

# Smallest Suffixient Sets as a Repetitiveness Measure

Gonzalo Navarro<sup>1,2</sup>[0000–0002–2286–741X],  
Giuseppe Romana<sup>3</sup>[0000–0002–3489–0684], and  
Cristian Urbina<sup>1,2</sup>[0000–0001–8979–9055]

<sup>1</sup> Department of Computer Science, University of Chile, Chile

<sup>2</sup> Center for Biotechnology and Bioengineering (CeBiB), Chile

{gnavarro, crurbina}@dcc.uchile.cl

<sup>3</sup> Department of Mathematics and Computer Science, University of Palermo, Italy

giuseppe.romana01@unipa.it

**Abstract.** A suffixient set is a novel combinatorial object that captures the essential information of repetitive strings in a way that, provided with a random access mechanism, supports various forms of pattern matching. In this paper, we study the size  $\chi$  of the smallest suffixient set as a repetitiveness measure: we place it between known measures and study its sensitivity to various string operations.

**Keywords:** Repetitive sequences · Burrows-Wheeler Transform · Text compressibility

## 1 Introduction

The study of repetitive string collections has recently attracted considerable interest from the stringology community, triggered by practical challenges such as representing huge collections of similar strings in a way that they can be searched and mined directly in highly compressed form [25,26]. An example is the *European '1+ Million Genomes' Initiative*<sup>4</sup>, which aims at sequencing over a million human genomes: while this data requires around 750TB of storage in raw form (using 2 bits per base), the high similarity between human genomes would allow storing it in queriable form using two orders of magnitude less space.

An important aspect of this research is to understand how to measure repetitiveness, especially when those measures reflect the size of compressed representations that offer different access and search functionalities on the collection. Various repetitiveness measures have been proposed, from abstract lower bounds to those related to specific text compressors and indices; a relatively up-to-date survey is maintained [27]. Understanding how those measures relate to each other sheds light on what search functionality is obtained at what space cost.

A relevant measure proposed recently is the size  $\chi$  of the smallest *sufficient* set of the text collection [6], whose precise definition will be given later. Within

---

<sup>4</sup> <https://digital-strategy.ec.europa.eu/en/policies/1-million-genomes>

$O(\chi)$  size, plus a random-access mechanism on the string, it is possible to support some text search functionalities, such as finding one occurrence of a pattern, or finding its maximal exact matches (MEMs), which is of central use on various bioinformatic applications [4].

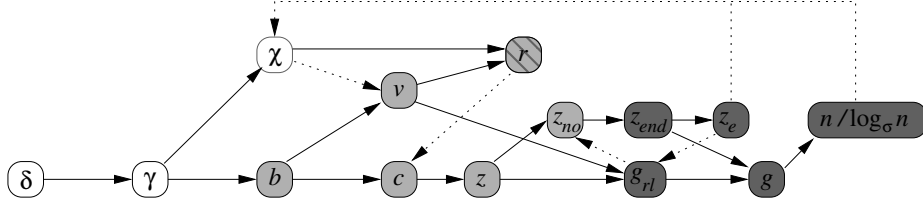
While there has been some work already on how to build minimal suffixient sets and how to index and search a string within their size, less is known about that size,  $\chi$ , as a measure of repetitiveness. It is only known [6] that  $\gamma = O(\chi)$  and  $\chi = O(\bar{r})$  on every string family, where  $\gamma$  is the size of the smallest *string attractor* of the collection (a measure that lower bounds most repetitiveness measures) [18] and  $\bar{r}$  is the number of equal-letter runs of the Burrows-Wheeler Transform (BWT) [3] of the reversed string.

In this paper we better characterize  $\chi$  as a repetitiveness measure. First, we study how it behaves when the string undergoes updates, showing in particular that it grows by  $O(1)$  when appending or prepending symbols, but that it can grow additively by  $\Omega(\log n)$  upon arbitrary edit operations or rotations, and by  $\Omega(\sqrt{n})$  when reversing the string. Second, we show that  $\chi = O(r)$  on every string family, where  $r$  is the number of equal-letter runs of the BWT of the string. We also show that there are string families where  $\chi = o(v)$ , where  $v$  is the size of the smallest lexicographic parse [28] (an alternative to the size of the Lempel-Ziv parse [20], which behaves similarly). In particular, this holds on the Fibonacci strings, where we fully characterize the only 2 smallest suffixient sets of size 4, and further prove that  $\chi \leq \sigma + 2$  on all substrings of episturmian words over an alphabet of size  $\sigma$ . Since  $v = O(r)$  on all string families, this settles  $\chi$  as a strictly smaller measure than  $r$ , which is a more natural characterization than in terms of the reverse string. We also show that  $\chi$  is incomparable with most “copy-paste” based measures [25], as there are families where it is strictly smaller and others where it is strictly larger than any of those measures.

