, for $k \geq 0$. We consider the words v such that $n+1 \in \mathcal{P}_2$. We know that $\mathtt{bwt}(v) = (ba)^{\frac{n}{2}}$. Let h such that $n = 2h$. Then $\mathtt{bwt}(v)$ contains exactly $2h$ equal-letter runs. We have to prove that if we recover the word v by using the FL-mapping μ we obtain $h+1$ runs.

We know that $n+1 = 4k+3$. Then $h = n/2 = (4k+2)/2 = 2k+1$ is odd. We can decompose $L = \mathtt{bwt}(v) = (ba)^{2k+1}$ in two parts each of length

$2k+1$, i.e., $L[1, 2k+1]$ and $L[2k+2, 4k+2]$. Such a decomposition induces a split of the column F of the BWT matrix into $F[1, 2k+1] = a^{2k+1}$ and $F[2k+2, 4k+2] = b^{2k+1}$. By Proposition 2.1, all the letters in $L[1, 2k+1]$ are followed by a and all the letters in $L[2k+2, 4k+2]$ are followed by b in the word v. Note that, since L starts with the letter b, $L[1, 2k+1]$ contains $k+1$ b's and k a's. This implies that v contains exactly $k+1$ runs of b's. Analogously, $L[2k+2, 4k+2]$ starts with the letter a and, therefore, contains $k+1$ letters equal to a and k letters equal to b. Consequently, v contains exactly $k+1$ runs of a's.

So, this implies that v contains $2k+2$ equal-letter runs. □

The following proposition gives information about the combinatorial structure of the binary words w for which the Burrows-Wheeler Transform produces the maximal number of equal-letter runs, i.e., $\rho(\mathtt{bwt}(w)) = |w|$.

Given a binary word u, we denote by $\bar{u}$ the word obtained from u by exchanging a with b and, conversely, b with a.

Proposition 4.8. *The binary Lyndon word v of even length n such that $\mathtt{bwt}(v) = (ba)^{n/2}$ is of the form $u\bar{u}$.*

Proof. We have to prove that for each $i = 1, \ldots, n/2$, one has $v[i] = a$ and $v[n/2+i] = b$ or $v[i] = b$ and $v[n/2+i] = a$. Note that the elements in slots $1, \ldots, n/2$ coincide with the letter a and the elements in slots $n/2+1, \ldots, n$ coincide with the letter b in the column F of the BWT matrix.

Since 2 is a generator modulo $n+1$, we have that $2^{n/2} \not\equiv 1 \pmod{n+1}$. Moreover, since we know that $\left(2^{n/2}\right)^2 \equiv 1 \pmod{n+1}$, the only other possibility is that $2^{n/2} \equiv -1 \pmod{n+1}$. Thus, $\mu^{n/2}(1) \equiv -1 \pmod{n+1}$. So, it is easy to prove that $\mu^{n/2+i}(1) \equiv 2^{n/2+i} \equiv -2^i \equiv -\mu^i(1) \pmod{n+1}$ for each $i = 1, \ldots, n/2$.

Hence:

$$\mu^i(1) + \mu^{n/2+i}(1) \equiv 0 \pmod{n+1}.$$

As $1 \leq \mu^i(1) \leq n$, the only way this can hold is if

$$\mu^{n/2+i}(1) = n + 1 - \mu^i(1)$$

This implies that $\mu^{n/2+i}(1) > n/2$ if and only if $\mu^i(1) \leq n/2$, i.e., that $v[i]$ and $v[n/2+i]$ are different letters. □

Example 4.9. We consider the word $\mathtt{bwt}(w) = (ba)^{n/2}$ where $n = 28$, so that $\mathtt{bwt}(w) \in \mathcal{B}_1$, indeed 29 can be written as $4k + 1$, $k = 7$. Then $w = u\bar{u} = (aaaabaaabbabaa)(bbbbabbbaababb)$. Note that $\rho(w) = 14$.

