Theoretical Computer Science 1027 (2025) 115004

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the number of equal-letter runs of the bijective Burrows-Wheeler transform ☆

Elena Biagi [a], Davide Cenzato [b], Zsuzsanna Lipták [c,*], Giuseppe Romana [d]

[a] University of Helsinki, Department of Computer Science, Helsinki, Finland
[b] Ca' Foscari University of Venice, Department of Environmental Sciences, Informatics and Statistics, Venice, Italy
[c] University of Verona, Department of Computer Science, Verona, Italy
[d] University of Palermo, Department of Mathematics and Informatics, Palermo, Italy

ARTICLE INFO

MSC:
68P20
68R15
94A17

Keywords:
Combinatorics on words
Extended Burrows-Wheeler transform
Bijective Burrows-Wheeler transform
Lyndon factorization
Equal-letter run
Repetitiveness measure
Fibonacci words

ABSTRACT

The Bijective Burrows-Wheeler Transform (BBWT) is a variant of the famous BWT [Burrows and Wheeler, 1994]. The BBWT was introduced by Gil and Scott in 2012, and is based on the extended BWT of Mantaci et al. [TCS 2007] and on the Lyndon factorization of the input string. In the original paper, the compression achieved with the BBWT was shown to be competitive with that of the BWT, and it has been gaining interest in recent years. In this work, we present the first study of the number r_B of runs of the BBWT, which is a measure of its compression power. We exhibit an infinite family of strings on which r_B of the string and of its reverse differ by a multiplicative factor of $\Theta(\log n)$, where n is the length of the string. We also give several theoretical results on the BBWT, including a characterization of binary strings for which the BBWT has two runs. Finally, we present experimental results and statistics on $r_B(s)$ and $r_B(s^{\text{rev}})$, as well as on the number of Lyndon factors in the Lyndon factorization of s and s^{rev}.

1. Introduction

The Burrows-Wheeler-Transform (BWT) [10] is a fundamental invertible string transform, originally introduced as a preprocessing step for string compression. The BWT is often easier to compress than the original text and supports efficient pattern matching tasks while keeping the transform in compressed form. Due to this, the BWT has become the cornerstone of several string compressors and compressed data structures [17,27], as well as of some of the most commonly used bioinformatics tools (Bowtie [24,23], BWA [26, 33]).

The BWT's good compressibility is due to the so-called *clustering effect* [31]: if the input text is highly repetitive, then the final transform will tend to have long runs of the same character. The number of runs of the BWT, usually denoted r, has in recent years become increasingly popular as a measure of text repetitiveness. Its suitability as a repetitiveness measure, as well as its effectiveness in comparison with other text repetitiveness measures, was studied, for example, in [19,30,1,20]. In particular, in [19] it was shown that the parameter r is not invariant with respect to string reversal; the authors gave an infinite family of strings s for which $r(s)$ and $r(s^{\text{rev}})$ differ by a multiplicative factor of $\Theta(\log n)$, where s^{rev} is the reverse of s and n the length of the string. This result indicates that r has certain limitations as a measure of repetitiveness, as s and s^{rev} are clearly repetitive in the same way.

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.
E-mail address: zsuzsanna.liptak@univr.it (Zs. Lipták).

https://doi.org/10.1016/j.tcs.2024.115004
Received 5 April 2024; Received in revised form 22 October 2024; Accepted 22 November 2024

Available online 28 November 2024

The Bijective Burrows-Wheeler Transform (BBWT), introduced by Gil and Scott in 2012 [18], is another invertible transformation, which is a variation of the BWT. It is defined as the extended BWT (eBWT) [28] of the multiset of factors of the Lyndon factorization of the input string. As opposed to the BWT, the BBWT is bijective: no two distinct strings have the same BBWT, and every string is the BBWT of some string.

There has been increasing interest in the BBWT in the last decade. Bannai et al. [3] presented the first self-index based on the BBWT which supports efficient pattern-matching queries on the input text, similar to the original BWT. This comes with a small additional $\log(|P|)$ factor in the backward search algorithm, where P is the pattern. Köppl et al. [21] showed how to use in-place algorithms for constructing the BBWT and converting the BWT to the BBWT in quadratic time, highlighting a strong connection between these two transforms. Then Bannai et al. [4] gave the first linear time algorithm for constructing the Bijective BWT, thus making possible the efficient computation of this transform for large inputs. The BBWT has also inspired the definition of other bijective BWT variants [22,13,12].

In the original BBWT paper [18], the authors studied the suitability of this transform for text compression. Their experimental results show that on the Calgary corpus [5], the compression guaranteed by the BBWT is competitive. In particular, on average, the BBWT achieved about 1% better compression than the BWT. However, the effectiveness of the number r_B of runs of the BBWT as a repetitiveness measure has not been studied before.

In this paper, we present the first results on r_B, the number of runs of the BBWT, as a repetitiveness measure, comparing r_B of a string and of its reverse. We give an infinite family of strings for which r_B of the string and its reverse differ by a multiplicative factor of $\Theta(\log n)$, where n is the length of the string, thus proving a parallel result for the BBWT to that of Giuliani et al. [19] on the BWT. This result shows that the number of runs of the BBWT, as a measure of repetitiveness, exhibits the same defect as the BWT, namely that reversing the string may change it by up to a logarithmic factor. The family of strings used in this paper derive from finite Fibonacci words, as do those of [19], but the similarities end there; both the strings themselves and the proof techniques employed are different.

We also give new theoretical results on the BBWT, including a characterization of words for which the BWT and BBWT coincide, and of words on which the BBWT has two runs, paralleling a famous result by Mantaci et al. on the BWT [29]. Finally, we present experimental results on r_B, studying the multiplicative and additive difference between a string and its reverse, as well as the relationship with the number of factors of its Lyndon factorization.

Very recently, Badkobeh et al. [2] presented a follow-up study on compression with the BBWT. In particular, they showed a connection between r_B and two other repetitiveness measures: b, the size of a smallest bidirection macro-scheme [32], and z, the number of phrases of the LZ77-parsing [25]. They also showed that there are string families on which the compression power of the BBWT is worse than that of the BWT by a logarithmic factor.

Throughout the paper, we assume familiarity of the reader with the BWT and give only a very brief recap, while we introduce all necessary details on the extended BWT (eBWT) and the bijective BWT (BBWT).

This is the extended version of our previous conference paper [8]. Some of the results in this paper, in a preliminary version, were contained in the first author's master thesis [7].

1.1. Overview of paper

Section 2 contains the necessary background and notation, including the BWT and the extended BWT. In Section 3, we give new theoretical results on the BBWT. This is followed by the presentation of an infinite family of words for which the BBWT of the word and its reverse differ by a logarithmic factor (Section 4). Section 5 contains a summary and discussion of our experimental results; full results are contained in the Appendix. We close with some open problems in Section 6.

2. Basics

Let Σ be a finite ordered *alphabet* of size σ. A *string* (or *word*) over Σ is a finite sequence $s = s[1..n] = s[1]\cdots s[n]$ of *characters* from Σ. We denote the length of string s with $|s|$. The *alphabet of a string* s, $\mathrm{alph}(s) = \{s[i] \mid 1 \le i \le |s|\} \subseteq \Sigma$ is the set of characters that occur in the string. The *empty string* is the only string of length 0 and is denoted ϵ. For $n \ge 0$, Σ^n denotes the set of all strings of length n, and $\Sigma^* = \cup_{n\ge 0}\Sigma^n$ the set of all finite strings over Σ. Given string $s = s[1]s[2]\cdots s[n]$, its *reverse* is $s^{\mathrm{rev}} = s[n]\cdots s[2]s[1]$.

Let $s, u, x, v \in \Sigma^*$ such that $s = uxv$. Then u is called a *prefix*, x a *substring*, and v a *suffix* of s. A string t is a *subsequence* of s if t can be obtained from s by deleting some (possibly 0, possibly all) characters from s. A prefix (suffix, substring) u of a string s is called *proper* if $u \ne s$. For a string u and an integer $k \ge 1$, $u^k = u\cdot u\cdots u$ denotes the k-fold concatenation of u. A string s is called *primitive* if $s = u^k$ implies $u = s$ and $k = 1$. If s is not primitive then it is called a *power*. Every string s can be uniquely written as $s = u^k$ for a primitive string u, called *root of* s. Given a string u, we also define the *infinite* word $u^\omega = u\cdot u\cdot u\cdots$. We denote the number of maximal equal-letter runs of a string s by *runs*(s). For example, *runs*$(\mathtt{bbaaabc}) = 4$.

The *lexicographic order* on Σ^* is defined as follows: $u <_{\mathrm{lex}} v$ if either u is a proper prefix of v, or if there exist $x \in \Sigma^*$ and $b, c \in \Sigma, b < c$, such that xb is a prefix of u and xc is a prefix of v. Another order relation on Σ^* is the *omega-order* defined as follows: Let $s = u^k$ and $t = v^\ell$, with u and v primitive. Then $s <_\omega t$ if $u = v$ and $k < \ell$, or else if $u^\omega <_{\mathrm{lex}} v^\omega$. The lexicographic and the omega-order coincide on strings of the same length, but they can differ if one is a proper prefix of the other, e.g. $\mathtt{ab} <_{\mathrm{lex}} \mathtt{aba}$ but $\mathtt{aba} <_\omega \mathtt{ab}$. Note that throughout the paper we assume the usual order on the characters of our alphabets, i.e., $\mathtt{a} < \mathtt{b} < \mathtt{c} < \ldots$.

sorted rotations	BWT
abanan	n
anaban	n
ananab	b
banana	a
nabana	a
nanaba	a

omega-rotations	sorted rotations	BBWT
aaaa…	a	a
anan…	an	n
anan…	an	n
bbbb…	b	b
nana…	na	a
nana…	na	a

Fig. 1. BWT and BBWT of the string banana, with $\mathcal{L}yn(\texttt{banana}) = \{\texttt{b}, \texttt{an}, \texttt{an}, \texttt{a}\}$. The right side therefore also shows eBWT($\{\texttt{b}, \texttt{an}, \texttt{an}, \texttt{a}\}$).

Two strings s and s' are called *conjugates* (or *rotations*) if there exist u, v, possibly empty, s.t. $s = uv$ and $s' = vu$. Conjugacy is an equivalence relation. We denote the conjugacy class of a string $s \in \Sigma$ as $[s] = \{s' \mid s \text{ and } s' \text{ are conjugates}\}$. A string s is primitive if and only if its conjugacy class has cardinality $|s|$.

A primitive string is called a *Lyndon word* if it is lexicographically strictly smaller than all of its rotations. A *necklace* is a string which is smaller than or equal to all of its rotations; it is easy to see that a necklace is either a Lyndon word or a power of a Lyndon word. For a primitive string s, we denote by $L(s)$ the unique conjugate which is a Lyndon word (its *Lyndon rotation*). It is well known that every string s can be uniquely written as $s = f_1 \cdot f_2 \cdots f_m$ such that f_i are Lyndon words for $i = 1, \ldots, m$, and $f_1 \geq f_2 \geq \ldots \geq f_m$ [11], called s's *Lyndon factorization*. It follows from the definition that a string s is Lyndon if and only if its Lyndon factorization consists of one factor only. The multiset of Lyndon factors in the Lyndon factorization of s is denoted $\mathcal{L}yn(s)$. As an example, $L(\texttt{banana}) = \texttt{abanan}$, the Lyndon factorization of banana is $\texttt{b} \cdot \texttt{an} \cdot \texttt{an} \cdot \texttt{a}$, and $\mathcal{L}yn(\texttt{banana}) = \{\texttt{b}, \texttt{an}, \texttt{an}, \texttt{a}\}$.

The *Burrows-Wheeler Transform* (BWT) [10] of a string s is a permutation of the characters of s, whose ith character is the last character of the ith rotation of s, where the rotations are listed in lexicographic order. For example, BWT(banana) = nnbaaa, see Fig. 1 (left). The number of runs of the BWT is denoted $r(s) = runs(\text{BWT}(s))$, e.g. $r(\texttt{banana}) = 3$. It follows from the definition that two strings have the same BWT if and only if they are conjugates. It has been shown that BWT of a power u^m can be derived from the BWT of u, namely $\text{BWT}(u^m) = v[1]^m v[2]^m \cdots v[r]^m$, where $\text{BWT}(u) = v[1]v[2] \cdots v[r]$ [10,29].

The *extended BWT* (eBWT) [28] is a generalization of the BWT to a multiset of primitive strings $\mathcal{M}$: eBWT($\mathcal{M}$) is a permutation of the characters of the strings in $\mathcal{M}$, whose ith character is the last character of the ith rotation, where the rotations of all strings in $\mathcal{M}$ are listed w.r.t. omega-order. For example, eBWT({a,an,an,b}) = annbaa, see Fig. 1 (right). Note that for all strings s, $\text{eBWT}(\{s\}) = \text{BWT}(s)$. Next we list some known properties of the eBWT.

Lemma 1 *(Properties of the eBWT [28]). Let $\mathcal{S}$ be a multiset of primitive strings. The following hold:*

1. *For each $s \in \mathcal{S}$, BWT(s) is a subsequence of eBWT($\mathcal{S}$).*
2. *If s' is a conjugate of some $s \in \mathcal{S}$, then the number of runs of eBWT($\mathcal{S} \cup \{s'\}$) equals the number of runs of eBWT($\mathcal{S}$).*
3. *Let $m > 0$ be an integer and s a primitive string. If $\mathcal{S}$ consists of m copies of s, then eBWT($\mathcal{S}$) = BWT(s^m).*

3. The bijective BWT

Let $s = f_1 \cdot f_2 \cdots f_m$ be the Lyndon factorization of string s. The *bijective BWT (BBWT) of s* is defined as $\text{BBWT}(s) = \text{eBWT}(\mathcal{L}yn(s))$, where $\mathcal{L}yn(s) = \{f_1, f_2, \ldots, f_m\}$ is the multiset of Lyndon factors of s. As an example, BBWT(banana) = annbaa, since the Lyndon factorization of banana is $\texttt{b} \cdot \texttt{an} \cdot \texttt{an} \cdot \texttt{a}$, thus $\mathcal{L}yn(\texttt{banana}) = \{\texttt{a}, \texttt{an}, \texttt{an}, \texttt{b}\}$, and therefore $\text{BBWT}(\texttt{banana}) = \text{eBWT}(\{\texttt{a}, \texttt{an}, \texttt{an}, \texttt{b}\}) =$ annbaa, see Fig. 1 (right).

As we can see from this example, the two transforms BWT and BBWT do not in general coincide. In the next proposition, we characterize those strings for which they do:

Proposition 1. *$\text{BWT}(s) = \text{BBWT}(s)$ if and only if s is a necklace.*

Proof. Let $s = u^m$, where u is primitive and $m \geq 1$.

First assume that $\text{BWT}(s) = \text{BBWT}(s)$ holds, and let t be the conjugate of s which is a necklace. Then clearly, $t = v^m$ with $v = L(u)$, and $\mathcal{L}yn(t)$ consists of m copies of v. Thus:

$$\text{BBWT}(s) = \text{BWT}(s) \overset{s,t \text{ conj.}}{=} \text{BWT}(t) \overset{\text{Lemma 1 (3)}}{=} \text{eBWT}(\mathcal{L}yn(t)) \overset{\text{def.}}{=} \text{BBWT}(t).$$

By bijectivity of the BBWT, this implies that s equals its own necklace rotation t, and is thus a necklace.

Conversely, if s is a necklace, then u is a Lyndon word, and $\mathcal{L}yn(s)$ consists of m copies of u. Then $\text{BBWT}(s) = \text{eBWT}(\mathcal{L}yn(s)) = \text{BWT}(s)$, again by Lemma 1, part 3. □

Let us denote by $r_B(s) = runs(\text{BBWT}(s))$, the number of runs of the BBWT of s.

It is known that the number $r(s)$ of runs of the BWT of a binary string s is always even, except for the cases $s = \texttt{a}^n$ or $s = \texttt{b}^n$. This is because necessarily the first character of the BWT must be b, and the last character must be a. Since the BBWT is a bijective transform, the same property clearly does not hold. In fact, it is easy to see that BBWT(s) starts with an a (resp. ends with a b) if and only if $\texttt{a} \in \mathcal{L}yn(s)$ (resp. $\texttt{b} \in \mathcal{L}yn(s)$). In the next lemma, we give necessary and sufficient conditions that a or b appear as a Lyndon factor.

Lemma 2. *Let $s \in \{\mathtt{a},\mathtt{b}\}^*$ be a binary string. Then $\mathtt{a} \in \mathcal{L}yn(s)$ if and only if $\mathtt{a}$ is the last character of s. Symmetrically, $\mathtt{b} \in \mathcal{L}yn(s)$ if and only if $\mathtt{b}$ is the first character of s.*

Proof. Let $s = f_1 f_2 \cdots f_m$ be the Lyndon factorization of s. If $\mathtt{a} \in \mathcal{L}yn(s)$ then $\mathtt{a} = f_i$ for some i, and therefore, for all $j \geq i$, $f_j \leq \mathtt{a}$. Since $\mathtt{a}$ is the smallest Lyndon word, this implies that for all $j \geq i$, $f_j = \mathtt{a}$. In particular, the last Lyndon factor $f_m = \mathtt{a}$, thus the last character of s is $\mathtt{a}$. Conversely, if the last character of s is $\mathtt{a}$, then f_m ends with an $\mathtt{a}$. But it is easy to see that the only Lyndon word ending in $\mathtt{a}$ is $\mathtt{a}$, therefore, $f_m = \mathtt{a}$ and $\mathtt{a} \in \mathcal{L}yn(s)$.

The second equivalence can be proved analogously, noting that the only Lyndon word beginning with $\mathtt{b}$ is $\mathtt{b}$. □

We next characterize when $r_B(s)$ is odd.

Lemma 3. *Let $s \in \{\mathtt{a},\mathtt{b}\}^*$ be a binary string. It holds that $r_B(s)$ is odd if and only if s starts and ends with the same character.*

Proof. Clearly, r_B is odd if and only if BBWT(s) starts and ends with the same character. For every $f_i \in \mathcal{L}yn(s)$, f_i is the smallest lexicographically among all its rotations, and since all rotations of f_i have the same length, f_i is also smallest w.r.t. the ω-order. Therefore, the first entry in the eBWT-matrix is necessarily a Lyndon word. Recall that $\mathtt{a}$ is the only Lyndon word ending in $\mathtt{a}$, and notice that, with respect to the ω-order, $\mathtt{a}$ is smaller than all other non-empty strings. This implies that BBWT(s) starts with an $\mathtt{a}$ if and only if $\mathtt{a} \in \mathcal{L}yn(s)$. Similarly, the last character of BBWT(s) is the last character of the largest rotation of some Lyndon factor f_i. But the only primitive string whose largest rotation ends with a $\mathtt{b}$ is $\mathtt{b}$ itself. Therefore, the last character of BBWT(s) is $\mathtt{b}$ if and only if $\mathtt{b} \in \mathcal{L}yn(s)$.

Altogether we get that $r_B(s)$ is odd if and only if either $\mathtt{a}$ or $\mathtt{b}$ are in $\mathcal{L}yn(s)$, but not both. By Lemma 2, this is equivalent to s starting and ending with $\mathtt{a}$, or s starting and ending with $\mathtt{b}$. □

The following simple connection between $r_B(s)$ and $r_B(s^{\text{rev}})$ follows directly from Lemma 3.

Corollary 1. *For every binary string s, the difference between the number of runs of the BBWT of s and the BBWT of its reverse is even.*

The BWT of a string achieves maximal compression if it has as many runs as the number of characters that appear in the string, i.e., if $r(s) = |\text{alph}(s)|$. In case of binary alphabets, it was shown in [29] that this perfect clustering effect of the BWT is a characterization of a specific family of finite words, called *standard words*. *Standard Sturmian words*, or simply *standard words*, are strings that can be defined via a *directive sequence*, an infinite sequence of integers $\{d_i\}_{i\geq 0}$ such that $d_0 \geq 0$ and $d_i > 0$ for all $i > 0$. Standard words are the strings $\{s_i\}_{i\geq 0}$ generated by this sequence, as follows: $s_0 = \mathtt{b}$, $s_1 = \mathtt{a}$, and $s_{i+1} = s_i^{d_{i-1}} s_{i-1}$ for $i \geq 1$. For instance, the directive sequence $1, 1, 1, 1, \ldots$ generates the well-known sequence of *finite Fibonacci words*: $s_0 = \mathtt{b}, s_1 = \mathtt{a}, s_2 = \mathtt{ab}, s_3 = \mathtt{aba}, s_4 = \mathtt{abaab}, s_5 = \mathtt{abaababa}, s_6 = \mathtt{abaababaabaab}, s_7 = \mathtt{abaababaabaababaababa}, \ldots$

Theorem 1 *([29]). BWT(s) $= \mathtt{b}^k\mathtt{a}^\ell$, for some $k, \ell \geq 1$, if and only if s is the power of a rotation of a standard word. In particular, $s = u^m$, for u rotation of a standard word v and $m = \gcd(k, \ell)$.*

Theorem 1 implies that standard words are primitive, since in that case $m = 1$, and therefore, the number of $\mathtt{b}$'s and the number of $\mathtt{a}$'s are co-prime, thus, s cannot be a power. We now give a characterization for the BBWT, similar to Theorem 1. As opposed to the BWT, here the transform does not necessarily begin with $\mathtt{b}$ and end with $\mathtt{a}$.