This result relates to the important question of whether a measure  $\mu$  is *reachable* (i.e., one can represent the string within  $O(\mu)$  space), *accessible* (i.e., one can access any string position from an  $O(\mu)$ -size representation, in sublinear time), or *searchable* (i.e., one can search for patterns in sublinear time within space  $O(\mu)$ ). Measure  $r$  is, curiously, the only one to date being reachable and searchable, but not known to be accessible. Now  $\chi$  emerges as a measure smaller than  $r$ , which can search if provided with a mechanism to efficiently access substrings ( $r$  does not need access to support searches). Unlike  $r$ ,  $\chi$  is yet not known to be reachable (as its relation to the smallest known reachable measure, the size  $b$  of the smallest bidirectional macro scheme [31], remains unknown). As said, it is known that  $\gamma = O(\chi)$ , but it is unknown whether  $\gamma$  is reachable or not.

## 2 Preliminaries

An *ordered alphabet*  $\Sigma = \{a_1, \dots, a_\sigma\}$  is a finite set of symbols equipped with a total order  $<$  such that  $a_1 < a_2 < \dots < a_\sigma$ . When  $\sigma = 2$ , we assume  $\Sigma = \{\mathbf{a}, \mathbf{b}\}$  with  $\mathbf{a} < \mathbf{b}$ . The special symbol  $\$$ , if it appears, is always assumed to be the smallest of the alphabet.



**Fig. 1.** Relations between relevant repetitiveness measures and how our results place  $\chi$  among them. An arrow  $\mu_1 \rightarrow \mu_2$  means that  $\mu_1 = O(\mu_2)$  for all strings and, save for  $c \rightarrow z$ ,  $z_{no} \rightarrow z_{end}$ , and  $z_{end} \rightarrow z_e$ , there is a string family where  $\mu_1 = o(\mu_2)$ . The dotted arrows mark only this last condition, so they are not transitive. Measures in light gray nodes are known to be reachable; those in dark gray are accessible and searchable; and  $r$  is hatched because it is searchable but not known to be accessible.

A *string*  $w[1..n]$  (or simply  $w$  if it is clear from the context) of *length*  $|w| = n$  over the alphabet  $\Sigma$  is a sequence  $w[1]w[2]\dots w[n]$  of symbols where  $w[i] \in \Sigma$  for all  $i \in [1, n]$ . The *empty string* of length 0 is denoted  $\epsilon$ . We denote by  $\Sigma^*$  the set of all strings over  $\Sigma$ . Additionally, we let  $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$  and  $\Sigma^k = \{w \in \Sigma^* \mid |w| = k\}$ . We denote by  $w[i..j]$  the substring  $w[i]w[i+1]\dots w[j]$ . If  $x = x[1..n]$  and  $y = y[1..m]$  are strings, we define the *concatenation operation* applied on  $x$  and  $y$ , as the string obtained by juxtaposing these two strings, that is,  $x \cdot y = x[1]x[2]\dots x[n]y[1]\dots y[m] = xy$ . A string  $x$  is a *substring* of  $w$  if  $w = yxz$  for some  $y, z \in \Sigma^*$ . A string  $x$  is a *prefix* of  $w$  if  $w = xy$  for some  $y \in \Sigma^*$ . Analogously,  $x$  is a *suffix* of  $w$  if  $w = yx$  for some  $y \in \Sigma^*$ . We say that substrings, prefixes, and suffixes are *non-trivial* if they are different from  $w$  and  $\epsilon$ . The set of substrings of  $w$  is denoted by  $\mathcal{F}_w$ . We also let  $\mathcal{F}_w(k) = \mathcal{F}_w \cap \Sigma^k$ . The *reverse* of a finite string  $w$  is the string  $w^R = w[n] \cdot w[n-1] \dots w[1]$ . We denote by  $\mathcal{R}(w)$  the multiset of rotations of  $w[1..n]$ , that is,  $\mathcal{R}(w) = \{w[i+1..n]w[1..i] \mid i \in [1..n]\}$ . The *Burrows-Wheeler transform* (BWT) of a string  $w$ , denoted  $\text{BWT}(w)$ , is the transformation of  $w$  obtained by collecting the last symbol of all rotations in  $\mathcal{R}(w)$  in lexicographic order. The *BWT matrix*  $B(w)$  of  $w$  is the  $(n \times n)$ -matrix where the  $i$ -th row is the  $i$ -th rotation of  $w$  in lexicographic order.