We consider the word $\mathtt{bwt}(v) = (ba)^{n/2}$ where $n = 18$, so that $\mathtt{bwt}(v) \in \mathcal{B}_2$, indeed 19 can be written as $4k + 3$, $k = 4$. Note that $\rho(\mathtt{bwt}(v)) = 18$, so $v = u\bar{u} = (aaaabbaba)(bbbbaabab)$ and $\rho(v) = 10$.

5. Equal-letter runs in de Bruijn Words

In this section we consider the family of de Bruijn words. Such words have been considered in order to asymptotically compare Lempel-Ziv and Burrows-Wheeler based compression (cf. [20]). Here we prove that BWT always determines an increase of the number of equal-letter runs when applied to a binary de Bruijn word. We recall that a *de Bruijn word of span n* over a finite k-ary alphabet is a Lyndon word d_n of length k^n for which every word of length n appears exactly once as a factor in a conjugate of d_n. Moreover, for every n and for every k-ary alphabet, $(k!)^{k^{n-1}}/k^n$ de Bruijn words d_n exist (cf. [8, 19]).

In the following proposition we give a direct proof of the number of runs appearing in a binary de Bruijn word. Note that this result can be inferred by using some combinatorial properties analyzed in [8].

Proposition 5.1. *Let d_n be a de Bruijn word of span n over the binary alphabet $\{a, b\}$. Then $\rho(d_n) = 2^{n-1}$.*

Proof. We first consider the runs of a's. First of all there is no run a^i with $i > n$ otherwise a^n would be a word of length n that appears more than once in d_n. The run a^n is a particular word of length n, then, since d_n is a Lyndon word, it appears exactly once in d_n (in particular as factor of ba^nb in a conjugate of d_n).

The words ba^{n-1} and $a^{n-1}b$ also appear once, but since they are factors of ba^nb, we have no runs of a's of length $n - 1$.

For any $1 \leq i \leq n - 2$ consider the runs of the form a^i. They appear as factors of all the words of the form ba^ibw where w is any word of length $n - i - 2$. Each of the words ba^ibw appear exactly once. There are 2^{n-i-2} of such words, therefore there are 2^{n-i-2} runs a^i. We have overall:

$$1 + \sum_{i=1}^{n-2} 2^{n-i-2} = 1 + \sum_{i=0}^{n-3} 2^i = 1 + 2^{n-2} - 1 = 2^{n-2}$$

So there are 2^{n-2} runs of a's. For the same reason there are 2^{n-2} runs of b's, then overall $2 \cdot 2^{n-2} = 2^{n-1}$ runs. □

Theorem 5.2. *Let d_n a de Bruijn word of span n over a binary alphabet with $n \geq 3$. Then*

$$1 + \frac{1}{2^{n-2}} \leq \frac{\rho(\texttt{bwt}(d_n))}{\rho(d_n)} \leq 2 - \frac{1}{2^{n-2}}.$$

Proof. Recall that any binary de Bruijn word of span n has length 2^n, with $n \geq 2$. It was proved that numbers as 2^n with $n \geq 2$ do not belong to $\mathcal{P}$ (cf. [1, 3], so by Proposition 4.3 $\rho(\texttt{bwt}(d_n)) < |d_n| = 2^n$. It was proved in [9, 19] that $\texttt{bwt}(d_n) \in \Gamma^{2^{n-1}}$, where $\Gamma = \{ab, ba\}$. Moreover, one can note that ba must be a prefix and a suffix of $\texttt{bwt}(d_n)$. Since $\texttt{bwt}(d_n) \neq (ba)^{2^{n-1}}$ then aa and bb must be factors of $\texttt{bwt}(d_n)$. So, the upper bound follows because $\rho(\texttt{bwt}(d_n)) \leq 2^n - 2$. The lower bound on the number of equal-letter runs is reached when $\texttt{bwt}(d_n) = b(aabb)^{2^{n-2}-1}aba$. In this case this value is $2^{n-1} + 2$. Then, the thesis follows. □

6. Acknowledgments

The authors are very grateful to the anonymous referees for their helpful remarks and constructive comments.