Theorem 2. *The number of runs of the BBWT of a binary string s is two if and only if (i) s has the form $\mathtt{b}^k\mathtt{a}^\ell$ for some $k, \ell \geq 1$, or (ii) there exist a standard word v and an integer $m \geq 1$ such that $s = L(v)^m$.*

To prove Theorem 2, we first need a lemma. This lemma says that if a string w is the BWT of some string t, then it is easy to know its pre-image under the BBWT: it is the necklace rotation of t.

Lemma 4. *Let $w, t \in \{\mathtt{a},\mathtt{b}\}^*$ be such that $w = BWT(t)$. Then $w = BBWT(L(u)^m)$, where $t = u^m$ and u is primitive.*

Proof. Let $t = u^m$, u primitive. Then BWT(t) $= z[1]^m z[2]^m \cdots z[r]^m$, where $z[1] \cdots z[r] =$ BWT(u). Set $s = v^m$, where $v = L(u)$. Then the Lyndon factorization of s is $v \cdot v \cdots v$, and therefore, $\mathcal{L}yn(s)$ consists of m copies of v, thus, by Lemma 1, BBWT(s) $=$ eBWT($\mathcal{L}yn(s)$) $=$ eBWT($\{v, v, \ldots, v\}$) $=$ BWT(v^m) $=$ BWT($L(u)^m$) $=$ BWT(s) $=$ BWT(t), as claimed. □

Proof of Theorem 2. We will show that

1. BBWT(s) $= \mathtt{a}^\ell\mathtt{b}^k$ if and only if $s = \mathtt{b}^k\mathtt{a}^\ell$, and
2. BBWT(s) $= \mathtt{b}^k\mathtt{a}^\ell$ if and only if $s = L(v)^m$, where v is a standard word and $m = \gcd(k, \ell)$.

To see (1), note that $\mathrm{BBWT}(s) = \mathtt{a}^{\ell}\mathtt{b}^{k}$ implies that $\mathcal{L}yn(s)$ consists of ℓ copies of $\mathtt{a}$ and k copies of $\mathtt{b}$, which in turn implies $s = \mathtt{b}^{k}\mathtt{a}^{\ell}$. Conversely, the Lyndon factorization of $\mathtt{b}^{k}\mathtt{a}^{\ell}$ is $\mathtt{b}\cdot\mathtt{b}\cdots\mathtt{b}\cdot\mathtt{a}\cdot\mathtt{a}\cdots\mathtt{a}$, and thus, $\mathrm{BBWT}(\mathtt{b}^{k}\mathtt{a}^{\ell}) = \mathtt{a}^{\ell}\mathtt{b}^{k}$.

To prove (2), let first $\mathrm{BBWT}(s) = \mathtt{b}^{k}\mathtt{a}^{\ell}$. By Theorem 1, $\mathtt{b}^{k}\mathtt{a}^{\ell} = \mathrm{BWT}(s)$, where $s = u^{m}$ and u is a rotation of a standard word v. Therefore, by Lemma 4, $\mathtt{b}^{k}\mathtt{a}^{\ell} = \mathrm{BBWT}(L(u)^{m}) = \mathrm{BBWT}(L(v)^{m})$, since $L(u) = L(v)$. As v is a standard word, the left-to-right implication is proven.

Conversely, let $s = L(v)^{m}$ for some standard word v. Then the Lyndon factorization of s is $s = L(v)\cdot L(v)\cdots L(v)$, and thus $\mathrm{BBWT}(s) = \mathrm{eBWT}(\mathcal{L}yn(s)) = \mathrm{eBWT}(\{L(v), L(v), \ldots, L(v)\}) = \mathrm{BWT}(s)$ by Lemma 1. By Theorem 1, $\mathrm{BWT}(s) = \mathtt{b}^{k}\mathtt{a}^{\ell}$ with $k = m\cdot k'$ and $\ell = m\cdot\ell'$ and $\mathrm{BWT}(v) = \mathrm{BWT}(L(v)) = \mathtt{b}^{k'}\mathtt{a}^{\ell'}$. This completes the proof. □

In the rest of the paper, we will look at the relationship between $r_B(s)$ and $r_B(s^{\mathrm{rev}})$. To this end, we define two functions, the *runs-ratio* $\rho_B(s)$ and the *runs-difference* $\delta_B(s)$ of string s as:

$$\rho_B(s) = \max\left(\frac{r_B(s)}{r_B(s^{\mathrm{rev}})}, \frac{r_B(s^{\mathrm{rev}})}{r_B(s)}\right) \qquad \text{(runs-ratio)},$$

$$\delta_B(s) = r_B(s) - r_B(s^{\mathrm{rev}}) \qquad \text{(runs-difference)}.$$

In the following section, we will study an infinite family of strings for which ρ_B is logarithmic in the length of the string. Then, in Section 5, we will give experimental results both on ρ_B and on δ_B, as well as on the number of Lyndon factors of the string and of its reverse.

4. Fibonacci words

In this section, we will show that the Lyndon rotations of Fibonacci words have the property that their runs-ratio is $\Theta(\log n)$, where n is the length of the word.

We start with an outline of the proof contained in this section. Let us denote the kth Fibonacci word by s_k, its Lyndon rotation by w_k, and the reverse of w_k by t_k (precise definitions to follow). We will prove that $r_B(w_k) = 2$ and $r_B(t_k) = 2(k-2)$. To do this, we first need some properties of Fibonacci words (Lemma 5 and Corollary 2). Then we compute the exact form of the Lyndon factorization of t_k (Proposition 2). From this we derive that the Lyndon factors of t_k are exactly the Lyndon rotations of the Fibonacci words s_i, for $0 \le i \le k-2$ (Corollary 2). Next, in Proposition 3, we show that the number of runs of the eBWT of the set $\mathcal{S}_k$ is $2k$, where $\mathcal{S}_k$ is the set of Fibonacci words of orders up to k. We do this by computing the exact form of $\mathrm{eBWT}(\mathcal{S}_k)$. These results will be used to show that $r_B(t_k) = 2(k-2)$. On the other hand, it follows from results in Section 3 that $r_B(w_k) = 2$. We put it all together in Theorem 3.

Recall the definition of Fibonacci words: $s_0 = \mathtt{b}$, $s_1 = \mathtt{a}$ and $s_{k+1} = s_k s_{k-1}$ for $k \ge 1$. It follows directly from the definition that for all $k \ge 0$, $|s_k| = F_k$, where $\{F_k\}_{k\ge 0}$ is the well-known Fibonacci sequence $1, 1, 2, 3, 5, 8, 13, \ldots$, defined by $F_0 = F_1 = 1$, $F_{k+1} = F_k + F_{k-1}$ for $k \ge 1$. Fibonacci words have been studied extensively, see [14] for an overview. Some of the properties of Fibonacci words also follow from properties that have been shown for standard words in general, see e.g. [16,15,6,9,29]. Next we list some of these properties:

Lemma 5 *(Some known properties of Fibonacci words [14,15,6,29]). Let s_k be the Fibonacci word of order $k \ge 0$. There exists a palindrome x_k, known as* central word*, with the following properties:*

1. *for all $k \ge 2$: if k is even, then $s_k = x_k\mathtt{ab}$, and if k is odd, then $s_k = x_k\mathtt{ba}$ (in particular, $x_2 = \epsilon$) [14].*
2. *for all $k \ge 4$:*
   - *if k is even, then $s_k = x_{k-1}\mathtt{ba}x_{k-2}\mathtt{ab} = x_{k-2}\mathtt{ab}x_{k-1}\mathtt{ab}$, and*
   - *if k is odd, then $s_k = x_{k-1}\mathtt{ab}x_{k-2}\mathtt{ba} = x_{k-2}\mathtt{ba}x_{k-1}\mathtt{ba}$ [15].*
3. *for all k: $(s_k)^{rev}$ is a conjugate of s_k [15].*
4. *for all $k \ge 2$: $\mathtt{a}x_k\mathtt{b}$ is a Lyndon word [6].*
5. *for $k \ge 2$: $BWT(s_k)$ has two runs; in particular, $BWT(s_k) = \mathtt{b}^{F_{k-2}}\mathtt{a}^{F_{k-1}}$ [29].*
6. *for $k \ge 2$: $x_k\mathtt{ab}$ and $x_k\mathtt{ba}$, called* standard rotations*, are adjacent in the BW-matrix of s_k [29].*

From these we can easily deduce further properties. Recall that $L(s)$ is the Lyndon rotation of the primitive string s.

Corollary 2. *Let x_k be the palindrome from Lemma 5, i.e. $s_k = x_k\mathtt{ab}$ for k even, and $s_k = x_k\mathtt{ba}$ for k odd. Then*

1. $L(s_k) = \mathtt{a}x_k\mathtt{b}$,
2. $x_k = x_{k-1}\mathtt{ba}x_{k-2} = x_{k-2}\mathtt{ab}x_{k-1}$, *if k even,*
3. $x_k = x_{k-1}\mathtt{ab}x_{k-2} = x_{k-2}\mathtt{ba}x_{k-1}$, *if k odd.*

Since Lyndon rotations of Fibonacci words will be of central importance, we list the first few here: $L(s_0) = \mathtt{b}$, $L(s_1) = \mathtt{a}$, $L(s_2) = \mathtt{ab}$, $L(s_3) = \mathtt{aab}$, $L(s_4) = \mathtt{aabab}$, $L(s_5) = \mathtt{aabaabab}$, $L(s_6) = \mathtt{aabaababaabab}$.

Next we study the strings t_k and their Lyndon factorization, where t_k is the reverse of the Lyndon rotation of the Fibonacci word s_k. We will show that the Lyndon factorization of t_k consists of the Lyndon rotations of s_i for $i = 0, \ldots, k-2$, where the words of

even order are listed increasingly (w.r.t. their order i), followed by those of odd order listed decreasingly, followed by one additional factor $L(s_1) = \mathtt{a}$. Formally:

Proposition 2. *Let $t_k = L(s_k)^{rev}$ be the reverse of the Lyndon rotation of the Fibonacci word of order k. The Lyndon factorization of t_k is as follows:*

1. $t_k = L(s_0) \cdot L(s_2) \cdot L(s_4) \cdots L(s_{k-2}) \cdot L(s_{k-3}) \cdot L(s_{k-5}) \cdots L(s_3) \cdot L(s_1) \cdot L(s_1)$ *if k is even, and*
2. $t_k = L(s_0) \cdot L(s_2) \cdot L(s_4) \cdots L(s_{k-3}) \cdot L(s_{k-2}) \cdot L(s_{k-4}) \cdots L(s_3) \cdot L(s_1) \cdot L(s_1)$ *if k is odd.*

Example 1. In the following we list t_k with its Lyndon factorization, for $k = 2, \dots, 8$:

- $t_2 = L(s_2)^{\text{rev}} = \mathtt{b} \cdot \mathtt{a}$,
- $t_3 = L(s_3)^{\text{rev}} = \mathtt{b} \cdot \mathtt{a} \cdot \mathtt{a}$,
- $t_4 = L(s_4)^{\text{rev}} = \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{a} \cdot \mathtt{a}$,
- $t_5 = L(s_5)^{\text{rev}} = \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{aab} \cdot \mathtt{a} \cdot \mathtt{a}$,
- $t_6 = L(s_6)^{\text{rev}} = \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{aabab} \cdot \mathtt{aab} \cdot \mathtt{a} \cdot \mathtt{a}$,
- $t_7 = L(s_7)^{\text{rev}} = \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{aabab} \cdot \mathtt{aabaabab} \cdot \mathtt{aab} \cdot \mathtt{a} \cdot \mathtt{a}$,
- $t_8 = L(s_8)^{\text{rev}} = \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{aabab} \cdot \mathtt{aabaababaabab} \cdot \mathtt{aabaabab} \cdot \mathtt{aab} \cdot \mathtt{a} \cdot \mathtt{a}$.

For the proof of Proposition 2, we will first need two lemmas. The first one gives a simple recursion for t_k:

Lemma 6. *Let $t_k = L(s_k)^{rev}$. The following recursion holds for $k \geq 2$:*

- $t_k = t_{k-2}t_{k-1}$ *if k is even, and*
- $t_k = t_{k-1}t_{k-2}$ *if k is odd.*

Proof. First note that since $L(s_k) = \mathtt{a}x_k\mathtt{b}$, and x_k is a palindrome, therefore $t_k = \mathtt{b}x_k\mathtt{a}$. Let $k \geq 2$ be even. By Corollary 2, $t_k = \mathtt{b}x_k\mathtt{a} = \mathtt{b}x_{k-2}\mathtt{ab}x_{k-1}\mathtt{a} = t_{k-2}t_{k-1}$. Similarly, if k is odd, then $t_k = \mathtt{b}x_k\mathtt{a} = \mathtt{b}x_{k-1}\mathtt{ab}x_{k-2}\mathtt{a} = t_{k-1}t_{k-2}$. □

The second lemma gives a factorization of the Lyndon rotations of the s_k.

Lemma 7. *Let $k \geq 4$. Then the following holds:*

- $L(s_k) = L(s_{k-3})L(s_{k-5}) \cdots L(s_3)L(s_1)L(s_1)L(s_0)L(s_2)L(s_4) \cdots L(s_{k-2})$, *if k is even, and*
- $L(s_k) = L(s_{k-2})L(s_{k-4}) \cdots L(s_3)L(s_1)L(s_1)L(s_0)L(s_2)L(s_4) \cdots L(s_{k-3})$, *if k is odd.*

Proof. The proof is by induction on k. For the induction base, we have $L(s_4) = \mathtt{aabab} = \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{b} \cdot \mathtt{ab}$, and $L(s_5) = \mathtt{aabaabab} = \mathtt{aab} \cdot \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{b} \cdot \mathtt{ab}$. Now let $k > 5$. If k is even, then

$$\begin{aligned} L(s_k) &= \mathtt{a}x_k\mathtt{b} = \mathtt{a}x_{k-1}\mathtt{ba}x_{k-2}\mathtt{b} = L(s_{k-1})L(s_{k-2}) \\ &= \underbrace{L(s_{k-3}) \cdots L(s_3)L(s_1)L(s_1)L(s_0)L(s_2) \cdots L(s_{k-4})}_{L(s_{k-1}),\ \text{by the I.H.},\ k-1\ \text{odd}} \cdot L(s_{k-2}). \end{aligned}$$

Similarly, if k is odd, then

$$\begin{aligned} L(s_k) &= \mathtt{a}x_k\mathtt{b} = \mathtt{a}x_{k-2}\mathtt{ba}x_{k-1}\mathtt{b} = L(s_{k-2})L(s_{k-1}) \\ &= L(s_{k-2}) \underbrace{L(s_{k-4}) \cdots L(s_3)L(s_1)L(s_1)L(s_0)L(s_2) \cdots L(s_{k-3})}_{L(s_{k-1}),\ \text{by the I.H.},\ k-1\ \text{even}}. \end{aligned}$$ □

Now we are ready to prove Proposition 2.

Proof of Proposition 2. The proof is by induction on k. For the base cases, $t_2 = \mathtt{b} \cdot \mathtt{a}$, and $t_3 = \mathtt{b} \cdot \mathtt{a} \cdot \mathtt{a}$, as claimed. Now let $k > 3$. If k is even, then $t_k = t_{k-2}t_{k-1} = \mathtt{b}x_{k-2}\mathtt{ab}x_{k-1}\mathtt{a}$. Thus,

$$\begin{aligned} t_k &= \underbrace{L(s_0)L(s_2) \cdots L(s_{k-4})L(s_{k-5}) \cdots L(s_3)L(s_1)L(s_1)}_{t_{k-2}\ \text{by the I.H.}} \cdot \underbrace{L(s_0)L(s_2) \cdots L(s_{k-4})L(s_{k-3}) \cdots L(s_3)L(s_1)L(s_1)}_{t_{k-1}\ \text{by the I.H.}} \\ &= L(s_0)L(s_2) \cdots L(s_{k-4}) \cdot \underbrace{L(s_{k-5}) \cdots L(s_3)L(s_1)L(s_1)L(s_0)L(s_2) \cdots L(s_{k-4})}_{L(s_{k-2})\ \text{by Lemma 7}} \cdot L(s_{k-3}) \cdots L(s_3)L(s_1)L(s_1) \end{aligned}$$

$$= L(s_0) \cdot L(s_2) \cdot L(s_4) \cdots L(s_{k-2}) \cdot L(s_{k-3}) \cdot L(s_{k-5}) \cdots L(s_3) \cdot L(s_1) \cdot L(s_1).$$

The claim for k odd follows analogously. □

Example 2. Let us consider the reverse of the Fibonacci word of order $k = 8$.

$$\begin{aligned} t_8 &= t_6 \cdot t_7 \\ &= \texttt{babaababaabaa} \cdot \texttt{babaababaabaababaabaa} \\ &= \texttt{b} \cdot \texttt{ab} \cdot \texttt{aabab} \cdot \underline{\texttt{aabaababaabab}} \cdot \texttt{aabaabab} \cdot \texttt{aab} \cdot \texttt{a} \cdot \texttt{a} \\ &= L(s_0) \cdot L(s_2) \cdot L(s_4) \cdot \underline{L(s_6)} \cdot L(s_5) \cdot L(s_3) \cdot L(s_1) \cdot L(s_1). \end{aligned}$$

Note that $\mathcal{L}yn(t_8) = \mathcal{L}yn(t_7) \cup \{L(s_6)\}$, where $L(s_6)$ is the underlined factor in the Lyndon factorization of t_8. The factorization from Lemma 7 is $L(s_6) = \texttt{aabaababaabab} = \texttt{aab} \cdot \texttt{a} \cdot \texttt{a} \cdot \texttt{b} \cdot \texttt{ab} \cdot \texttt{aabab}$.

Corollary 3 *(from Proposition 2). Let $t_k = L(s_k)^{rev}$ be the reverse of the Lyndon rotation of the Fibonacci word of order k. Then*

$$\mathcal{L}yn(t_k) = \{L(s_0), L(s_1), L(s_2), \ldots, L(s_{k-2})\} \cup \{L(s_1)\},$$

i.e. the factor $L(s_1) = \texttt{a}$ appears with multiplicity 2 in the factorization, while all other factors appear exactly once.

Standard rotations of Fibonacci words will play a crucial role in the following. These are the words $u_k = x_k\texttt{ab}$ and $v_k = x_k\texttt{ba}$ (see Lemma 5). We list these words up to order 7 in Table 1.

Table 1
Standard rotations of Fibonacci words of order k.

k	2	3	4	5	6	7
u_k	ab	aab	abaab	abaabaab	abaababaabaab	abaababaabaababaabaab
v_k	ba	aba	ababa	abaababa	abaababaababa	abaababaabaababaababa

In the following lemma, we describe the omega-order between standard rotations of consecutive orders.

Lemma 8. *Let u_k and v_k be the standard rotations of the kth order Fibonacci word, for some $k > 2$. The following hold:*

1. *$u_k >_\omega v_{k-1}$ if k is even;*
2. *$v_k <_\omega u_{k-1}$ if k is odd.*

Proof. As before, we denote with s_k the kth Fibonacci word. By Lemma 5, $s_k = u_k$ if k is even, otherwise $s_k = v_k$. We will first prove that $u_k^\omega >_{\text{lex}} v_{k-1}^\omega$ for all even $k > 2$.

Recall that, by Corollary 2, we have $x_k = x_{k-1}\texttt{ba}x_{k-2} = x_{k-2}\texttt{ab}x_{k-1}$, if k is even, and $x_k = x_{k-1}\texttt{ab}x_{k-2} = x_{k-2}\texttt{ba}x_{k-1}$, if k is odd. We can therefore factorize the prefix of u_k^ω as follows:

$$\begin{aligned} u_k^\omega &= (x_k\texttt{ab})^\omega \\ &= x_k\texttt{ab}x_k\texttt{ab}\cdots \\ &= x_{k-1}\texttt{ba}x_{k-2}\texttt{ab}x_{k-1}\texttt{ba}x_{k-2}\texttt{ab}\cdots \\ &= x_{k-1}\texttt{ba}x_{k-2}\texttt{ab}x_{k-3}\texttt{ba}x_{k-2}\underline{\texttt{ba}}x_{k-2}\texttt{ab}\cdots, \end{aligned}$$

where we underlined the first character that will differ from v_{k-1}^ω, whose prefix we can factorize similarly as follows:

$$\begin{aligned} v_{k-1}^\omega &= (x_{k-1}\texttt{ba})^\omega \\ &= x_{k-1}\texttt{ba}x_{k-1}\texttt{ba}x_{k-1}\texttt{ba}\cdots \\ &= x_{k-1}\texttt{ba}x_{k-2}\texttt{ab}x_{k-3}\texttt{ba}x_{k-2}\underline{\texttt{a}}\texttt{b}x_{k-3}\texttt{ba}\cdots \end{aligned}$$

Observe that u_k^ω and v_{k-1}^ω share the common prefix $x_{k-1}\texttt{ba} \cdot x_{k-2}\texttt{ab}x_{k-3}\texttt{ba} \cdot x_{k-2}$, immediately followed by b and a in u_k^ω and v_{k-1}^ω respectively (the underlined letters in the last line of both equations), therefore, $u_k >_\omega v_{k-1}$.

The claim for the case k odd can be shown analogously:

$$\begin{aligned} v_k^\omega &= (x_k\texttt{ba})^\omega \\ &= x_k\texttt{ba}x_k\texttt{ba}\cdots \end{aligned}$$

	sorted rotations	eBWT
	a	a
u_3	aab	b
v_3	aba	a
u_2	ab	b
	baa	a
v_2	ba	a
	b	b

	sorted rotations	eBWT
	a	a
u_3	aab	b
	aabab	b
v_3	aba	a
u_4	abaab	b
v_4	ababa	a
u_2	ab	b
	baa	a
	baaba	a
	babaa	a
v_2	ba	a
	b	b

	sorted rotations	eBWT
	a	a
u_3	aab	b
	aabaabab	b
	aababaab	b
	aabab	b
v_3	aba	a
u_5	abaabaab	b
v_5	abaababa	a
u_4	abaab	b
	ababaaba	a
v_4	ababa	a
u_2	ab	b
	baa	a
	baabaaba	a
	baababaa	a
	baaba	a
	babaabaa	a
	babaa	a
v_2	ba	a
	b	b

Fig. 2. Tables showing the eBWT matrix of S_k for $k = \{3, 4, 5\}$. The standard rotations are marked on the left of each table. S_3 : eBWT({b, a, ab, aba}) = ababaab, S_4 : eBWT({b, a, ab, aba, abaab}) = abbababaaaab, S_5 : eBWT({b, a, ab, aba, abaab, abaababa}) = abbbbababaabaaaaaaab.

$$
\begin{aligned}
&= x_{k-1}\mathtt{ab}x_{k-2}\mathtt{ba}x_{k-1}\mathtt{ab}x_{k-2}\mathtt{ba}\cdots \\
&= x_{k-1}\mathtt{ab}x_{k-2}\mathtt{ba}x_{k-3}\mathtt{ab}x_{k-2}\underline{\mathtt{a}}\mathtt{b}x_{k-2}\mathtt{ba}\cdots \\
&<_{\text{lex}} x_{k-1}\mathtt{ab}x_{k-2}\mathtt{ba}x_{k-3}\mathtt{ab}x_{k-2}\underline{\mathtt{b}}\mathtt{a}x_{k-3}\mathtt{ab}\cdots \\
&= x_{k-1}\mathtt{ab}x_{k-1}\mathtt{ab}x_{k-1}\mathtt{ab}\cdots \\
&= (x_{k-1}\mathtt{ab})^{\omega} \\
&= u_{k-1}^{\omega}. \quad \square
\end{aligned}
$$

The final piece we need for the proof of the main theorem of this section will be Proposition 3, which gives the number of runs of the eBWT of the set of Fibonacci words $S_k = \{s_0, s_1, \ldots, s_k\}$.

Proposition 3. *Let $k > 0$ and S_k be the set of Fibonacci words of order up to k, i.e. $S_k = \{s_0, s_1, \ldots, s_k\}$. Then eBWT($S_k$) has $2k$ runs.*

Proof. We prove the claim by induction on k. For $k = 1, 2$, eBWT({a, b}) = ab, which has 2 runs, and eBWT({a, b, ab}) = abab, which has 4 runs, as claimed.

Now let $k > 2$. We will observe what happens to eBWT(S_{k-1}) when we insert s_k into S_{k-1} and will show that exactly 2 new runs are created, see Fig. 2.

Let eBWT(S_{k-1}) be given, and consider what happens when we add s_k. As before, denote the two standard rotations of order k as $u_k = x_k\mathtt{ab}$ and $v_k = x_k\mathtt{ba}$. Recall that $s_k = u_k$ if k is even, and $s_k = v_k$ if k is odd. Clearly, $u_k <_\omega v_k$ for all k. Moreover, by Lemma 8, the following relationships between standard rotations of subsequent order hold: $u_k >_\omega v_{k-1}$ if k is even, and $v_k <_\omega u_{k-1}$ if k is odd.

Now, when considering eBWT(S_{k-1}), the two strings u_k and v_k are inserted together, i.e. they are adjacent in the eBWT-matrix of S_k. This is because they have the common prefix x_k, of length $|s_k| - 2$, which, for $k > 3$, is longer than the longest string previously present (of length $|s_{k-1}| = F_{k-1} < F_k - 2$); for $k = 3$ the claim can be seen by direct inspection (see Fig. 2, left).

Moreover, the two standard rotations will be inserted immediately adjacent to one of the two standard rotations of order $k - 1$. This is because one of the two standard rotations is always equal to s_k (u_k for k even, v_k for k odd), and thus, u_{k-1} is a proper prefix of v_k or v_{k-1} is a proper prefix of u_k, and no longer prefix of these strings can be present. Thus we have $s_k = u_k >_\omega v_{k-1} = s_{k-1}$ if k is even, and $s_k = v_k <_\omega u_{k-1} = s_{k-1}$ if k is odd. Therefore, u_k and v_k will be inserted immediately after $s_{k-1} = v_{k-1}$ if k is even, and immediately before $s_{k-1} = u_{k-1}$ if k is odd. It follows by induction that s_{k-1} was inserted immediately after (immediately before) s_{k-2} if k is even (if k is odd). This in turn implies that the string just before (just after) u_k and v_k is s_{k-2}, for k even (for k odd).

Now consider the number of runs of eBWT(S_{k-1}). By the induction hypothesis, eBWT(S_{k-1}) has $2k - 2$ runs. Inserting the two standard rotations creates two new runs. This is because they end in b and a, in this order, and they are inserted between the two previous Fibonacci words: if k is even, then they are inserted between s_{k-1}, which ends in a, and s_{k-2}, which ends in b; if k is odd, then between s_{k-2}, which ends in a, and s_{k-1}, which ends in b.

Note that, since s_k is a standard word, BWT(s_k) has the form $\mathtt{b}^k\mathtt{a}^\ell$, and by Lemma 1, this will be a subsequence of eBWT(S_k). This implies that, if the two standard rotations words of order k are inserted between, say, position i and position $i + 1$ of eBWT(S_{k-1}),

Table 2
All forward strings s of length $n = 3, 4, 5$ and $\sigma = 2$.

n	forward strings	no. forward strings
3	aab, abb	2
4	aaab, aaba, aabb, abab, abbb, babb	6
5	aaaab, aaaba, aaabb, aabab, aabba, aabbb, abaab, ababb, abbab, abbbb, baabb, babbb	12

Table 3
All forward strings s of length $n = 3, 4$ and $\sigma = 3$.

n	forward strings	no. forward strings
3	aab, aac, abb, abc, acb, acc, bac, bbc, bcc	9
4	aaab, aaac, aaba, aabb, aabc, aaca, aacb, aacc, abab, abac, abbb, abbc, abca, abcb, abcc, acab, acac, acbb, acbc, accb, accc, baac, babb, babc, bacb, bacc, bbac, bbbc, bbcb, bbcc, bcac, bcbc, bccc, cabc, cacc, cbcc	36

then all rotations of s_k ending in b will be inserted before i, and all rotations ending in a will be inserted after $i + 1$. Inserting a b will create a new run only if it is inserted between two a's; likewise, inserting an a will create a new run only if it is inserted between two b's. It is not difficult to prove (by induction) that all a-runs before the position of s_{k-1} have length 1, and that all b-runs after the position of s_{k-1} have length 1.

Thus, exactly two new runs are created. This completes the proof. □

Theorem 3. *Let $w_k = L(s_k)$ be the Lyndon rotation of the kth Fibonacci word s_k. Then $\rho_B(w_k) = \Theta(\log |w_k|)$.*

Proof. First note that since w_k is a Lyndon word, $\mathrm{BBWT}(w_k) = \mathrm{BWT}(w_k)$ by Proposition 1. Since it is a rotation of s_k, $\mathrm{BWT}(w_k) = \mathrm{BWT}(s_k)$. By Lemma 5, $\mathrm{BWT}(s_k)$ has two runs, so $\mathrm{BBWT}(w_k)$ has two runs, thus, $r_B(w_k) = 2$.

Now let $t_k = (w_k)^{\mathrm{rev}}$. By Corollary 3, the set of Lyndon factors of t_k is $\mathcal{S}_{k-2} = \{s_0, \ldots, s_{k-2}\}$. It follows from Lemma 1 that the number of runs of a multiset depends only on the set of the elements (and not on the multiplicity of each element). By Proposition 3, the number of runs of $\mathrm{eBWT}(\mathcal{S}_{k-2})$ is $2(k-2)$, and therefore, $r_B(t_k) = 2(k-2)$.

Finally, using the fact that $|w_k| = |s_k| = F_k$ grows exponentially in k, it follows that $\rho_B(w_k) = \frac{2(k-2)}{2} = \Theta(k) = \Theta(\log |w_k|)$. □

5. Experimental results

In our experiments, we compared $r_B(s)$ and $r_B(s^{\mathrm{rev}})$ exhaustively for all strings up to length $n = 25$ for binary alphabet, and $n = 15$ for ternary alphabet. In order to avoid reproducing the same experiment twice, we restricted the experiments to only those strings which are strictly smaller than their reverse. In the following, we refer to these strings as *forward strings*. Note that our experiments exclude palindromes, as for those, trivially, $r_B(s) = r_B(s^{\mathrm{rev}})$. Since the number of palindromes of length n equals $2^{\lceil n/2 \rceil}$ for binary alphabet, and $3^{\lceil n/2 \rceil}$ for ternary alphabet, the number of forward strings of length n is $(2^n - 2^{\lceil n/2 \rceil})/2$ for binary alphabet, and $(3^n - 3^{\lceil n/2 \rceil})/2$ for ternary alphabet. In Tables 2 and 3 we give all forward strings of length $n = 3, 4, 5$ (binary) resp. $n = 3, 4$ (ternary) for illustration.

For each forward string s, we computed $r_B(s)$ and $r_B(s^{\mathrm{rev}})$, and from these, the functions $\rho_B(s) = \max(\frac{r_B(s)}{r_B(s^{\mathrm{rev}})}, \frac{r_B(s^{\mathrm{rev}})}{r_B(s)})$ (runs-ratio) and $\delta_B(s) = r_B(s) - r_B(s^{\mathrm{rev}})$ (runs-difference, see Section 3). Moreover, we are interested in the maximum runs-ratio ρ_B over all strings of length n, and therefore define $\rho_B^{\max}(n) = \max\{\rho_B(s) : |s| = n\}$. Note that by definition, $\rho_B(s) \geq 1$ for all strings. For the runs-difference δ, both maximum and minimum over all strings of length n are of interest, and we define $\delta_B^{\max}(n) = \max\{\delta_B(s) : |s| = n\}$ and $\delta_B^{\min}(n) = \min\{\delta_B(s) : |s| = n\}$.

We also computed two functions regarding the Lyndon factorization of s, namely $\ell(s)$, the number of Lyndon factors in the Lyndon factorization of s (i.e., the cardinality of the multiset $\mathcal{L}yn(s)$), and $d(s)$, the number of *distinct* Lyndon factors of s (i.e., the cardinality of the corresponding *set*).

The software to replicate the experiments in this section is publicly available at https://github.com/davidecenzato/Number-of-runs-BBWT.git.

5.1. Runs-ratio

In Table 4, we give an excerpt of our results on ρ_B, including $\rho_B^{\max}$, mean and standard deviation of ρ_B, and the percentage of strings s for which $\rho_B(s) = 1$ (for full results, see Table A.13 in the Appendix).

We observe that the mean, standard deviation, and the percentage of strings with ρ_B equal to 1 all decrease with increasing n. Thus, in spite of the fact that with increasing length an ever smaller percentage of strings has the same number of runs of the BBWT as its reverse, on average the runs-ratio gets closer and closer to 1 as the string length increases. As an example, in Fig. 3 (left) we plot the distribution of ρ_B for all forward strings of length $n = 21$. Note that the y-axis is in logarithmic scale.

Table 4
Statistics of ρ_B for all forward string s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 5, 10, 15, 25$, sampling string lengths for every 5th value: $\rho_B^{\max}(n)$, mean, standard deviation, and percentage of strings for which $\rho_B(s) = 1$. See Table A.13 for complete results.

$\rho_B(s)$				
n	$\rho_B^{\max}(n)$	mean	sd	$= 1$
5	2	1.445	0.479	50.00%
10	3	1.307	0.348	44.96%
15	3	1.244	0.259	38.13%
20	3.5	1.210	0.210	33.25%
25	4	1.187	0.180	29.63%

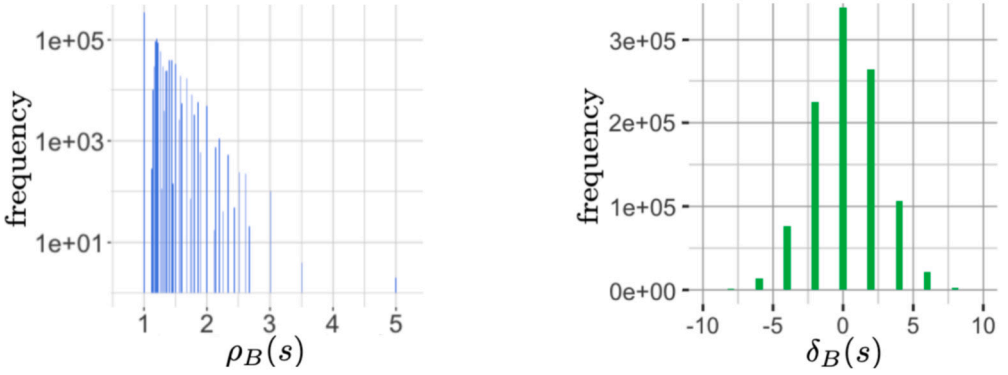

Fig. 3. Results for all 1,047,552 forward strings $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 21$. Left: histogram of $\rho_B(s)$; right: histogram of $\delta_B(s)$. Note that the left plot is in log-linear scale.

Table 5
The maximum runs-ratio ρ_B of binary strings of length n, for $n = 1, \ldots, 25$.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$\rho_B^{\max}(n)$	1	1	1	2	2	2	2	3	2.33	3	3	3	4	3	3	4	3	4	4	3.5	5	4	4	4	4

Table 6
Values of the function E for the first five values of m.

m	1	2	3	4	5
$E(m)$	1	4	8	13	21

There are very few strings that attain the maximum $\rho_B^{\max}(n)$; in Table A.14 in the Appendix we report a complete list of extremal words for all lengths up to $n = 25$. For example, for $n = 21$, only two strings out of a total of 1,047,552 forward strings have $\rho_B(s) = \rho_B^{\max}(21) = 5$: `aabaababaabaababaabab` and `ababbababbabbababbabb` (see Table 7).

Both of these are Lyndon rotations of standard words (namely of the Fibonacci words of order 7) and therefore $r_B(s) = 2$. On the other hand, $r_B(s^{\text{rev}}) = 10$, thus $\rho_B(s) = 5$.

Overall, we observe that approximately 54% (68 out of 126) of extremal words listed in Table A.14 are necklace rotations either of standard words or of powers of standard words and more than half of these strings (46 strings, or 36.5% of all extremal words) are Lyndon rotations of standard words. For example, for $n = 10$, two out of the three extremal words, `aababaabab` and `ababbababb`, are necklace rotations of a power of a standard word (`aababaabab` $= L(\mathtt{ababa})^2$) and (`ababbababb` $= L(\mathtt{babab})^2$). Moreover, all extremal words in our experiments with $\rho_B^{\max}(n) \geq 4$ are necklace rotations of a standard word or of a power of a standard word.

We now turn our focus to the maximum ρ_B for each length, $\rho_B^{\max}(n)$, and report $\rho_B^{\max}(n)$ over the binary alphabet in Table 5. It is interesting to note that the maximum $\rho_B^{\max}$ is not monotonically increasing. If we denote by $E : \mathbb{N} \mapsto \mathbb{N}$ the function that associates to each $m > 0$ the smallest n for which $\rho_B^{\max}(n) \geq m$, one can experimentally observe that for $3 \leq m \leq 5$, the function E follows the Fibonacci numbers, as can be seen in Table 6. That is, Fibonacci words appear as the shortest strings for which an increment in $\rho_B^{\max}(n)$ is observed. For instance, $n = 8$ is the smallest n which allows to obtain $\rho_B^{\max}(n) = 3$, and the two extremal cases are `aabaabab` and `ababbabb`, again Lyndon rotations of Fibonacci words. In Table 7, we collect all extremal words such that their length corresponds to a value in $\{E(m) : m = 1, 2, \ldots, 5\}$.

Table 7
Extremal words for ρ_B for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 1, \ldots, 25$. In this table we report only the strings of minimum length for increasing values of $\rho_B^{\max}(n)$. Rotations of standard words are marked with a 1 in the last column.

n	s	$\rho_B(s)$	$\delta_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
1	a	1	0	1	1	0
	b	1	0	1	1	0
4	aabb	2	2	4	2	0
	abab	2	-2	2	4	0
8	aabaabab	3	-4	2	6	1
	ababbabb	3	-4	2	6	1
13	aabaababaabab	4	-6	2	8	1
	ababbababbabb	4	-6	2	8	1
21	aabaababaabaababaabab	5	-8	2	10	1
	ababbababbabbababbabb	5	-8	2	10	1

Table 8
Statistics of δ_B for all forward string s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 5, 10, 15, 25$, sampling string lengths for every 5th value: $\delta_B^{\min}(n)$, $\delta_B^{\max}(n)$, mean and standard deviation. See Table A.15 for complete results.

	$\delta_B(s)$			
n	$\delta_B^{\min}(n)$	$\delta_B^{\max}(n)$	mean	sd
5	-2	2	0.333	1.435
10	-4	6	0.214	1.718
15	-8	8	0.236	2.063
20	-10	10	0.246	2.388
25	-14	14	0.253	2.679

Table 9
Extremal words for ρ_B for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 21, \ldots, 25$. See Table A.14 for the full list of extremal words of other lengths.

n	s	$\rho_B(s)$	$\delta_B(s)$	$\delta_B^{\min}(n)$
21	aabaababaabaababaabab	5.0	-8	-10
	ababbababbabbababbabb	5.0	-8	
22	aaabaaabaabaaabaaabaab	4.0	-6	-10
	aabaabaababaabaabaabab	4.0	-6	
	ababbabbabbababbabbabb	4.0	-6	
	abbabbbabbbabbabbbabbb	4.0	-6	
23	aaaabaaaabaaabaaaabaaab	4.0	-6	-12
	aabaababaababaababaabab	4.0	-6	
	ababbababbababbababbabb	4.0	-6	
	abbbabbbbabbbabbbbabbbb	4.0	-6	

n	s	$\rho_B(s)$	$\delta_B(s)$	$\delta_B^{\min}(n)$
24	aabaababaabaababaabaabab	4.0	-6	-12
	aabaabbabaabaabbbaabbbab	4.0	-12	
	aababaababababaababaabab	4.0	-6	
	abababbababbabababbababb	4.0	-6	
	ababbabbababbabbababbabb	4.0	-6	
25	aaabaaabaabaaabaabaaabaab	4.0	-6	-14
	aabaabaabaababaabaabaabab	4.0	-6	
	aabababaababababaabababab	4.0	-6	
	abababab babababab bababab b	4.0	-6	
	ababbabbabbababbabbabbabb	4.0	-6	
	abbabbbabbabbbabbabbbabbb	4.0	-6	

5.2. Runs-difference

Recall that the difference in the number of runs of the BBWT a string and of its reverse, $\delta_B(s)$, is always even (Corollary 1). In Table 8, we present an excerpt of the statistics on δ_B from our experiments, in particular, $\delta_B^{\min}(n), \delta_B^{\max}(n)$, mean, and standard deviation (for full results, see the Appendix, Table A.15). Again, we plot the distribution of δ_B for $n = 21$, see Fig. 3 (right). We note that $\delta_B^{\min} \in \{-\delta_B^{\max}, -\delta_B^{\max} + 2\}$, that the mean is very close to 0, and that the distribution is slightly skewed to the right. The latter seems to indicate that the number of runs of the BBWT of forward strings is on average a bit more than that of their reverse.