References

[1] P. R. J. Asveld. Permuting operations on strings and their relation to prime numbers. *Discrete Applied Mathematics*, 159(17):1915–1932, 2011.

[2] J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei. A locally adaptive data compression scheme. *Commun. ACM*, 29(4):320–330, 1986.

[3] M. Bringer. Sur un problème de R. Queneau. *Math. Sci. Humaines No.*, 27:13–20, 1969.

[4] M. Burrows and D. J. Wheeler. A block sorting data compression algorithm. Technical report, DIGITAL System Research Center, 1994.

[5] S. Ferenczi and L. Q. Zamboni. Clustering Words and Interval Exchanges. *Journal of Integer Sequences*, 16(2):Article 13.2.1, 2013.

[6] P. Ferragina, R. Giancarlo, G. Manzini, and M. Sciortino. Boosting textual compression in optimal linear time. *J. ACM*, 52(4):688–713, 2005.

[7] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS 2000*, pages 390–398. IEEE Computer Society, 2000.

[8] H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM Review*, 24(2):195–221, 1982.

[9] P. M. Higgins. Burrows-Wheeler transformations and de Bruijn words. *Theor. Comput. Sci.*, 457:128–136, 2012.

[10] C. Hooley. On Artin's conjecture. *Journal für die reine und angewandte Mathematik*, 225:209–220, 1967.

[11] H. Kaplan, S. Landau, and E. Verbin. A simpler analysis of Burrows-Wheeler-based compression. *Theoret. Comput. Sci.*, 387(3):220–235, 2007.

[12] H. Kaplan and E. Verbin. Most Burrows-Wheeler based compressors are not optimal. In *CPM 2007*, volume 4580 of *LNCS*, pages 107–118. Springer, 2007.

[13] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.

[14] V. Mäkinen and G. Navarro. Compressed compact suffix arrays. In *CPM 2004*, volume 3109 of *LNCS*, pages 420–433. Springer, 2004.

[15] V. Mäkinen, G. Navarro, J. Sirén, and N. Välimäki. Storage and retrieval of highly repetitive sequence collections. *Journal of Computational Biology*, 17(3):281–308, 2010.

[16] S. Mantaci, A. Restivo, and M. Sciortino. Burrows-Wheeler transform and Sturmian words. *Information Processing Letters*, 86:241–246, 2003.

[17] G. Manzini. An analysis of the Burrows-Wheeler transform. *J. ACM*, 48(3):407–430, 2001.

[18] The On-Line Encyclopedia of Integer Sequences. Primes with primitive root 2. `https://oeis.org/A001122`.

[19] D. Perrin and A. Restivo. Words. In Miklos Bona, editor, *Handbook of Enumerative Combinatorics*. CRC Press, 2015.

[20] N. Prezza. Can Lempel-Ziv and Burrows-Wheeler compression be asymptotically compared? International Workshop on Combinatorial Algorithms - Problems Section, 2016.

[21] A. Restivo and G. Rosone. Burrows-Wheeler transform and palindromic richness. *Theoret. Comput. Sci.*, 410(30-32):3018 – 3026, 2009.

[22] A. Restivo and G. Rosone. Balancing and clustering of words in the Burrows-Wheeler transform. *Theoret. Comput. Sci.*, 412(27):3019 – 3032, 2011.

[23] J. Simpson and S. J. Puglisi. Words with simple Burrows-Wheeler transforms. *Electronic Journal of Combinatorics*, 15, article R83, 2008.

[24] J. Sirén. *Compressed Full-Text Indexes for Highly Repetitive Collections*. PhD thesis, 2012.

[25] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.