Lastly, we noticed that about 52% extremal cases for δ_B are different than the extremal cases for ρ_B (see Tables A.14, A.16 and A.17 in the Appendix). The percentage of such strings increases as n increases. In particular, all extremal words for ρ_B with $n = 21, \ldots, 25$ (see Table 9) are extremal only for ρ_B and not for δ_B with the exception of a single string with $\delta_B(s) = -12$.

5.3. Number of Lyndon factors and r_B

As the BBWT is the extended BWT of the multiset of Lyndon factors of the string, it is an obvious question whether the number of Lyndon factors is related to the number of runs of the BBWT.

In the two the scatter plots at the top of Fig. 4, we plot the number $\ell(s)$ of Lyndon factors against the number of runs of the BBWT of forward strings (left) and reverse strings (right) for $n = 21$. The color intensity of each dot in the scatter plots is proportional to the number of strings represented by that single dot. From these two plots we note that there are no strings with both a high number of

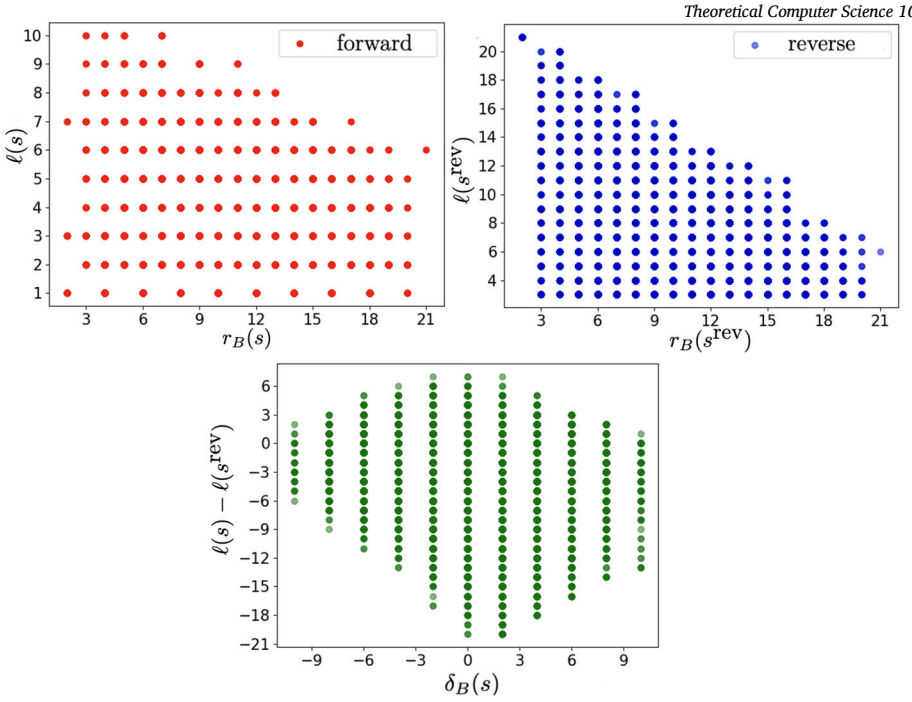

Fig. 4. Results for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 21$. Top: scatter plots of $r_B(s)$ and number of Lyndon factors of s (left) and s^{rev} (right). Bottom: scatter plot of $\delta_B(s)$ and $\ell(s) - \ell(s^{\text{rev}})$.

Lyndon factors and a high number of runs, as the top right corners of both graphs are empty. This is more evident in the plot for the reverse strings since $\ell(s^{\text{rev}})$ can be equal to the length of s^{rev}. Indeed, the results for forward and reverse strings are quite similar, with the exception that number of Lyndon factors are up to twice as much in reverse strings. The values of r_B span the same range on both forward and reverse strings.

We believe that this is related to the way we defined forward strings s; for example, since a forward string is always lexicographically strictly smaller than its reverse, only forward strings can be Lyndon words and thus have only one Lyndon factor. For instance, the Lyndon factorization of the reverse string $s^{\text{rev}} = \mathtt{bbbbababaabbbaaaa}$ results in eight length-one Lyndon factors: $\mathtt{b} \cdot \mathtt{b} \cdot \mathtt{b} \cdot \mathtt{b} \cdot \mathtt{ab} \cdot \mathtt{ab} \cdot \mathtt{aababb} \cdot \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{a}$, while the corresponding forward string is a Lyndon word, so its Lyndon factorization contains one single factor. In general, if a forward string s starts with a run of k a's and ends with m b's, then the prefix $\mathtt{a}^k$ will be the prefix of the first Lyndon factor of s, and the suffix $\mathtt{b}^m$ the suffix of the last Lyndon factor of s. On the other hand, the Lyndon factorization of s^{rev} will start with m factors b and end with k factors a.

We explicitly studied the connection between δ_B and the difference in the number of Lyndon factors. In the bottom scatter plot in Fig. 4, we plot the difference in the number of Lyndon factors (forward - reverse) against $\delta_B(s)$ for $n = 21$. From this plot we can observe that 32% of the strings analyzed are found on the vertical line for $\delta_B(s) = 0$. The data points concentrate around the vertical region between $\delta_B(s) = -2.5$ and $\delta_B(s) = 2.5$ indicating little to no difference in the number of runs of s and s^{rev}.

Note that the data points in this region span the whole y-axis, indicating that a large absolute value in terms of the difference in the number of Lyndon factors does not necessarily result in very different r_B values. The other region of the scatter plot that has many data points is the horizontal area between values 0 and -5 on the y-axis. Here data points span the whole x-axis, indicating that a small difference in terms of the number of Lyndon factors can correspond to all values of δ_B.

In fact, we observed strings s with a large difference between $\ell(s)$ and $\ell(s^{\text{rev}})$ but $\delta_B(s) = 0$. An example is $s = \mathtt{aaaabaaabaabbaaba}$ $\mathtt{bbabbbab}$, a Lyndon word, thus $\ell(s) = 1$. The Lyndon factorization of its reverse results in 11 factors, namely $\mathtt{babbbabbabaabbaaba}$ $\mathtt{aabaaaa} = \mathtt{b} \cdot \mathtt{abbb} \cdot \mathtt{abb} \cdot \mathtt{ab} \cdot \mathtt{aabb} \cdot \mathtt{aab} \cdot \mathtt{aaab} \cdot \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{a} \cdot \mathtt{a}$, but $r_B(s) = r_B(s^{\text{rev}}) = 16$, thus $\delta_B(s) = 0$.

According to these results, we could not derive any clear relationship between these two measures since, in most cases, even if two strings have two very different factorizations, that does not necessarily translate into a significant difference in the number of runs.

We then considered the number of *distinct* Lyndon factors d of forward strings and their reverse, since we know from Lemma 1 that the number of runs of a multiset does not depend on the multiplicity of each element. In the histogram in Fig. 5, we plot the distribution of the difference in the number of distinct Lyndon factors between a forward string and its reverse for $n = 21$. This distribution is unimodal, with mode -1 (27% of the strings analyzed) and slightly left-skewed. It spans values from -6 to 4, with a mean of -1.168. These results are consistent with the conclusion that, in general, the number of distinct Lyndon factors in the reverse strings is slightly higher than that of their forward strings.

In the scatter plots in Fig. 6 (top) we plot the number of distinct Lyndon factors against the number of runs of the BBWT of forward strings (left) and their reverse (right) for $n = 21$. In these two scatter plots, we can observe a similar behavior as the one for

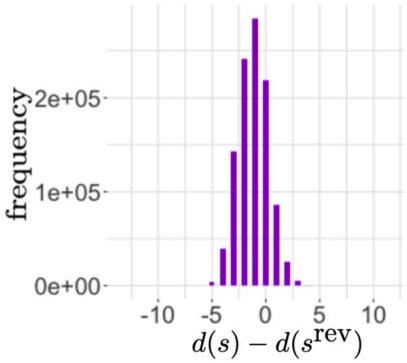

Fig. 5. Histogram of $d(s) - d(s^{\text{rev}})$ for all 1,047,552 forward strings $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 21$.

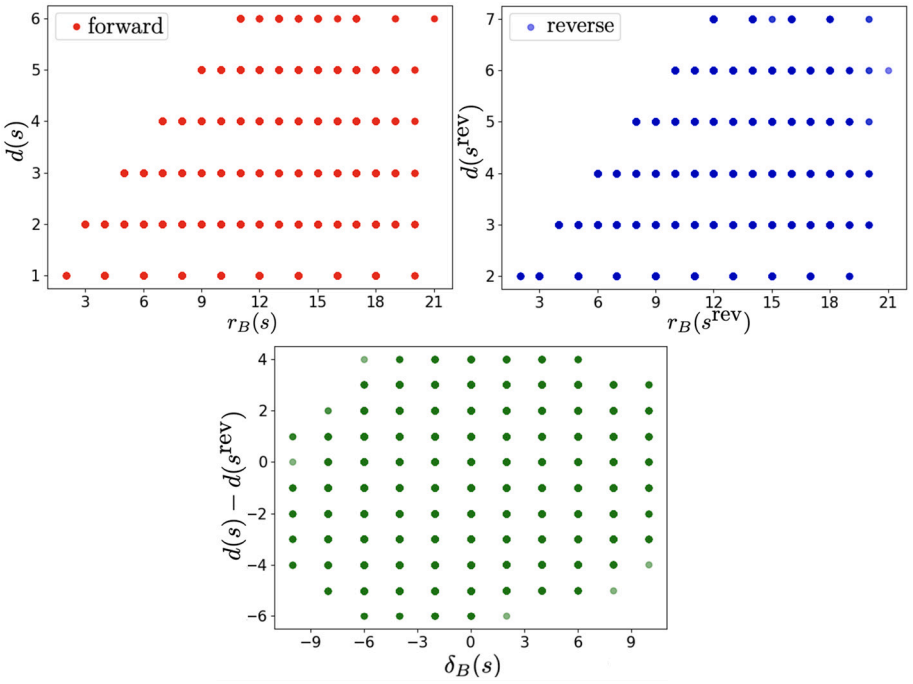

Fig. 6. Results for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 21$. Top: Scatter plots of $r_B(s)$ and $d(s)$ (left) and s^{rev} and $d(s^{\text{rev}})$ (right). Bottom: Scatter plot of $\delta_B(s)$ and $d(s) - d(s^{\text{rev}})$.

the absolute number of Lyndon factors. In particular, we observed that there are no strings in the top left sides of the plots (top plots), meaning that there are no strings with both a low number of runs and a high number of distinct Lyndon factors. However, differently than observed in Fig. 4 (top plots), the range of values on the x-axis is similar between forward and reverse strings, suggesting that the Lyndon factorization of the reverse strings tends to generate more repeated Lyndon factors (see Table A.19) than the Lyndon factorization of a forward string.

In the bottom scatter plot in Fig. 6, we plot the difference in the number of distinct Lyndon factors (forward - reverse) against $\delta_B(s)$ for $n = 21$. With respect to the scatter plot at the bottom of Fig. 4, here we can observe how removing the identical Lyndon factors complicates finding a relationship between the Lyndon factorization and δ_B. Data points cover the whole are of the plot with the exception of the corners. This is consistent with the distributions of $\delta_B(s)$ and $d(s) - d(s^{\text{rev}})$ that are unimodal and almost symmetrical, with the majority of the values close to the mean values, that are 0.247 and -1.168, respectively, for $n = 21$. Thus few strings have extremal values for one of the two distributions and, from this plot, we cannot spot strings that are both extremal for $\delta_B(21)$ and for the difference in the number of distinct Lyndon factors.

Thus, also in this case, we could not detect a clear connection with δ_B.

5.4. Ternary alphabet

We present the results over a ternary alphabet for ρ_B in Table 10. For $\rho_B(s)$, we observed again the same pattern with average values always close to 1 for all lengths n, and larger n values associated with larger $\rho_B^{\max}(n)$ values. In Table A.20 in the appendix we report all extremal words; 65% are Lyndon words and almost 19% are powers of Lyndon rotations (data not shown).

Table 10
Statistics of ρ_B for all forward strings s on $\Sigma = \{a, b, c\}$ and $n = 3, \ldots, 15$: $\rho_B^{\max}(n)$, mean, standard deviation and percentage of strings for which $\rho_B(s) = 1$.

	$\rho_B(s)$			
n	$\rho_B^{\max}(n)$	mean	sd	$= 1$
3	1	1	0	100%
4	2	1.240	0.370	61.11%
5	2	1.295	0.334	41.67%
6	2	1.262	0.292	40.17%
7	2.33	1.256	0.257	32.19%
8	3	1.239	0.254	31.64%
9	2.67	1.228	0.225	27.98%
10	3	1.216	0.214	26.04%
11	3.33	1.206	0.201	24.53%
12	3	1.197	0.190	23.30%
13	4	1.190	0.179	22.06%
14	3.67	1.182	0.171	21.09%
15	3.67	1.176	0.163	20.29%

Table 11
Statistics of δ_B for $\Sigma = \{a, b, c\}$ and $n = 3, \ldots, 15$: minimum ($\delta_B^{\min}(n)$) and maximum values ($\delta_B^{\max}(n)$), mean and standard deviation.

	$\delta_B(s)$			
n	$\delta_B^{\min}(n)$	$\delta_B^{\max}(n)$	mean	sd
3	0	0	0	0
4	-2	2	0	0.956
5	-2	2	0.139	1.139
6	-3	3	0.048	1.183
7	-4	3	0.066	1.301
8	-5	4	0.048	1.405
9	-5	5	0.048	1.498
10	-6	6	0.040	1.586
11	-7	6	0.038	1.672
12	-7	7	0.035	1.753
13	-8	8	0.033	1.831
14	-9	9	0.032	1.906
15	-9	10	0.032	1.978

Table 12
Extremal words for ρ_B for $\Sigma = \{a, b, c\}$ and $n = 12, \ldots, 15$. See Table A.20 for the full list of extremal words of other lengths.

n	s	$\rho_B(s)$	$\delta_B(s)$	$\delta_B^{\max}(n)$	$\delta_B^{\min}(n)$
12	aaaaabbbabba	3.0	6	7	-7
	aaaaacccacca	3.0	6		
	aaabbbbabbba	3.0	6		
	aaaccccaccca	3.0	6		
	aabababababab	3.0	-4		
	aacacaacacac	3.0	-4		
	abababbabab	3.0	-4		
	abacacbabacb	3.0	-6		
	abacacbacacb	3.0	-6		
	abcabcaacaac	3.0	-6		
	abcacabcbcac	3.0	-6		
	abcbcabcbcbc	3.0	-6		
	acacaccacacc	3.0	-4		
	babaabaabbbb	3.0	6		
	bbbbbcccbccb	3.0	6		
12	bbbccccbcccb	3.0	6	7	-7
	bbcbbcabcabc	3.0	-6		
	bbcbcbbcbcbc	3.0	-4		
	bcbcbccbcbcc	3.0	-4		
	cacaacaacccc	3.0	6		
	cbcbbcbbcccc	3.0	6		
13	aabaababaabab	4.0	-6	8	-8
	aacaacacaacac	4.0	-6		
	ababbababbabb	4.0	-6		
	acaccacaccacc	4.0	-6		
	bbcbbcbcbbcbc	4.0	-6		
	bcbccbcbccbcc	4.0	-6		
14	abacacbabacacb	3.67	-8	9	-9
15	abbcabbcbbbcbbc	3.67	-8	10	-9
	acaccacacbacacb	3.67	-8		

We refer to a string s as *perfectly clustering* if BBWT(s) has exactly |alph(s)| runs. Nearly 68% of the extremal words for ρ_B are perfectly clustering. However, we could not detect any clear pattern since we observed extremal words that are not Lyndon, nor perfectly clustering. An example is the string babacc, that is not a Lyndon word, nor perfectly clustering. It is an extremal word since $\rho_B^{\max}(6) = \rho_B(\texttt{babacc}) = 2$, indeed $r_B(s) = 6$, $r_B(s^{\text{rev}}) = 3$.

The distribution of δ_B is slightly negatively skewed, even though less skewed than the distribution for the binary alphabet (see Table 11).

The extremal words for ρ_B and δ_B do not coincide, in particular, nearly 63% of the extremal words for ρ_B are not extremal for δ_B. To this set of strings belong all forward strings with $n = 12, \ldots, 15$ (see Table 12). The complete list of extremal words for δ_B can be found in the Appendix: Tables A.21 and A.22.

Regarding the number of Lyndon factors, s^{rev} is again associated with a larger number of Lyndon factors on average (see Tables A.23 and A.24 in the Appendix), but, as for the binary alphabet, we were not able to find a clear correlation between the number of Lyndon factors and δ_B.

6. Conclusion

In this paper, we gave first results on the number r_B of runs of the Bijective BWT (BBWT) as a repetitiveness measure. In particular, we proved that r_B can exhibit a $\Theta(\log n)$ multiplicative difference between a string and its reverse, where n is the length of the string. We gave an infinite family of strings for which this holds. This result is analogous to that on the BWT [19].

We also presented new theoretical results on the BBWT, including a characterization of strings on which the BWT and the BBWT coincide, proving that these strings are exactly the class of necklaces. We also characterized binary strings on which the BBWT exhibits the maximal clustering effect, paralleling a famous result by Mantaci et al. for the BWT [29].

Open problems include the question whether our lower bound of $\Theta(\log n)$ on the runs-ratio $\rho_B(s)$ is tight. The fact that the increase in the function $\rho_B^{\max}$ is attained by lengths n which are Fibonacci numbers (and that the extremal words for those lengths are actually Fibonacci words) seems to indicate that the bound is indeed tight, in other words, that our example family actually attains the maximum runs-ratio. In this context, we conjecture that from some length n onwards, all extremal binary words for ρ_B are rotations

of standard words or of powers of standard words. Another interesting question is whether the results of Section 4 can be extended to standard words in general.

We are further interested in the function d, the number of distinct strings from the Lyndon factorization, and what role this function plays in the difference between the measures r (number of runs of the BWT) and r_B (number of runs of the BBWT).

Finally, all conjugates of a string have the same r, while this is not the case for r_B. Is it possible to derive $r_B(t)$ from $r_B(s)$, for conjugates t of a string s? Clearly, the Lyndon factorization of s and t play a role here. We believe that similar results to our Proposition 3 could help in better understanding how the Lyndon factors of a string impact r_B.

Author contributions

All authors contributed equally.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank two anonymous reviewers for their careful reading of our manuscript and for suggesting a new open problem. DC is funded by the European Union (ERC, REGINDEX, 101039208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. EB is partially supported by the Academy of Finland via grant 351150. ZsL and GR are partially funded by MUR PRIN project no. 2022YRB97K - 'PINC' (Pangenome INformatiCs: From Theory to Applications), and by the INdAM-GNCS project no. E53C23001670001 (Compressione, indicizzazione, analisi e confronto di dati biologici).

Appendix A. Further experimental results

In this appendix, we give statistics on the multiplicative and additive difference in r_B of a string and its reverse for the binary alphabet: Tables A.13 and A.15 respectively. We also include a table reporting all extremal words for ρ_B (Table A.14) and two with all extremal words for δ_B (Tables A.16 and A.17). Table A.18 reports some statistics on the difference in the number of Lyndon factors, $\ell'(s)$, and Table A.19 reports the same statistics on the difference in the number of distinct Lyndon factors, $d(s)$. Finally, we include the tables listed above with results for the ternary alphabet, with the exception of the first two, since those are fully included in the main paper. The tables for the ternary alphabet are: Tables A.20, A.21, A.22, A.23 and A.24.

A.1. Multiplicative difference in r_B of a string and its reverse

Table A.13
Statistics of ρ_B for all forward strings s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 3, \ldots, 25$: $\rho_B^{\max}(n)$, mean, standard deviation, and percentage of strings for which $\rho_B(s) = 1$.

	$\rho_B(s)$			
n	$\rho_B^{\max}(n)$	mean	sd	$= 1$
3	1	1	0	100%
4	2	1.333	0.516	66.67%
5	2	1.445	0.479	50.00%
6	2	1.358	0.435	57.14%
7	2	1.372	0.38	46.43%
8	3	1.345	0.427	51.67%
9	2.33	1.326	0.349	45.42%
10	3	1.307	0.348	44.96%
11	3	1.290	0.330	43.55%
12	3	1.271	0.311	44.64%
13	4	1.265	0.288	40.40%
14	3	1.254	0.275	39.49%
15	3	1.244	0.259	38.13%
16	4	1.236	0.248	37.14%
17	3	1.229	0.236	35.84%
18	4	1.222	0.227	35.07%
19	4	1.216	0.218	34.05%
20	3.5	1.210	0.210	33.25%
21	5	1.205	0.203	32.33%
22	4	1.200	0.197	31.62%
23	4	1.195	0.191	30.90%
24	4	1.191	0.185	30.25%
25	4	1.187	0.180	29.63%

Table A.14
All extremal forward strings s on $\Sigma = \{a, b\}$ for ρ_B and $n = 3, \ldots, 25$. Rotations of standard words are marked with a 1 in the last column.

n	s	$\rho_B(s)$	$\delta_B(s)$	$r_B(s)$	$r_B(s^{rev})$	rot.s of st. words
3	aab	1.0	0	2	2	1
	abb	1.0	0	2	2	1
4	aabb	2.0	2	4	2	0
	abab	2.0	-2	2	4	0
5	aaabb	2.0	2	4	2	0
	aabab	2.0	-2	2	4	1
	aabbb	2.0	2	4	2	0
	ababb	2.0	-2	2	4	1
6	aaaabb	2.0	2	4	2	0
	aaabbb	2.0	2	4	2	0
	aabaab	2.0	-2	2	4	0
	aabbbb	2.0	2	4	2	0
	ababab	2.0	-2	2	4	0
	abbabb	2.0	-2	2	4	0
7	aaaaabb	2.0	2	4	2	0
	aaaabbb	2.0	2	4	2	0
	aaabaab	2.0	-2	2	4	1
	aaabbbb	2.0	2	4	2	0
	aababab	2.0	-2	2	4	1
	aabbbbb	2.0	2	4	2	0
	abababb	2.0	-2	2	4	1
	abbabbb	2.0	-2	2	4	1
8	aabaabab	3.0	-4	2	6	1
	ababbabb	3.0	-4	2	6	1
9	aaaabbaba	2.33	4	7	3	0
	aaabbabba	2.33	4	7	3	0
	aabbbabba	2.33	4	7	3	0
	baabaabbb	2.33	4	7	3	0
	babaabbbb	2.33	4	7	3	0
10	aaabbbabba	3.0	6	9	3	0
	aababaabab	3.0	-4	2	6	0
	ababbababb	3.0	-4	2	6	0
11	aaaabbbabba	3.0	6	9	3	0
	aaabaaabaab	3.0	-4	2	6	1
	aabaabaabab	3.0	-4	2	6	1
	ababbabbabb	3.0	-4	2	6	1
	abbabbbabbb	3.0	-4	2	6	1
	babaabaabbb	3.0	6	9	3	0
12	aaaaabbbabba	3.0	6	9	3	0
	aaabbbbabbba	3.0	6	9	3	0
	aababaababab	3.0	-4	2	6	1
	abababbababb	3.0	-4	2	6	1
	babaabaabbbb	3.0	6	9	3	0
13	aabaababaabab	4.0	-6	2	8	1
	ababbababbabb	4.0	-6	2	8	1
14	aaaaaaabbbabba	3.0	6	9	3	0
	aaaaabbbbabbba	3.0	6	9	3	0
	aaaabaaaabaaab	3.0	-4	2	6	1
	aaaabbbabbabba	3.0	6	9	3	0
	aaabaabaaabaab	3.0	-4	2	6	0
	aaabbbabbbabba	3.0	6	9	3	0
	aaabbbbbabbbba	3.0	6	9	3	0
	aabaabaabaabab	3.0	-4	2	6	1
	aabababaababab	3.0	-4	2	6	0
	abababbabababb	3.0	-4	2	6	0
	ababbabbabbabb	3.0	-4	2	6	1
	abbabbbabbabbb	3.0	-4	2	6	0
	abbbabbbbabbbb	3.0	-4	2	6	1
	baabaaabaaabbb	3.0	6	9	3	0
	babaabaabaabbb	3.0	6	9	3	0
	babaabaabbbbbb	3.0	6	9	3	0
	babaababaabbbb	3.0	6	9	3	0
15	aaaaaaaabbbabba	3.0	6	9	3	0
	aaaaaabbbbabbba	3.0	6	9	3	0
	aaaaabbbabbabba	3.0	6	9	3	0
	aaaabbbabbbabba	3.0	6	9	3	0
	aaaabbbbbabbbba	3.0	6	9	3	0
	aaabaaabaaabaab	3.0	-4	2	6	1
	aabaabbbabaabab	3.0	-8	4	12	0
	aababaababaabab	3.0	-4	2	6	0
	ababbababbababb	3.0	-4	2	6	0
	abbabbbabbbabbb	3.0	-4	2	6	1
	baabaaabaaabbbb	3.0	6	9	3	0
	babaabaabaabbbb	3.0	6	9	3	0
	babaabaabbbbbbb	3.0	6	9	3	0
	babaababaabbbbb	3.0	6	9	3	0
	bababaababaabbb	3.0	6	9	3	0
16	aabaababaabaabab	4.0	-6	2	8	0
	ababbabbababbabb	4.0	-6	2	8	0
17	aaaaaaaaaabbbabba	3.0	6	9	3	0
	aaaaaaaabbbbabbba	3.0	6	9	3	0
	aaaaaaabbbabbabba	3.0	6	9	3	0
	aaaaaabbbabbbabba	3.0	6	9	3	0
	aaaaaabbbbbabbbba	3.0	6	9	3	0
	aaaaabaaaaabaaaab	3.0	-4	2	6	1
	aaaaabababaababaa	3.0	6	9	3	0
	aaaabbabbababbaba	3.0	6	9	3	0
	aaaabbbabbbabbabba	3.0	6	9	3	0
	aaaabbbbabbbabbba	3.0	6	9	3	0
	aaaabbbbbbabbbbba	3.0	6	9	3	0
	aaabaabaaabaabaab	3.0	-4	2	6	1
	aaababbaaababbabb	3.0	-8	4	12	0
	aaabbaabaaabbbaab	3.0	-8	4	12	0
	aaabbbabbabbbabba	3.0	6	9	3	0
	aaabbbbabbbbabbba	3.0	6	9	3	0
18	aaabaaabaabaaabaab	4.0	-6	2	8	1
	aabaababaababaabab	4.0	-6	2	8	1
	ababbababbababbabb	4.0	-6	2	8	1
	abbabbbabbabbbabbb	4.0	-6	2	8	1
19	aabaabaababaabaabab	4.0	-6	2	8	1
	aababaababababaababab	4.0	-6	2	8	1
	abababbababababbababb	4.0	-6	2	8	1
	ababbabbababbabbabb	4.0	-6	2	8	1
20	aabaabbbabaababaabab	3.5	-10	4	14	0
	abbabbaabbbabaabbbab	3.5	-10	4	14	0
21	aabaababaabaababaabab	5.0	-8	2	10	1
	ababbababbabbababbabb	5.0	-8	2	10	1
22	aaabaaabaabaaabaaabaab	4.0	-6	2	8	0
	aabaabaababaabaabaabab	4.0	-6	2	8	0
	ababbabbabbababbabbabb	4.0	-6	2	8	0
	abbabbbabbbabbabbbabbb	4.0	-6	2	8	0
23	aaaabaaaabaaabaaaabaaab	4.0	-6	2	8	1
	aabaababaababaababaabab	4.0	-6	2	8	1
	ababbababbababbababbabb	4.0	-6	2	8	1
	abbbabbbbabbbabbbbabbbb	4.0	-6	2	8	1
24	aabaababaabaababaabaabab	4.0	-6	2	8	0
	aabaabbabaabaabbbaabbbab	4.0	-12	4	16	0
	aababaababababaababababab	4.0	-6	2	8	0
	abababbababbabababbababb	4.0	-6	2	8	0
	ababbabbababbabbababbabb	4.0	-6	2	8	0
25	aaabaaabaabaaabaabaaabaab	4.0	-6	2	8	1
	aabaabaabaababaabaabaabab	4.0	-6	2	8	1
	aababababaababababaababab	4.0	-6	2	8	1
	ababababbababababbababab b	4.0	-6	2	8	1
	ababbabbabbababbabbabbabb	4.0	-6	2	8	1
	abbabbbabbabbbabbabbbabbb	4.0	-6	2	8	1

A.2. Additive difference in r_B of a string and its reverse

Table A.15
Statistics of δ_B for all forward strings s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 3, \ldots, 25$: minimum ($\delta_B^{\min}(n)$) and maximum values ($\delta_B^{\max}(n)$), mean and standard deviation.

	$\delta_B(s)$			
n	$\delta_B^{\min}(n)$	$\delta_B^{\max}(n)$	mean	sd
3	0	0	0	0
4	-2	2	0	1.265
5	-2	2	0.333	1.435
6	-2	2	0.143	1.325
7	-2	2	0.214	1.461
8	-4	4	0.200	1.586
9	-4	4	0.233	1.640
10	-4	6	0.214	1.718
11	-4	6	0.224	1.784
12	-4	6	0.225	1.852
13	-6	6	0.232	1.927
14	-6	6	0.233	1.995
15	-8	8	0.236	2.063
16	-8	8	0.239	2.130
17	-8	8	0.240	2.198
18	-8	10	0.242	2.262
19	-10	10	0.244	2.326
20	-10	10	0.246	2.388
21	-10	10	0.247	2.449
22	-10	12	0.249	2.508
23	-12	14	0.25	2.566
24	-12	14	0.251	2.623
25	-14	14	0.253	2.679

Table A.16
All extremal forward strings s on $\Sigma = \{a, b\}$ with non-negative values of δ_B and $n = 3, \ldots, 25$. Rotations of standard words are marked with a 1 in the last column.

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
3	aab	0	1.0	2	2	1
	abb	0	1.0	2	2	1
4	aabb	2	2.0	4	2	0
5	aaabb	2	2.0	4	2	0
	aabba	2	1.67	5	3	0
	aabbb	2	2.0	4	2	0
	baabb	2	1.67	5	3	0
6	aaaabb	2	2.0	4	2	0
	aaabba	2	1.67	5	3	0
	aaabbb	2	2.0	4	2	0
	aabbba	2	1.67	5	3	0
	aabbbb	2	2.0	4	2	0
	baaabb	2	1.67	5	3	0
	baabbb	2	1.67	5	3	0
7	aaaaabb	2	2.0	4	2	0
	aaaabba	2	1.67	5	3	0
	aaaabbb	2	2.0	4	2	0
	aaababa	2	1.67	5	3	0
	aaabbab	2	1.5	6	4	0
	aaabbba	2	1.67	5	3	0
	aaabbbb	2	2.0	4	2	0
	aababbb	2	1.5	6	4	0
	aabbaba	2	1.67	5	3	0
	aabbbba	2	1.67	5	3	0
	aabbbbb	2	2.0	4	2	0
	abaaabb	2	1.5	6	4	0
	abaabbb	2	1.5	6	4	0
	baaaabb	2	1.67	5	3	0
	baaabbb	2	1.67	5	3	0
	baabbbb	2	1.67	5	3	0
	babaabb	2	1.67	5	3	0
	bababbb	2	1.67	5	3	0
8	aaabbaba	4	2.33	7	3	0
	abaaabbb	4	2.0	8	4	0
	babaabbb	4	2.33	7	3	0
9	aaaabbaba	4	2.33	7	3	0
	aaabaabbb	4	2.0	8	4	0
	aaabbabba	4	2.33	7	3	0
	aaabbbabb	4	2.0	8	4	0
	aabbbabba	4	2.33	7	3	0
	abaaaabbb	4	2.0	8	4	0
	abaaabbbb	4	2.0	8	4	0
	baabaabbb	4	2.33	7	3	0
	babaaabbb	4	1.8	9	5	0
	babaabbbb	4	2.33	7	3	0
10	aaabbbabba	6	3.0	9	3	0
11	aaaabbbabba	6	3.0	9	3	0
	babaabaabbb	6	3.0	9	3	0
12	aaaaabbbabba	6	3.0	9	3	0
	aaabbababbbb	6	2.5	10	4	0
	aaabbbbabbba	6	3.0	9	3	0
	babaaabaabbb	6	2.2	11	5	0
	babaabaabbbb	6	3.0	9	3	0
13	aaaaaabbbabba	6	3.0	9	3	0
	aaaabbababbbb	6	2.5	10	4	0
	aaaabbbbabbba	6	3.0	9	3	0
	aaabaabbbabba	6	1.86	13	7	0
	aaababbaabbbb	6	2.0	12	6	0
	aaabbababbbbb	6	2.5	10	4	0
	aaabbabbbbaba	6	2.2	11	5	0
	aaabbbabbabba	6	3.0	9	3	0
13	abbaaabaabbbb	6	2.0	12	6	0
	babaaabaabbbb	6	2.2	11	5	0
	babaabaaaabbb	6	2.2	11	5	0
	babaabaabbbbb	6	3.0	9	3	0
	babaababaabbb	6	3.0	9	3	0
	babbaabaabbbb	6	2.2	11	5	0
	bbabaaabaabbb	6	2.2	11	5	0
14	aaaaaaabbbabba	6	3.0	9	3	0
	aaaaabbababbbb	6	2.5	10	4	0
	aaaaabbbbabbba	6	3.0	9	3	0
	aaaabaabbbabba	6	1.86	13	7	0
	aaaababbaababa	6	2.2	11	5	0
	aaaababbaabbbb	6	2.0	12	6	0
	aaaababbbaabaa	6	2.2	11	5	0
	aaaabbabaababb	6	2.0	12	6	0
	aaaabbababbbbb	6	2.5	10	4	0
	aaaabbabbbbaba	6	2.2	11	5	0
	aaaabbbabaabba	6	1.86	13	7	0
	aaaabbbabbabba	6	3.0	9	3	0
	aaabaabbbabbaa	6	1.86	13	7	0
	aaababbaabbbaa	6	2.2	11	5	0
	aaababbaabbbbb	6	2.0	12	6	0
	aaabbababbbbb	6	2.5	10	4	0
	aaabbababbbaba	6	2.2	11	5	0
	aaabbababbbbbb	6	2.5	10	4	0
	aaabbabbbbbaba	6	2.2	11	5	0
	aaabbbbabaabba	6	2.2	11	5	0
	aaabbbbabbbabb	6	2.5	10	4	0
	aaabbbbbabbbba	6	3.0	9	3	0
	aabaaaabbbabba	6	1.86	13	7	0
	aabbabaaaabbbb	6	2.0	12	6	0
	aabbababbbbaba	6	2.2	11	5	0
	ababbaabaabbbb	6	2.0	12	6	0
	abbaaabaabbbbb	6	2.0	12	6	0
	abbbabaabbaaab	6	1.75	14	8	0
	baabaaabaaabbb	6	3.0	9	3	0
	babaaaabaaabbb	6	2.2	11	5	0
	babaaabaabbbbb	6	2.2	11	5	0
	babaaabababbbb	6	2.2	11	5	0
	babaabaaaaabbb	6	2.2	11	5	0
	babaabaaaabbbb	6	2.2	11	5	0
	babaabaabaabbb	6	3.0	9	3	0
	babaabaabbbbbb	6	3.0	9	3	0
	babaababaaabbb	6	2.2	11	5	0
	babaababaabbbb	6	3.0	9	3	0
	babbaaabaabbbb	6	1.86	13	7	0
	babbaabaabbbbb	6	2.2	11	5	0
	babbaababababbb	6	2.2	11	5	0
	bbabaaabaabbbb	6	2.2	11	5	0
	bbabbabaaabbbb	6	2.2	11	5	0
15	aaabbababbbbaba	8	2.6	13	5	0
16	aaaabbabaabaabbbb	8	2.33	14	6	0
	aaaabbababbbbaba	8	2.6	13	5	0
	aaaabbbabbaababa	8	2.14	15	7	0
	aaabbababbbbbaba	8	2.6	13	5	0
	babaabaabbaabbbb	8	2.6	13	5	0
	bbabaaabbababbbb	8	2.6	13	5	0
17	aaaaabbabaabababbbb	8	2.33	14	6	0
	aaaaabbababbbbaba	8	2.6	13	5	0
	aaaaabbbabbaababa	8	2.14	15	7	0
	aaaababaabababbbaab	8	2.33	14	6	0
	aaaababbaabaabbbb	8	2.33	14	6	0
	aaaabbabaababababbb	8	2.33	14	6	0
	aaaabbabaababbbbb	8	2.33	14	6	0

Table A.16 (*continued*)

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
17	aaaabbababbbbbaba	8	2.6	13	5	0
	aaaabbbabaababbba	8	2.14	15	7	0
	aaaabbbabbaababaa	8	2.14	15	7	0
	aaaabbbabbabaabba	8	2.14	15	7	0
	aaaabbbbabaababbb	8	2.0	16	8	0
	aaabababbbbaabbba	8	2.6	13	5	0
	aaabababbbbabbbab	8	2.33	14	6	0
	aaabbabaabababbbb	8	2.33	14	6	0
	aaabbababbbbababa	8	2.6	13	5	0
	aaabbababbbbbbaba	8	2.6	13	5	0
	aaabbabbbababbaba	8	2.6	13	5	0
	aabbabbbbababbaba	8	2.6	13	5	0
	abaabbaaabaaabbbb	8	2.33	14	6	0
	babaaabaabbaabbbb	8	2.14	15	7	0
	babaabaaaabaaabbb	8	2.6	13	5	0
	babaabaaaabbaabbb	8	2.14	15	7	0
	babaabaaabbaabbbb	8	2.14	15	7	0
	babaabaabbaabbbbb	8	2.6	13	5	0
	bababaababaaaabbb	8	2.6	13	5	0
	bababbaababaabbbb	8	2.6	13	5	0
	babbaababaaaabbbb	8	2.14	15	7	0
	bbaaaabbabaaabbbb	8	2.6	13	5	0
	bbabaaaabbababbbb	8	2.6	13	5	0
	bbabaaabbababbbbb	8	2.6	13	5	0
	bbabaabbabaaabbbb	8	2.6	13	5	0
18	aaaabbabaabababbbb	10	2.67	16	6	0
	aaabbabbbbababbaba	10	3.0	15	5	0
	babaabaaaabbaabbbb	10	2.43	17	7	0
19	aaaaabbabaabababbbb	10	2.67	16	6	0
	aaaababaababbbaabaa	10	3.0	15	5	0
	aaaabbabaabababbbbb	10	2.67	16	6	0
	aaaabbababbbbaababa	10	2.11	19	9	0
	aaaabbabbbbababbaba	10	3.0	15	5	0
	aaabbabbbbbababbaba	10	3.0	15	5	0
	babaaaabaaabbaabbbb	10	2.43	17	7	0
	babaabaaaaabbaabbbb	10	2.43	17	7	0
	babaabaaaabbaabbbbb	10	2.43	17	7	0
	bababbaababaaaabbbb	10	2.43	17	7	0
20	aaaaaabbabaabababbbb	10	2.67	16	6	0
	aaaaababaababbbaabaa	10	3.0	15	5	0
	aaaaabbabaabababbbbb	10	2.67	16	6	0
	aaaaabbababbbbaababa	10	2.11	19	9	0
	aaaaabbabbbbababbaba	10	3.0	15	5	0
	aaaababaababbbbaabaa	10	3.0	15	5	0
	aaaabbabaababababbbb	10	2.67	16	6	0
	aaaabbabaabababbbbbb	10	2.67	16	6	0
	aaaabbababaabababbbb	10	2.67	16	6	0
	aaaabbababbbbbaababa	10	2.11	19	9	0
	aaaabbabbbbbababbaba	10	3.0	15	5	0
	aaaabbbbabbbaabbabba	10	2.43	17	7	0
	aaaabbbbabbbabaababb	10	2.25	18	8	0
	aaabbabbbbbbababbaba	10	3.0	15	5	0
	aababbaaaababaaabbbb	10	2.25	18	8	0
	aabbabaaaabbbbababbb	10	2.25	18	8	0
	abbabaaabbbbabaaaabb	10	2.25	18	8	0
	baababbaaababaaabbbb	10	2.43	17	7	0
	babaaaabaaabbaabbbbb	10	2.43	17	7	0
	babaabaaaaaabbaabbbb	10	2.43	17	7	0
	babaabaaaaabbaabbbbb	10	2.43	17	7	0
	babaabaaaababbaabbbb	10	2.11	19	9	0
	babaabaaaabbaabbbbbb	10	2.43	17	7	0
	babaababaababbaabbbb	10	3.0	15	5	0

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
20	babaababbaabaaaabbbb	10	2.43	17	7	0
	babaabbaaababaaabbbb	10	2.43	17	7	0
	bababbaaaababaaabbbb	10	2.43	17	7	0
	bababbaababaaaaabbbb	10	2.43	17	7	0
	bababbaababaaaabbbbb	10	2.43	17	7	0
	bababbbbaababaaaabbb	10	2.43	17	7	0
	babbaabaaaababaabbbb	10	2.43	17	7	0
	bbabaabaabbabaaabbbb	10	3.0	15	5	0
	bbabbabaaabbabaabbbb	10	3.0	15	5	0
21	aaaaaaabbabaabababbbb	10	2.67	16	6	0
	aaaaaababaababbbaabaa	10	3.0	15	5	0
	aaaaaabbabaabababbbbb	10	2.67	16	6	0
	aaaaaabbababbbbaababa	10	2.11	19	9	0
	aaaaaabbabbbbababbaba	10	3.0	15	5	0
	aaaaababaababbbbaabaa	10	3.0	15	5	0
	aaaaabbabaababababbbb	10	2.67	16	6	0
	aaaaabbabaabababbbbbb	10	2.67	16	6	0
	aaaaabbababaabababbbb	10	2.67	16	6	0
	aaaaabbababbbbbaababa	10	2.11	19	9	0
	aaaaabbabbbbbababbaba	10	3.0	15	5	0
	aaaaabbbbabbbaabbabba	10	2.43	17	7	0
	aaaaabbbbabbbabaababb	10	2.25	18	8	0
	aaaabaabababbbaaabbab	10	2.25	18	8	0
	aaaababaababbbbbaabaa	10	3.0	15	5	0
	aaaababbababaabaabbbb	10	2.25	18	8	0
	aaaabbaaabbbbabaabbba	10	2.43	17	7	0
	aaaabbaabbbbabaababbb	10	2.25	18	8	0
	aaaabbaabbbbababbbaab	10	2.25	18	8	0
	aaaabbabaaabbbbababbb	10	2.25	18	8	0
	aaaabbabaababababbbbb	10	2.67	16	6	0
	aaaabbabaabababbbbbbb	10	2.67	16	6	0
	aaaabbababaabababbbbb	10	2.67	16	6	0
	aaaabbabababbbbaababa	10	2.11	19	9	0
	aaaabbababbbbaabababa	10	2.11	19	9	0
	aaaabbababbbbbbaababa	10	2.11	19	9	0
	aaaabbabbabaabababbbb	10	2.25	18	8	0
	aaaabbabbbababbaababa	10	2.11	19	9	0
	aaaabbabbbbababbaabab	10	2.0	20	10	0
	aaaabbabbbbbbababbaba	10	3.0	15	5	0
	aaaabbbaabababbbabbab	10	2.25	18	8	0
	aaaabbbabbabaabababbb	10	2.25	18	8	0
	aaaabbbabbabaababbbbb	10	2.25	18	8	0
	aaaabbbbabbabaababbba	10	2.11	19	9	0
	aaaabbbbabbbaabbabbaa	10	2.43	17	7	0
	aaabababbbbaabbbabbba	10	3.0	15	5	0
	aaababbaababaaaaabbbb	10	2.25	18	8	0
	aaabbabbbbababbaababa	10	2.11	19	9	0
	aaabbabbbbbbbababbaba	10	3.0	15	5	0
	aabaabbbaaabaaabbabba	10	2.43	17	7	0
	aababaaaabaaabbaabbbb	10	2.25	18	8	0
	aababbaaaababaaabbbbb	10	2.25	18	8	0
	aabbabaaaaabbbbababbb	10	2.25	18	8	0
	aabbabaaaabbbbababbba	10	2.11	19	9	0
	aabbabaaaabbbbbababbb	10	2.25	18	8	0
	abaababbaaababaaabbbb	10	2.25	18	8	0
	abaabbbaaabaabbaaabbb	10	2.67	16	6	0
	abaabbbbabbbaaabaaabb	10	2.25	18	8	0
	ababbaaababaabbaabbbb	10	2.25	18	8	0
	ababbaabaaaababaabbbb	10	2.25	18	8	0
	ababbaabbaaaabaaabbbb	10	2.25	18	8	0
	ababbaabbaababaaaabbb	10	2.25	18	8	0
	ababbaabbaaababaaabbb	10	2.25	18	8	0
	ababbaabbaabaabaabbbb	10	2.67	16	6	0
	ababbaabbaababaaabbbb	10	2.25	18	8	0

(*continued on next page*)

Table A.16 (*continued*)

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
21	abbabaaabbbabaaabaabb	10	2.25	18	8	0
	abbabaaabbbbbabaaaabb	10	2.25	18	8	0
	abbabaabaabbabaaabbbb	10	2.67	16	6	0
	abbabaabaabbbbbabaaabb	10	2.25	18	8	0
	baabaabbaaaabaabaaabbbb	10	2.11	19	9	0
	baababbaaabaabaaabbbbb	10	2.43	17	7	0
	babaaaaabaaaabbaabbbbb	10	2.43	17	7	0
	babaaaaababaabaaaabbb	10	3.0	15	5	0
	babaaaaabbaabaaaabbbb	10	2.43	17	7	0
	babaaaabaaabbaabbbbbb	10	2.11	19	9	0
	babaaaabaaabbaabbbbbb	10	2.43	17	7	0
	babaaaababbaabaaabbbb	10	2.43	17	7	0
	babaaababaabaabbaabbbb	10	2.43	17	7	0
	babaaabbaabaaaabbbbb	10	2.43	17	7	0
	babaabaaaaaaabbaabbbb	10	2.43	17	7	0
	babaabaaaaaabbaabbbbb	10	2.43	17	7	0
	babaabaaaaabbaabbbbb	10	2.11	19	9	0
	babaabaaaaabbaabbbbbb	10	2.43	17	7	0
	babaabaaaabaaabbaabb	10	2.43	17	7	0
	babaabaaaababaabaabbb	10	3.0	15	5	0
	babaabaaaabbaabaabbbb	10	2.11	19	9	0
	babaabaaaabbaabaabbbb	10	2.43	17	7	0
	babaabaaaabbaabbbbbbb	10	2.43	17	7	0
	babaabaaabaaabbaabbbb	10	2.43	17	7	0
	babaabaabaaaabbaabbbb	10	2.43	17	7	0
	babaabaabaaaababbaabbbb	10	2.43	17	7	0
	babaababaabababbaabbbb	10	3.0	15	5	0
	babaababbaaabaabaabbb	10	2.43	17	7	0
	babaababbaabaaaaabbbb	10	2.43	17	7	0
	babaababbaabaaaabbbbb	10	2.43	17	7	0
	babaababbaababaaabbbb	10	2.43	17	7	0
	babaabbaaaabaabaaabbbb	10	2.11	19	9	0
	babaabbaaababaaabbbbb	10	2.43	17	7	0
	babaabbaabaabababbbbb	10	3.0	15	5	0
	bababbaaaababaaabbbbb	10	2.43	17	7	0
	bababbaababaaaaabbbb	10	2.43	17	7	0
	bababbaababaaaaabbbbb	10	2.43	17	7	0
	bababbaababaaaabbbbbb	10	2.43	17	7	0
	bababbaabaabaaaabbbbb	10	2.43	17	7	0
	bababbaababababaaabbbb	10	2.43	17	7	0
	bababbbaabababaaabbbb	10	2.43	17	7	0
	bababbbbaaaababaaabbb	10	2.43	17	7	0
	bababbbbaababaaaabbb	10	2.43	17	7	0

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
21	babbaaaababaaabbbaabb	10	2.11	19	9	0
	babbaabaaaaabababbbbb	10	2.43	17	7	0
	babbaabaaaababaabbbbb	10	2.43	17	7	0
	babbaabaaaabbbaabbbbb	10	2.43	17	7	0
	babbaabbaaaababaaabbb	10	2.43	17	7	0
	babbaabbaaababaaabbbb	10	2.43	17	7	0
	babbaabbaabababaaabbbb	10	2.43	17	7	0
	babbabaabaaaabbaabbbb	10	2.11	19	9	0
	babbbaabaaaabababbbb	10	2.43	17	7	0
	babbbaabaababbaabbbbb	10	3.0	15	5	0
	bbaabaaaababbabaabbbb	10	2.11	19	9	0
	bbabaaaabbababaabbbb	10	3.0	15	5	0
	bbabaaabaabbabaabbbb	10	2.43	17	7	0
	bbabaaabbabaaabaabbbb	10	3.0	15	5	0
	bbabaabaabbabaaaabbbb	10	2.43	17	7	0
	bbabaabaabbabaaabbbbb	10	3.0	15	5	0
	bbababbabaabbabaabbbb	10	3.0	15	5	0
	bbabbabaaaabbabaaabbb	10	3.0	15	5	0
	bbabbabaaabbabaabbbbb	10	3.0	15	5	0
	bbabbabaabbabaaaabbbb	10	2.43	17	7	0
22	aaaabbabbbbababbaababa	12	2.33	21	9	0
	ababbaabbaaaabababbb	12	2.5	20	8	0
	ababbaabbaaababaaabbbb	12	2.5	20	8	0
	ababbaabbaababaaaabbbb	12	2.5	20	8	0
	abbabaaabbbbabaaabaabb	12	2.5	20	8	0
	babaabaaaabaaabbaabbbb	12	2.71	19	7	0
	babaababbaaababaaabbbb	12	2.71	19	7	0
	babbaabbaaaababaaabbbb	12	2.71	19	7	0
	bbabbabaaaabbabaaabbbb	12	3.4	17	5	0
23	ababbaabbaaaabababbbbb	14	2.75	22	8	0
24	ababbaabbaaaababaaabbbbb	14	2.75	22	8	0
	babababbaabbaaaababaaabbbb	14	2.56	23	9	0
25	ababbaabbaaaaababaaaabbbb	14	2.75	22	8	0
	ababbaabbaaaababaaaabbbbbb	14	2.75	22	8	0
	ababbaabbaaaababababaabbbb	14	2.75	22	8	0
	ababbaabbaabaaaababaabbbb	14	2.75	22	8	0
	ababbbaabbaaaababaaabbbbb	14	2.4	24	10	0
	babaababbaabaaaababaabbbb	14	3.0	21	7	0
	babababaabaaaababaabbbbb	14	3.8	19	5	0
	babababaabbaaaababaaabbbbb	14	2.56	23	9	0

Table A.17
All extremal forward strings s on $\Sigma = \{a, b\}$ with non-positive values of δ_B and $n = 3, \ldots, 25$. Rotations of standard words are marked with a 1 in the last column.

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	rot.s of st. words
3	aab	0	1.0	2	2	1
	abb	0	1.0	2	2	1
4	abab	-2	2.0	2	4	0
5	aabab	-2	2.0	2	4	1
	ababb	-2	2.0	2	4	1
6	aabaab	-2	2.0	2	4	0
	ababab	-2	2.0	2	4	0
	ababba	-2	1.67	3	5	0
	abbabb	-2	2.0	2	4	0
	baabab	-2	1.67	3	5	0
7	aaabaab	-2	2.0	2	4	1
	aabaaba	-2	1.67	3	5	1
	aababab	-2	2.0	2	4	1
	aabbaab	-2	1.5	4	6	0
	abaabab	-2	1.5	4	6	1
	abaabba	-2	1.4	5	7	0
	abababb	-2	2.0	2	4	1
	ababbab	-2	1.5	4	6	1
	abbaabb	-2	1.5	4	6	0
	abbabbb	-2	2.0	2	4	1
	baabbab	-2	1.4	5	7	0
	babbabb	-2	1.67	3	5	1
8	aabaabab	-4	3.0	2	6	1
	abaabbab	-4	2.0	4	8	0
	ababbabb	-4	3.0	2	6	1
9	aabaabbab	-4	2.0	4	8	0
10	aabaabbbab	-4	2.0	4	8	0
	aababaabab	-4	3.0	2	6	0
	abaabbabab	-4	2.0	4	8	0
	ababaabbab	-4	2.0	4	8	0
	ababbababb	-4	3.0	2	6	0
	abbaabbbab	-4	2.0	4	8	0
	baabbbabab	-4	1.8	5	9	0
11	aaabaaabaab	-4	3.0	2	6	1
	aaabbbaaaba	-4	1.8	5	9	0
	aabaaabbaab	-4	2.0	4	8	0
	aabaabaabab	-4	3.0	2	6	1
	aabaabbbaba	-4	1.8	5	9	0
	aabaabbbbab	-4	2.0	4	8	0
	aababaaabab	-4	2.0	4	8	0
	aababaabbab	-4	2.0	4	8	0
	aababbaaabb	-4	2.0	4	8	0
	aabbbaabaab	-4	2.0	4	8	0
	abaabbbabab	-4	1.67	6	10	0
	ababaaabbab	-4	1.67	6	10	0
	ababbabaaab	-4	1.67	6	10	0
	ababbababba	-4	2.33	3	7	0
	ababbabbabb	-4	3.0	2	6	1
	ababbbababb	-4	2.0	4	8	0
	abbabbaaabb	-4	2.0	4	8	0
	abbabbbaaab	-4	1.67	6	10	0
	abbabbbabbb	-4	3.0	2	6	1
	baababaabab	-4	2.33	3	7	0
	baabbbbabab	-4	1.8	5	9	0
	babbaabbbab	-4	1.8	5	9	0
12	aaaabbbaaaba	-4	1.8	5	9	0
	aaabaaabaaba	-4	2.33	3	7	0
	aaabaaabbaab	-4	2.0	4	8	0
	aaababaaabab	-4	2.0	4	8	0
	aaabbabbbaaa	-4	1.8	5	9	0
12	aaabbbaabaab	-4	2.0	4	8	0
	aaabbbbaaaba	-4	1.8	5	9	0
	aabaaabaabab	-4	2.0	4	8	0
	aabaaababaab	-4	2.0	4	8	0
	aabaaababbaa	-4	1.8	5	9	0
	aabaaabbaaba	-4	1.8	5	9	0
	aabaabaabbab	-4	2.0	4	8	0
	aabaabbaabab	-4	2.0	4	8	0
	aabaabbbabaa	-4	1.8	5	9	0
	aabaabbbabab	-4	1.67	6	10	0
	aabaabbbaba	-4	1.8	5	9	0
	aabaabbbbbab	-4	2.0	4	8	0
	aababaaaabab	-4	2.0	4	8	0
	aababaaabbab	-4	1.67	6	10	0
	aababaababab	-4	3.0	2	6	1
	aababbaaabba	-4	1.8	5	9	0
	aabbaaabbbaa	-4	1.8	5	9	0
	aabbaabaabab	-4	2.0	4	8	0
	aabbaabbbaab	-4	2.0	4	8	0
	aabbabaabbab	-4	2.0	4	8	0
	aabbbaabaaab	-4	1.67	6	10	0
	aabbbaabaaba	-4	1.8	5	9	0
	aabbbabaabab	-4	2.0	4	8	0
	aabbbbaabaab	-4	2.0	4	8	0
	abaaabbabaab	-4	1.67	6	10	0
	abaababaabab	-4	2.0	4	8	1
	abaababbaabb	-4	2.0	4	8	0
	abaabbaabbab	-4	2.0	4	8	0
	abaabbababab	-4	2.0	4	8	0
	abaabbbbabab	-4	1.67	6	10	0
	ababaaaabbab	-4	1.67	6	10	0
	ababaabbabab	-4	2.0	4	8	0
	abababaabbab	-4	2.0	4	8	0
	abababbababb	-4	3.0	2	6	1
	ababbaaabab	-4	2.0	4	8	0
	ababbaababba	-4	1.8	5	9	0
	ababbaabbabb	-4	2.0	4	8	0
	ababbaabbbab	-4	1.67	6	10	0
	ababbabaaaab	-4	1.67	6	10	0
	ababbabaabab	-4	2.0	4	8	1
	ababbababbba	-4	1.8	5	9	0
	ababbbabaaab	-4	1.5	8	12	0
	ababbbababbb	-4	2.0	4	8	0
	ababbbbababb	-4	2.0	4	8	0
	abbaaabababba	-4	1.57	7	11	0
	abbaaabbaabb	-4	2.0	4	8	0
	abbaaabbbaab	-4	1.67	6	10	0
	abbaabbbabab	-4	2.0	4	8	0
	abbabaabbbab	-4	1.67	6	10	0
	abbababbbabb	-4	2.0	4	8	0
	abbabbaaaabb	-4	2.0	4	8	0
	abbabbaabaab	-4	1.67	6	10	0
	abbabbbaaaab	-4	1.67	6	10	0
	abbabbbaaabb	-4	1.67	6	10	0
	abbabbbaabab	-4	1.67	6	10	0
	abbabbbaabbb	-4	2.0	4	8	0
	abbbaaaababbb	-4	1.67	6	10	0
	abbbaabaabab	-4	1.67	6	10	0
	abbbaabbbbab	-4	2.0	4	8	0
	baaababaabab	-4	1.8	5	9	0
	baaabbbabaab	-4	1.57	7	11	0
	baababaabbab	-4	1.8	5	9	0
	baabbabaabab	-4	1.8	5	9	0
	baabbbabbbab	-4	1.8	5	9	0
	baabbbababab	-4	1.8	5	9	0

(*continued on next page*)

Table A.17 (*continued*)

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{rev})$	rot.s of st. words
12	baabbbbbabab	-4	1.8	5	9	0
	babbaaabbbab	-4	1.57	7	11	0
	babbabbbabbb	-4	2.33	3	7	0
	bbaaababbabb	-4	1.8	5	9	0
	bbaabaaababb	-4	1.8	5	9	0
	bbabbbaaabbb	-4	1.8	5	9	0
13	aabaababaabab	-6	4.0	2	8	1
	aabbaaabbbaab	-6	2.5	4	10	0
	aabbabaabbbab	-6	2.5	4	10	0
	ababbaaababba	-6	2.2	5	11	0
	ababbababbabb	-6	4.0	2	8	1
	abbabbbaabaab	-6	2.0	6	12	0
	baabbbabaabab	-6	2.2	5	11	0
14	aaabaaabbbaaba	-6	2.2	5	11	0
	aabaaabbbabaab	-6	2.0	6	12	0
	aabaabbabaabab	-6	2.5	4	10	0
	aababaabbbabab	-6	2.5	4	10	0
	aabbabbbaaabbb	-6	2.5	4	10	0
	aabbabbbaabaab	-6	2.0	6	12	0
	aabbbaabaabbab	-6	2.5	4	10	0
	abaabbaababbab	-6	2.0	6	12	0
	abaabbaababbba	-6	1.86	7	13	0
	abaabbabaabbab	-6	2.5	4	10	0
	ababaabbaabbab	-6	2.5	4	10	0
	ababbaaaababba	-6	2.2	5	11	0
	ababbaababbabb	-6	2.5	4	10	0
	ababbababbbabb	-6	2.5	4	10	0
	abbaaababbbabb	-6	2.0	6	12	0
	abbabbaabaaabb	-6	2.0	6	12	0
	abbabbbaaabaab	-6	2.0	6	12	0
	abbabbbaabaaab	-6	1.75	8	14	0
	abbbaabbbbabab	-6	2.5	4	10	0
	baaabbbabaabab	-6	1.86	7	13	0
	baabbabaabbbab	-6	2.2	5	11	0
	baabbbbabaabab	-6	2.2	5	11	0
15	aabaabbbabaabab	-8	3.0	4	12	0
16	aabaabbbbabaabab	-8	3.0	4	12	0
	aababbaaababbabb	-8	3.0	4	12	0
17	aaababbaaababbabb	-8	3.0	4	12	0
	aaabbaabaaabbbaab	-8	3.0	4	12	0
	aabaabbbbbabaabab	-8	3.0	4	12	0
	aababbaaababbbabb	-8	2.33	6	14	0
	aabbabbbaabaaabbb	-8	2.33	6	14	0
	abbbaabaabbbbabab	-8	2.33	6	14	0
18	aaabaabbbabaaaabab	-8	2.33	6	14	0
	aaababbaaaababbabb	-8	3.0	4	12	0
	aaababbaaababbbabb	-8	2.33	6	14	0
	aaabbaabaaabbbaaba	-8	2.6	5	13	0
	aaabbabbbaabaaabbb	-8	2.33	6	14	0
	aaabbbabaaabbbbaab	-8	2.33	6	14	0
	aabaabbabbbabaabab	-8	2.33	6	14	0
	aabaabbbabaabaabab	-8	3.0	4	12	0
	aabaabbbbbbabaabab	-8	3.0	4	12	0
	aababaabbabaaaabab	-8	2.33	6	14	0
	aababbaaababbbbabb	-8	2.33	6	14	0
	aabbabaabbbaabbbab	-8	3.0	4	12	0
	aabbbaaabbbbaabaab	-8	3.0	4	12	0
	aabbbbaabbaabaabab	-8	2.33	6	14	0
	abaabbbbabaababbab	-8	2.0	8	16	0
	ababaabbaaaababbab	-8	2.0	8	16	0
	ababaabbaaaababbba	-8	1.89	9	17	0
	ababaabbabaaaabbab	-8	2.33	6	14	0
	ababaabbbbabaabbab	-8	2.33	6	14	0
	ababbaaaabaababbab	-8	2.0	8	16	0
	ababbaaabaaaababba	-8	2.14	7	15	0
18	ababbaabbbbabbabab	-8	2.33	6	14	0
	ababbababbaaababba	-8	2.6	5	13	0
	ababbabbaaababbabb	-8	3.0	4	12	0
	ababbbababbaaababb	-8	2.33	6	14	0
	ababbbbababbaababb	-8	2.33	6	14	0
	abbabbaaabbbaabaab	-8	2.33	6	14	0
	abbabbbaaabaaabaab	-8	2.33	6	14	0
	abbbaabaaabbbbabab	-8	2.0	8	16	0
	baabbbabaababaabab	-8	2.6	5	13	0
	baabbbabbbbabaabab	-8	2.14	7	15	0
19	ababbbbababbaaababb	-10	2.67	6	16	0
20	aaabbbaabaaabbbbaaba	-10	3.0	5	15	0
	aabaabbbabaababaabab	-10	3.5	4	14	0
	aabaabbbabbbbabaabab	-10	2.67	6	16	0
	aabbbbaabbaaaabaabab	-10	2.25	8	18	0
	ababaabbbbabaaaabbab	-10	2.25	8	18	0
	ababbbbababbaaaababb	-10	2.67	6	16	0
	ababbbbbababbaaababb	-10	2.67	6	16	0
	abbabbaabbbabaabbbab	-10	3.5	4	14	0
	baabbbabaababaaaabab	-10	2.43	7	17	0
21	aaabbaabaaabbabbbaaba	-10	2.43	7	17	0
	aabaababaaabbbabaabab	-10	2.67	6	16	0
	aabaabbbabaababaaabab	-10	2.67	6	16	0
	aabaabbbbabaababaabab	-10	3.5	4	14	0
	aababaaabbbabaabbabab	-10	2.67	6	16	0
	aababbaaababbababbabb	-10	3.5	4	14	0
	aabbabaaaabaabbbaabab	-10	2.25	8	18	0
	aabbabaabaabbbaabbbab	-10	3.5	4	14	0
	aabbabbabaabbbaabbbab	-10	3.5	4	14	0
	aabbbaaaabaabbbbaabab	-10	2.25	8	18	0
	aabbbabaabaabbbbaabab	-10	2.67	6	16	0
	aabbbbaabbaaaaabaabab	-10	2.25	8	18	0
	aabbbbaabbaaabaaaabab	-10	2.25	8	18	0
	aabbbbbaabbaaaabaabab	-10	2.25	8	18	0
	abaaabbabaababbaabbab	-10	2.25	8	18	0
	abaabbabaaaabbaabbbab	-10	2.25	8	18	0
	abaabbabaababbaabbbab	-10	2.25	8	18	0
	ababaaaabbbbabaaabbab	-10	2.25	8	18	0
	ababaaaabbbbababbaaab	-10	2.0	10	20	0
	ababaabbababbaaaabbab	-10	2.25	8	18	0
	ababaabbbabaaaabbabab	-10	2.25	8	18	0
	ababaabbbbabaaaaabbab	-10	2.25	8	18	0
	ababaabbbbabaaaabbaba	-10	2.11	9	19	0
	ababaabbbbbabaaaabbab	-10	2.25	8	18	0
	ababbbbababbaaaaababb	-10	2.67	6	16	0
	ababbbbbababbaaaababb	-10	2.67	6	16	0
	ababbbbbbababbaaababb	-10	2.67	6	16	0
	abbaababbabbbaaababbb	-10	2.67	6	16	0
	abbabbaabbbabaaabbbab	-10	2.67	6	16	0
	abbabbaabbbabaabbbbab	-10	2.67	6	16	0
	abbbaaaababaabbbbabab	-10	2.25	8	18	0
	baaabbaababbbbabbaabb	-10	2.43	7	17	0
	baabbbabaababaaaaabab	-10	2.43	7	17	0
	baabbbabaababaabbabab	-10	3.0	5	15	0
22	aaaababaaaabbbabaaabab	-10	2.67	6	16	0
	aaabaabbabaababaaaabab	-10	2.67	6	16	0
	aaabaabbbabaababaaabab	-10	2.67	6	16	0
	aaababaabbbbabaaaabbab	-10	2.25	8	18	0
	aaababbaaabaaaababbabb	-10	2.67	6	16	0
	aaababbaaababbababbabb	-10	3.5	4	14	0
	aaabbaabbbaaabaaaabbab	-10	2.67	6	16	0
	aaabbbabaaabaaabbbbaab	-10	2.67	6	16	0
	aaabbbbaabaaabbbbbaaba	-10	3.0	5	15	0
	aabaababaaaabbbabaabab	-10	2.25	8	18	0
	aabaababaaabbbbabaabab	-10	2.67	6	16	0

Table A.17 (*continued*)

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\mathrm{rev}})$	rot.s of st. words
22	aabaabbabaaabbbaabbbab	-10	2.67	6	16	0
	aabaabbabaababbaabbbab	-10	2.25	8	18	0
	aabaabbbabaababaaaabab	-10	2.25	8	18	0
	aabaabbbbabaababaaabab	-10	2.67	6	16	0
	aabaabbbbabaababaabbab	-10	2.67	6	16	0
	aabaabbbbabbbbbabaabab	-10	2.67	6	16	0
	aabaabbbbbabaababaabab	-10	3.5	4	14	0
	aababaaabbbbabaabbabab	-10	2.25	8	18	0
	aabababbbbababbaaababb	-10	2.67	6	16	0
	aababbaaababbababbbabb	-10	2.67	6	16	0
	aababbaaabbaabaababbba	-10	2.43	7	17	0
	aababbbabbaaababbbbabb	-10	2.67	6	16	0
	aabbaaaabaabbabbbaabab	-10	2.0	10	20	0
	aabbaabaabbabbbaababbb	-10	2.67	6	16	0
	aabbaabbbaabaabaaabbab	-10	2.67	6	16	0
	aabbabaaaaabaabbbaabab	-10	2.25	8	18	0
	aabbabaaaabaabbbaabbab	-10	2.25	8	18	0
	aabbabaaabaabbbaabbbab	-10	2.67	6	16	0
	aabbabaabaabbbaabbbbab	-10	2.67	6	16	0
	aabbabbabaaabbbaabbbab	-10	2.67	6	16	0
	aabbabbabaabbbaabbbbab	-10	2.67	6	16	0
	aabbbaaaaabaabbbbaabab	-10	2.25	8	18	0
	aabbbaaabaaabbbbaabaab	-10	2.67	6	16	0
	aabbbaabbbbaaabaaaabab	-10	2.25	8	18	0
	aabbbabaaaababaabbabab	-10	2.67	6	16	0
	aabbbabaaabaabbbbaabab	-10	2.25	8	18	0
	aabbbbaaababaaabbaabab	-10	2.25	8	18	0
	aabbbbaabbaaaaaabaabab	-10	2.25	8	18	0
	aabbbbbaabbaaaaabaabab	-10	2.25	8	18	0
	aabbbbbaabbaaabaaaabab	-10	2.25	8	18	0
	aabbbbbbaabbaaaabaabab	-10	2.25	8	18	0
	abaaaabbabaaabbaabbbab	-10	2.25	8	18	0
	abaaabbabaababbaabbbab	-10	2.0	10	20	0
	abaababab baabbabaabbab	-10	2.67	6	16	0
	abaabbabaaaaabbaabbbab	-10	2.25	8	18	0
	abaabbabaaaababbaabbab	-10	2.0	10	20	0
	abaabbabaaababbaabbbab	-10	2.0	10	20	0
	abaabbabaababbbaabbbab	-10	2.25	8	18	0
	ababaaaabbababbaaabbab	-10	2.25	8	18	0
	ababaaaabbbabaaabbabab	-10	2.25	8	18	0
	ababaaaabbbbabaaabbaba	-10	2.11	9	19	0
	ababaaaabbbbbabaaabbab	-10	2.25	8	18	0
	ababaaaabbbbbababbaaab	-10	2.0	10	20	0
	ababaabbababbaaaaabbab	-10	2.25	8	18	0
	ababaabbbabaaaaabbabab	-10	2.25	8	18	0
	ababaabbbababbaabbbbab	-10	2.25	8	18	0
	ababaabbbbabaaaaaabbab	-10	2.25	8	18	0
	ababaabbbbabaaaaabbaba	-10	2.11	9	19	0
	ababaabbbbabaaaababbab	-10	2.0	10	20	0
	ababaabbbbabaaaabbabab	-10	2.0	10	20	0
	ababaabbbbbabaaaaabbab	-10	2.25	8	18	0
	ababaabbbbbabaaaabbaba	-10	2.11	9	19	0
	ababaabbbbbbabaaaabbab	-10	2.25	8	18	0
	abababaabbbbabaaaabbab	-10	2.25	8	18	0
	abababbaaaabbabaabbbab	-10	2.0	10	20	0
	abababbaaabaaabbbabaab	-10	2.25	8	18	0
	ababbaaabaaaabaababbab	-10	2.25	8	18	0
	ababbaabaabbabbabaabab	-10	2.67	6	16	0
	ababbababbbbabbaaabaab	-10	2.25	8	18	0
	ababbababbbbabbaaababb	-10	2.25	8	18	0
	ababbbaaabababbbbababb	-10	2.67	6	16	0
	ababbbbababbaaaaaababb	-10	2.67	6	16	0
	ababbbbababbaaaabaabab	-10	2.0	10	20	0
	ababbbbababbaabaaababb	-10	2.25	8	18	0
	ababbbbbababbaaaaababb	-10	2.67	6	16	0
22	ababbbbbbababbaaaababb	-10	2.67	6	16	0
	ababbbbbbbababbaaababb	-10	2.67	6	16	0
	abbaaaababbbaaababbabb	-10	2.25	8	18	0
	abbaaababbabbbaaababbb	-10	2.67	6	16	0
	abbaabaaababbbaaababba	-10	2.11	9	19	0
	abbaababbabbbaaaababbb	-10	2.25	8	18	0
	abbaabbbaaabaaaababbab	-10	2.0	10	20	0
	abbaabbbaaababaaaabbab	-10	2.0	10	20	0
	abbabbaaabaaabbbaabaab	-10	2.25	8	18	0
	abbabbaaabbbabbbaabaab	-10	2.25	8	18	0
	baabbbbabaababaaaabab	-10	2.43	7	17	0
	bbaaaababbbaaababbabb	-10	2.43	7	17	0
	abbabbaabbbabaaaabbbab	-10	2.67	6	16	0
	abbabbaabbbabaaabbbbab	-10	2.25	8	18	0
	abbabbaabbbababaabbbab	-10	2.67	6	16	0
	abbabbababaabbbaabbbab	-10	2.67	6	16	0
	abbbaaaaababaabbbbabab	-10	2.25	8	18	0
	abbbaaabaaabbbbababaab	-10	2.67	6	16	0
	abbbaaababaaaabbbbabab	-10	2.25	8	18	0
	abbbaababbbbaaabbaaabb	-10	2.67	6	16	0
	abbbabaaaabaabbbbaabab	-10	2.0	10	20	0
	abbbabbbbaaababbaaaabb	-10	2.25	8	18	0
	abbbabbbbaaababbaababb	-10	2.25	8	18	0
	baaaababaaabaabbaabbab	-10	2.43	7	17	0
	baaabbaababbbbbabbaabb	-10	2.43	7	17	0
	baababaaabbbabaabbabab	-10	2.43	7	17	0
	baabbabaaaababbaabbbab	-10	1.91	11	21	0
	baabbabaabaabbbaabbbab	-10	3.0	5	15	0
	baabbabbabaabbbaabbbab	-10	3.0	5	15	0
	baabbbabaababaaaaaabab	-10	2.43	7	17	0
	baabbbabaababaaabbabab	-10	2.43	7	17	0
	baabbbbabaababaaaaabab	-10	2.43	7	17	0
	baabbbbabaababaabbabab	-10	2.43	7	17	0
	baabbbbbabaababaaaabab	-10	2.43	7	17	0
	bbaaaabaabbabbbaababbb	-10	2.11	9	19	0
	bbaaaababbbbaaababbabb	-10	2.11	9	19	0
	bbaabaaababbabbaaababb	-10	2.43	7	17	0
23	aaabaabbbabaababaaaabab	-12	3.0	6	18	0
	aabbabaaaabaabbbaabbbab	-12	3.0	6	18	0
	aabbbabaaaabaabbbbaabab	-12	2.5	8	20	0
	abaabbabaaaababbaabbbab	-12	2.2	10	22	0
24	aaaabaabbbabaababaaaabab	-12	3.0	6	18	0
	aaabaabbbbabaababaaaabab	-12	3.0	6	18	0
	aaababaaaabbbabaaabbabab	-12	3.0	6	18	0
	aaabbaabbbaaababaaaabbab	-12	2.5	8	20	0
	aabaabbabaabaabbbaabbbab	-12	4.0	4	16	0
	aabbaaaabaabbabbbaababbb	-12	2.5	8	20	0
	aabbabaaaaabaabbbaabbbab	-12	3.0	6	18	0
	aabbabaaaabaabbbaabbbbab	-12	2.5	8	20	0
	aabbbaaababaaaabbaabbaba	-12	2.33	9	21	0
	aabbbabaaaaabaabbbbaabab	-12	2.5	8	20	0
	aabbbbaabbaaabaaaabaabab	-12	2.5	8	20	0
	abaaaabbabaaababbaabbbab	-12	2.2	10	22	0
	abaabbabaaaaababbaabbbab	-12	2.2	10	22	0
	abaabbabaaaababbbaabbbab	-12	2.2	10	22	0
	abababbaaabaaabbbababaab	-12	2.5	8	20	0
	ababbbbababbaaabaaaababb	-12	2.5	8	20	0
	abbaabbbaaababaaaababbab	-12	2.2	10	22	0
	abbaabbbaaababaaaabbabab	-12	2.2	10	22	0
	abbbababbbbaaababbaababb	-12	2.5	8	20	0
	baaabbabaaabaabbbaabbbab	-12	2.71	7	19	0
	baabbabaaaabaabbbaabbbab	-12	2.71	7	19	0
25	baaaabbabaaabaabbbaabbbab	-14	3.0	7	21	0

A.3. Number of Lyndon factors

Table A.18
Statistics of the differences in the number of Lyndon factors for all forward string s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 3, \ldots, 25$: minimum and maximum values, mean and standard deviation.

	$\ell(s) - \ell(s^{\text{rev}})$			
n	min	max	mean	sd
3	-2	-2	-2	0
4	-3	-1	-2	1.095
5	-4	-1	-2.5	1.314
6	-5	-1	-2.5	1.528
7	-6	0	-2.679	1.8
8	-7	0	-2.7	1.904
9	-8	1	-2.767	2.075
10	-9	1	-2.766	2.161
11	-10	2	-2.804	2.263
12	-11	2	-2.805	2.321
13	-12	3	-2.825	2.39
14	-13	3	-2.829	2.435
15	-14	4	-2.843	2.482
16	-15	4	-2.847	2.52
17	-16	5	-2.859	2.555
18	-17	5	-2.864	2.586
19	-18	6	-2.874	2.616
20	-19	6	-2.879	2.643
21	-20	7	-2.887	2.668
22	-21	7	-2.892	2.691
23	-22	8	-2.898	2.713
24	-23	8	-2.903	2.734
25	-24	9	-2.908	2.754

Table A.19
Statistics of the differences in the number of distinct Lyndon factors for all forward string s on $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $n = 3, \ldots, 25$: minimum and maximum values, mean and standard deviation.

	$d(s) - d(s^{\text{rev}})$			
n	min	max	mean	sd
3	-1	-1	-1	0
4	-2	0	-0.833	0.753
5	-2	0	-0.833	0.718
6	-2	0	-0.893	0.786
7	-2	1	-0.929	0.783
8	-3	1	-0.933	0.867
9	-3	1	-0.983	0.901
10	-3	2	-1.002	0.957
11	-4	2	-1.032	1.001
12	-4	2	-1.045	1.058
13	-4	2	-1.072	1.099
14	-4	3	-1.084	1.147
15	-5	3	-1.104	1.188
16	-5	3	-1.115	1.231
17	-5	3	-1.13	1.269
18	-5	4	-1.139	1.306
19	-6	4	-1.151	1.341
20	-6	4	-1.159	1.374
21	-6	4	-1.168	1.406
22	-6	5	-1.175	1.436
23	-6	5	-1.183	1.464
24	-7	5	-1.189	1.491
25	-7	5	-1.195	1.517

A.4. Ternary alphabet

Table A.20
All extremal forward strings s on $\Sigma = \{a, b, c\}$ for ρ_B and $n = 3, \ldots, 15$. Perfectly clustering strings are marked with a 1 in the last column.

n	s	$\rho_B(s)$	$\delta_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	perfectly clustering
3	aab	1.0	0	2	2	1
	aac	1.0	0	2	2	1
	abb	1.0	0	2	2	1
	abc	1.0	0	3	3	1
	acb	1.0	0	3	3	1
	acc	1.0	0	2	2	1
	bac	1.0	0	3	3	1
	bbc	1.0	0	2	2	1
	bcc	1.0	0	2	2	1
4	aabb	2.0	2	4	2	0
	aacc	2.0	2	4	2	0
	abab	2.0	-2	2	4	1
	acac	2.0	-2	2	4	1
	bbcc	2.0	2	4	2	0
	bcbc	2.0	-2	2	4	1
5	aaabb	2.0	2	4	2	0
	aaacc	2.0	2	4	2	0
	aabab	2.0	-2	2	4	1
	aabbb	2.0	2	4	2	0
	aacac	2.0	-2	2	4	1
	aaccc	2.0	2	4	2	0
	ababb	2.0	-2	2	4	1
	acacc	2.0	-2	2	4	1
	bbbcc	2.0	2	4	2	0
	bbcbc	2.0	-2	2	4	1
	bbccc	2.0	2	4	2	0
	bcbcc	2.0	-2	2	4	1
6	aaaabb	2.0	2	4	2	0
	aaaacc	2.0	2	4	2	0
	aaabbb	2.0	2	4	2	0
	aaaccc	2.0	2	4	2	0
	aabaab	2.0	-2	2	4	1
	aabbbb	2.0	2	4	2	0
	aacaac	2.0	-2	2	4	1
	aacccc	2.0	2	4	2	0
	ababab	2.0	-2	2	4	1
	abbabb	2.0	-2	2	4	1
	abcaac	2.0	-3	3	6	1
	abcabc	2.0	-3	3	6	1
	acabab	2.0	-3	3	6	1
	acacac	2.0	-2	2	4	1
	accacc	2.0	-2	2	4	1
	babacc	2.0	3	6	3	0
	bbbbcc	2.0	2	4	2	0
	bbbccc	2.0	2	4	2	0
	bbcbbc	2.0	-2	2	4	1
	bbcccc	2.0	2	4	2	0
	bcbcbc	2.0	-2	2	4	1
	bccbcc	2.0	-2	2	4	1
7	abacacb	2.33	-4	3	7	1
	acbaccb	2.33	-4	3	7	1
8	aabaabab	3.0	-4	2	6	1
	aacaacac	3.0	-4	2	6	1
	ababbabb	3.0	-4	2	6	1
	acaccacc	3.0	-4	2	6	1
	bbcbbcbc	3.0	-4	2	6	1
	bcbccbcc	3.0	-4	2	6	1
9	aabbcaabc	2.67	-5	3	8	1
	abacabbac	2.67	-5	3	8	1
	acbbcacbc	2.67	-5	3	8	1
10	aaabbbabba	3.0	6	9	3	0
	aaacccacca	3.0	6	9	3	0
	aababaabab	3.0	-4	2	6	1
	aacacaacac	3.0	-4	2	6	1
	ababbababb	3.0	-4	2	6	1
	abcbcabcbc	3.0	-6	3	9	1
	acaccacacc	3.0	-4	2	6	1
	bbbcccbccb	3.0	6	9	3	0
	bbcbcbbcbc	3.0	-4	2	6	1
	bcbccbcbcc	3.0	-4	2	6	1
11	abacbabbabb	3.33	-7	3	10	1
	abcabcbbcbc	3.33	-7	3	10	1
12	aaaaabbbabba	3.0	6	9	3	0
	aaaaacccacca	3.0	6	9	3	0
	aaabbbbabbba	3.0	6	9	3	0
	aaaccccaccca	3.0	6	9	3	0
	aababaababab	3.0	-4	2	6	1
	aacacaacacac	3.0	-4	2	6	1
	abababbababb	3.0	-4	2	6	1
	abacacbabacb	3.0	-6	3	9	1
	abacacbacacb	3.0	-6	3	9	1
	abcabcaacaac	3.0	-6	3	9	1
	abcacabcbcac	3.0	-6	3	9	1
	abcbcabcbcbc	3.0	-6	3	9	1
	acacaccacacc	3.0	-4	2	6	1
	babaabaabbbb	3.0	6	9	3	0
	bbbbbcccbccb	3.0	6	9	3	0
	bbbccccbcccb	3.0	6	9	3	0
	bbcbbcabcabc	3.0	-6	3	9	1
	bbcbcbbcbcbc	3.0	-4	2	6	1
	bcbcbccbcbcc	3.0	-4	2	6	1
	cacaacaaccccc	3.0	6	9	3	0
	cbcbbcbbcccc	3.0	6	9	3	0
13	aabaababaabab	4.0	-6	2	8	1
	aacaacacaacac	4.0	-6	2	8	1
	ababbababbabb	4.0	-6	2	8	1
	acaccacaccacc	4.0	-6	2	8	1
	bbcbbcbcbbcbc	4.0	-6	2	8	1
	bcbccbcbccbcc	4.0	-6	2	8	1
14	abacacbabacacb	3.67	-8	3	11	1
15	abbcabbcbbbcbbc	3.67	-8	3	11	1
	acaccacacbacacb	3.67	-8	3	11	1

Table A.21
All extremal forward strings s on $\Sigma = \{a, b, c\}$ with non-negative values of δ_B and $n = 3, \ldots, 15$. Perfectly clustering strings are marked with a 1 in the last column.

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	perfectly clustering
3	aab	0	1.0	2	2	1
	aac	0	1.0	2	2	1
	abb	0	1.0	2	2	1
	abc	0	1.0	3	3	0
	acb	0	1.0	3	3	0
	acc	0	1.0	2	2	1
	bac	0	1.0	3	3	0
	bbc	0	1.0	2	2	1
	bcc	0	1.0	2	2	1
4	aabb	2	2.0	4	2	0
	aacc	2	2.0	4	2	0
	bbcc	2	2.0	4	2	0
5	aaabb	2	2.0	4	2	0
	aaacc	2	2.0	4	2	0
	aabba	2	1.67	5	3	0
	aabbb	2	2.0	4	2	0
	aacbb	2	1.67	5	3	0
	aacca	2	1.67	5	3	0
	aaccc	2	2.0	4	2	0
	abbcc	2	1.67	5	3	0
	baabb	2	1.67	5	3	0
	baacc	2	1.67	5	3	0
	babac	2	1.67	5	3	0
	bbbcc	2	2.0	4	2	0
	bbccb	2	1.67	5	3	0
	bbccc	2	2.0	4	2	0
	caacc	2	1.67	5	3	0
	cbbcc	2	1.67	5	3	0
6	babacc	3	2.0	6	3	0
7	aabbccb	3	1.75	7	4	0
	aacbbcc	3	1.75	7	4	0
	aacbcbb	3	2.0	6	3	0
	aaccbca	3	1.75	7	4	0
	aaccbcb	3	2.0	6	3	0
	baabbcc	3	1.75	7	4	0
	baacbcb	3	1.75	7	4	0
	babaacc	3	2.0	6	3	0
	bababcc	3	1.75	7	4	0
	babaccc	3	2.0	6	3	0
	baccbcb	3	1.75	7	4	0
	bbabacc	3	2.0	6	3	0
	caabacc	3	1.75	7	4	0
8	aaabbaba	4	2.33	7	3	0
	aaaccaca	4	2.33	7	3	0
	aabcacca	4	2.0	8	4	0
	aacbbccc	4	2.0	8	4	0
	abaaabbb	4	2.0	8	4	0
	acaaaccc	4	2.0	8	4	0
	baacbcbb	4	2.0	8	4	0
	baaccbcb	4	2.0	8	4	0
	babaabbb	4	2.33	7	3	0
	bababbcc	4	2.0	8	4	0
	bbbccbcb	4	2.33	7	3	0
	bcbbbccc	4	2.0	8	4	0
	cacaaccc	4	2.33	7	3	0
	cbacbbcc	4	2.0	8	4	0
	cbcbbccc	4	2.33	7	3	0
9	aaacbcaca	5	2.25	9	4	0
	babacaccc	5	2.25	9	4	0
	cbacbaccc	5	2.25	9	4	0
	cbacbbccc	5	2.25	9	4	0
10	aaabbbabba	6	3.0	9	3	0
	aaacbbcaca	6	2.5	10	4	0
	aaaccbcaca	6	2.5	10	4	0
	aaacccacca	6	3.0	9	3	0
	bbbcccbccb	6	3.0	9	3	0
	cbacbaaccc	6	2.5	10	4	0
11	aaaabbbabba	6	3.0	9	3	0
	aaaacbbcaca	6	2.5	10	4	0
	aaaaccbcaca	6	2.5	10	4	0
	aaaacccacca	6	3.0	9	3	0
	aaabbabcaba	6	2.5	10	4	0
	aaacbbbcaca	6	2.5	10	4	0
	aabbcccbccb	6	2.5	10	4	0
	abbbcccbccb	6	2.5	10	4	0
	acababbaccc	6	2.2	11	5	0
	baacbcbbcbb	6	2.5	10	4	0
	babaabaabbb	6	3.0	9	3	0
	bababccaacb	6	2.2	11	5	0
	babcaabaccc	6	2.2	11	5	0
	babcbcaacbb	6	2.2	11	5	0
	babccaacbcb	6	2.2	11	5	0
	bacaabaaccc	6	2.5	10	4	0
	bbabbabcacc	6	2.5	10	4	0
	bbbbcccbccb	6	3.0	9	3	0
	bbbccbcbacb	6	2.2	11	5	0
	caacbabaccc	6	2.2	11	5	0
	cabaaabbacc	6	2.2	11	5	0
	cacaacaaccc	6	3.0	9	3	0
	cbaacbaaccc	6	2.5	10	4	0
	cbaacbbaccc	6	2.5	10	4	0
	cbacbaacccc	6	2.5	10	4	0
	cbacbabbccc	6	2.2	11	5	0
	cbacbacbbcc	6	2.5	10	4	0
	cbbcbacbacc	6	2.2	11	5	0
	cbcbacbaccc	6	2.5	10	4	0
	cbcbbcbbccc	6	3.0	9	3	0
12	aabbbcccbccb	7	2.75	11	4	0
	bbabbabcaccc	7	2.75	11	4	0
	bcaabcacaacb	7	2.4	12	5	0
	cbaacbbaaccc	7	2.75	11	4	0
	cbacbacbbccc	7	2.75	11	4	0
	cbacbacbcbbc	7	2.4	12	5	0
	cbbacbbaaccc	7	2.75	11	4	0
	cbbcbacbaccc	7	2.4	12	5	0
13	aaacababbaccc	8	2.6	13	5	0
	bababcbcaacbb	8	2.6	13	5	0
	bababccaacbcb	8	2.6	13	5	0
	bcaaabcacaacb	8	2.6	13	5	0
	cbaaacbbaaccc	8	3.0	12	4	0
	cbacbbcbbaccc	8	3.0	12	4	0
14	bcbabcbacbbccc	9	2.8	14	5	0
15	bababcbbcaacbbb	10	3.0	15	5	0

Table A.22
All extremal forward strings s on $\Sigma = \{a, b, c\}$ for with non-positive values of δ_B and $n = 3, \ldots, 15$. Perfectly clustering strings are marked with a 1 in the last column.

n	s	$\delta_B(s)$	$\rho_B(s)$	$r_B(s)$	$r_B(s^{\text{rev}})$	perfectly clustering
3	aab	0	1.0	2	2	1
	aac	0	1.0	2	2	1
	abb	0	1.0	2	2	1
	abc	0	1.0	3	3	0
	acb	0	1.0	3	3	0
	acc	0	1.0	2	2	1
	bac	0	1.0	3	3	0
	bbc	0	1.0	2	2	1
	bcc	0	1.0	2	2	1
4	abab	-2	2.0	2	4	1
	acac	-2	2.0	2	4	1
	bcbc	-2	2.0	2	4	1
5	aabab	-2	2.0	2	4	1
	aacac	-2	2.0	2	4	1
	ababb	-2	2.0	2	4	1
	abacb	-2	1.67	3	5	0
	abcac	-2	1.67	3	5	0
	abcbc	-2	1.67	3	5	0
	acacc	-2	2.0	2	4	1
	accbb	-2	1.67	3	5	0
	bbcbc	-2	2.0	2	4	1
	bcbcc	-2	2.0	2	4	1
6	abcaac	-3	2.0	3	6	0
	abcabc	-3	2.0	3	6	0
	acabab	-3	2.0	3	6	0
7	abacacb	-4	2.33	3	7	0
	acbaccb	-4	2.33	3	7	0
8	abbcaabc	-5	2.67	3	8	0
	abcabcbc	-5	2.67	3	8	0
9	aabbcaabc	-5	2.67	3	8	0
	abacabbac	-5	2.67	3	8	0
	abcabcbbc	-5	2.25	4	9	0
	acbbcacbc	-5	2.67	3	8	0
10	abbcbcaabc	-6	2.5	4	10	0
	abcbcabcbc	-6	3.0	3	9	0
11	abacbabbabb	-7	3.33	3	10	0
	abcabcbbcbc	-7	3.33	3	10	0
	abcbbcabcbc	-7	2.75	4	11	0
12	abbcaacccabc	-7	2.4	5	12	0
	abbcbbcacabc	-7	2.4	5	12	0
	abcabcbbccbc	-7	2.4	5	12	0
	abcbbccabcbc	-7	2.4	5	12	0
	acaccaaacacb	-7	2.75	4	11	0
	cabcbbcabcbc	-7	2.4	5	12	0
13	abaccbbabbabb	-8	3.0	4	12	0
	abcbbcbcabcbc	-8	3.67	3	11	0
14	abacbabcccbacb	-9	2.8	5	14	0
15	abacacbababcacb	-9	2.8	5	14	0
	abacacbabbacacb	-9	3.25	4	13	0
	abacbabcaccaccb	-9	2.5	6	15	0
	abacbabccccbacb	-9	2.8	5	14	0
	abcacbcabcbabcb	-9	2.8	5	14	0
	abcacbcabcbbcbc	-9	3.25	4	13	0
	abcbbcccbcabcbc	-9	2.8	5	14	0
	abccabcccaacaac	-9	3.25	4	13	0
	abcccaaccacabab	-9	2.5	6	15	0
	abcccabcaacabab	-9	2.5	6	15	0
	abcccabcaacabac	-9	2.5	6	15	0
	acacabcbababccb	-9	2.8	5	14	0
	acbcaacacbaaacb	-9	2.5	6	15	0
	accaacaccbaaacb	-9	2.5	6	15	0
	bbbacabcacbabcb	-9	2.5	6	15	0
	bbbaccabcccacac	-9	2.5	6	15	0
	bbcacbcabcbbcbc	-9	2.8	5	14	0
	bbcacbcbbbaccbc	-9	2.5	6	15	0
	bbcbbcaaabcabcb	-9	2.8	5	14	0

Table A.23
Statistics of the differences in the number of Lyndon factors for all forward string s on $\Sigma = \{a, b, c\}$ and $n = 3, \ldots, 15$: minimum and maximum values, mean and standard deviation.

	$\ell(s) - \ell(s^{\text{rev}})$			
n	min	max	mean	sd
3	-2	0	-1.667	0.707
4	-3	1	-1.611	1.153
5	-4	2	-1.815	1.409
6	-5	3	-1.792	1.567
7	-6	4	-1.852	1.714
8	-7	5	-1.855	1.804
9	-8	6	-1.882	1.889
10	-9	7	-1.891	1.954
11	-10	8	-1.909	2.014
12	-11	9	-1.919	2.066
13	-12	10	-1.933	2.113
14	-13	11	-1.942	2.156
15	-14	12	-1.952	2.196

Table A.24
Statistics of the differences in the number of distinct Lyndon factors for all forward string s on $\Sigma = \{a, b, c\}$ and $n = 3, \ldots, 15$: minimum and maximum values, mean and standard deviation.

	$d(s) - d(s^{\text{rev}})$			
n	min	max	mean	sd
3	-2	0	-1	0.5
4	-2	0	-0.917	0.77
5	-3	1	-0.972	0.837
6	-3	1	-1.011	0.92
7	-4	2	-1.061	0.991
8	-4	2	-1.085	1.07
9	-5	2	-1.121	1.139
10	-5	3	-1.143	1.206
11	-5	3	-1.166	1.267
12	-6	4	-1.183	1.325
13	-6	4	-1.199	1.378
14	-6	4	-1.212	1.427
15	-7	5	-1.224	1.472

References

[1] T. Akagi, M. Funakoshi, S. Inenaga, Sensitivity of string compressors and repetitiveness measures, Inf. Comput. 291 (2023) 104999.

[2] G. Badkobeh, H. Bannai, D. Köppl, Bijective BWT based compression schemes, in: Zs. Lipták, E.S. de Moura, K. Figueroa, R. Baeza-Yates (Eds.), String Processing and Information Retrieval - 31st International Symposium, Proceedings, SPIRE 2024, Puerto Vallarta, Mexico, September 23-25, 2024, in: Lecture Notes in Computer Science, vol. 14899, Springer, 2024, pp. 16–25.

[3] H. Bannai, J. Kärkkäinen, D. Köppl, M. Piatkowski, Indexing the bijective BWT, in: Proc. of 30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019, in: LIPIcs, vol. 128, 2019, pp. 17:1–17:14.

[4] H. Bannai, J. Kärkkäinen, D. Köppl, M. Piatkowski, Constructing the bijective and the extended Burrows-Wheeler Transform in linear time, in: Proc. of 32nd Annual Symposium on Combinatorial Pattern Matching, CPM 2021, in: LIPIcs, vol. 191, 2021, pp. 7:1–7:16.
[5] T.C. Bell, J.G. Cleary, I.H. Witten, Text Compression, Prentice Hall, 1990.
[6] J. Berstel, A. de Luca, Sturmian words, Lyndon words and trees, Theor. Comput. Sci. 178 (1–2) (1997) 171–203.
[7] E. Biagi, On comparing the Bijective Burrows-Wheeler-Transform of a word and its reverse, Master's thesis, University of Verona, Dept. of Computer Science, Verona, Italy, 2022.
[8] E. Biagi, D. Cenzato, Zs. Lipták, G. Romana, On the number of equal-letter runs of the Bijective Burrows-Wheeler Transform, in: Proc. of the 24th Italian Conference on Theoretical Computer Science, ICTCS 2023, in: CEUR Workshop Proceedings, vol. 3587, 2023, pp. 129–142.
[9] J. Borel, C. Reutenauer, On Christoffel classes, RAIRO Theor. Inform. Appl. 40 (1) (2006) 15–27.
[10] M. Burrows, D.J. Wheeler, A block-sorting lossless data compression algorithm, Technical report, DIGITAL System Research Center, 1994.
[11] K.T. Chen, R.H. Fox, R. Lyndon, Free differential calculus, iv. The quotient groups of the lower central series, Ann. Math. 68 (1958) 81.
[12] J.W. Daykin, R. Groult, Y. Guesnet, T. Lecroq, A. Lefebvre, M. Léonard, É. Prieur-Gaston, A survey of string orderings and their application to the Burrows-Wheeler transform, Theor. Comput. Sci. 710 (2018) 52–65.
[13] J.W. Daykin, W.F. Smyth, A bijective variant of the Burrows-Wheeler transform using v-order, Theor. Comput. Sci. 531 (2014) 77–89.
[14] A. de Luca, A combinatorial property of the Fibonacci words, Inf. Process. Lett. 12 (4) (1981) 193–195.
[15] A. de Luca, Sturmian words: structure, combinatorics, and their arithmetics, Theor. Comput. Sci. 183 (1) (1997) 45–82.
[16] A. de Luca, F. Mignosi, Some combinatorial properties of Sturmian words, Theor. Comput. Sci. 136 (2) (1994) 361–385.
[17] T. Gagie, G. Navarro, N. Prezza, Fully functional suffix trees and optimal text searching in BWT-runs bounded space, J. ACM 67 (1) (2020) 2:1–2:54.
[18] J.Y. Gil, D.A. Scott, A bijective string sorting transform, CoRR, arXiv:1201.3077, 2012.
[19] S. Giuliani, S. Inenaga, Zs. Lipták, N. Prezza, M. Sciortino, A. Toffanello, Novel results on the number of runs of the Burrows-Wheeler-Transform, in: Proc. of 47th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2021, in: LNCS, vol. 12607, 2021, pp. 249–262.
[20] S. Giuliani, S. Inenaga, Zs. Lipták, G. Romana, M. Sciortino, C. Urbina, Bit catastrophes for the Burrows-Wheeler Transform, in: Proc. of Developments in Language Theory - 27th International Conference, DLT 2023, in: LNCS, vol. 13911, 2023, pp. 86–99.
[21] D. Köppl, D. Hashimoto, D. Hendrian, A. Shinohara, In-place bijective Burrows-Wheeler transforms, in: Proc. of 31st Annual Symposium on Combinatorial Pattern Matching, CPM 2020, in: LIPIcs, vol. 161, 2020, pp. 21:1–21:15.
[22] M. Kufleitner, On bijective variants of the Burrows-Wheeler transform, in: Proc. of the Prague Stringology Conference 2009, 2009, pp. 65–79.
[23] B. Langmead, S.L. Salzberg, Fast gapped-read alignment with bowtie 2, Nat. Methods 9 (4) (2012) 357–359.
[24] B. Langmead, C. Trapnell, M. Pop, S.L. Salzberg, Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, Genome Biol. 10 (2009) R25.
[25] A. Lempel, J. Ziv, On the complexity of finite sequences, IEEE Trans. Inf. Theory 22 (1) (1976) 75–81.
[26] H. Li, R. Durbin, Fast and accurate long-read alignment with Burrows-Wheeler transform, Bioinformatics 26 (5) (2010) 589–595.
[27] V. Mäkinen, G. Navarro, Succinct suffix arrays based on run-length encoding, Nord. J. Comput. 12 (1) (2005) 40–66.
[28] S. Mantaci, A. Restivo, G. Rosone, M. Sciortino, An extension of the Burrows-Wheeler transform, Theor. Comput. Sci. 387 (3) (2007) 298–312.
[29] S. Mantaci, A. Restivo, M. Sciortino, Burrows-Wheeler transform and Sturmian words, Inf. Process. Lett. 86 (5) (2003) 241–246.
[30] G. Navarro, Indexing highly repetitive string collections, part I: Repetitiveness measures, ACM Comput. Surv. 54 (2) (2022) 29:1–29:31.
[31] G. Rosone, M. Sciortino, The Burrows-Wheeler transform between data compression and combinatorics on words, in: Computation in Europe (CiE 2013), in: LNCS, vol. 7921, 2013, pp. 353–365.
[32] J.A. Storer, T.G. Szymanski, Data compression via textual substitution, J. ACM 29 (4) (1982) 928–951.
[33] M. Vasimuddin, S. Misra, H. Li, S. Aluru, Efficient architecture-aware acceleration of BWA-MEM for multicore systems, in: 2019 IEEE International Parallel and Distributed Processing Symposium, IEEE, Rio de Janeiro, Brazil, IPDPS 2019, May 20-24, 2019, pp. 314–324.