A *right-infinite string*  $\mathbf{w}$ —we use **boldface** to emphasize its infinite length—over  $\Sigma$  is any infinite sequence  $\mathbb{Z}^+ \rightarrow \Sigma$ . The set of all infinite strings over  $\Sigma$  is denoted  $\Sigma^\omega$ . A substring of  $\mathbf{w}$  is the finite string  $\mathbf{w}[i..j]$  for any  $1 \leq i \leq j$ . A prefix of  $\mathbf{w}$  is a finite substring of the form  $\mathbf{w}[1..n]$  for some  $n \geq 0$ . The *substring complexity function*  $P_{\mathbf{w}}(k) : \mathbb{Z}^+ \cup \{0\} \rightarrow \mathbb{Z}^+$  counts the number of distinct substrings of length  $k$  in  $\mathbf{w}$ , for any  $k \in \mathbb{Z}^+ \cup \{0\}$ , that is,  $P_{\mathbf{w}}(k) = |\mathcal{F}_{\mathbf{w}}(k)|$ . For a finite string  $w[1..n]$ , the domain of  $P_w$  is restricted to  $[0..n]$ .

## 2.1 Measures of Repetitiveness

In this work, we will relate  $\chi$ , in asymptotic terms, with several well-established measures of repetitiveness [25,27]:  $\delta = \max_{k \in [0..n]} (\mathcal{F}_w(k)/k)$  (a measure of substring complexity),  $\gamma$  (the smallest string attractor),  $b$  (the size of the smallest

bidirectional macro scheme),  $z$  (the size of the Lempel-Ziv parse),  $z_{no}$  (the same without allowing phrases to overlap their sources),  $z_e$  (the size of the greedy LZ-End parse),  $z_{end}$  (the size of the minimal LZ-End parse),  $v$  (the size of the smallest lexicographic parse),  $r$  (the number of equal-letter runs in the BWT of the string),  $g$  (the size of the smallest context-free grammar generating only the string),  $g_{rl}$  (the same allowing run-length rules), and  $c$  (the size of the smallest collage system generating only the string). Except for  $\delta$ ,  $\gamma$  and  $r$ , these measures are said to be *copy-paste* because they refer to a way of cutting the sequence into chunks that can be copied from elsewhere in the same sequence. Indeed,  $\delta$  and  $\gamma$  are lower-bound measures, the former known to be unreachable and the latter not known to date to be reachable; all the others are. The smallest measures known to be accessible (and searchable) are  $z_{end}$  and  $g_{rl}$ , and  $r$  is searchable but not known to be accessible.

The known relations between those measures are summarized in Fig. 1, where we have added the results we obtain in this paper with respect to  $\chi$ .

## 2.2 Edit Operations and Sensitivity Functions

The so-called *edit operations* are *insertion*, *substitution* and *deletion* of a single character on a string. We denote  $\mathbf{ins}_\Sigma(w)$ ,  $\mathbf{sub}_\Sigma(w)$ ,  $\mathbf{del}_\Sigma(w)$  the sets of strings that can be obtained by applying an edit operation to  $w$ . In addition, we let  $\mathbf{prepend}_\Sigma(w)$  and  $\mathbf{append}_\Sigma(w)$  be  $\mathbf{ins}_\Sigma(w)$  restricted to the insertion being made at the beginning and the end of the string, respectively.

A repetitiveness measure  $\mu$  is *monotone* or *non-decreasing* to the insertion of a single character if  $\mu(w') - \mu(w) \geq 0$  for any  $w$  and  $w' \in \mathbf{ins}_\Sigma(w)$ . More generally, the *additive sensitivity* and *multiplicative sensitivity* functions of a repetitiveness measure  $\mu$  to the insertion of a single character are the maximum possible values of  $\mu(w') - \mu(w)$  and  $\mu(w')/\mu(w)$ , respectively. We define the concept of monotonicity and sensitivity functions for the remaining string operations analogously.

## 3 Suffixient Sets and the Measure $\chi$

In this section we define the central combinatorial objects and measures we analyse on this work. Note that some of our definitions are slightly different from their original formulation [4,5], because we do not always assume that all strings are  $\$$ -terminated.

### Definition 1 (Right-maximal Substrings and Right-extensions [4,5]).

Let  $w \in \Sigma^*$ . A substring  $x$  of  $w$  is *right-maximal* if there exist at least two distinct symbols  $a, b \in \Sigma$  such that both  $xa$  and  $xb$  are substrings of  $w$ . For any right-maximal substring  $x$  of  $w$ , the substrings  $xa$  with  $a \in \Sigma$  are called *right-extensions*. We denote the set of right-extensions in  $w$  by  $E_r(w) = \{xa \mid \exists b : b \neq a, xa \in \mathcal{F}_w, xb \in \mathcal{F}_w\}$ .

We distinguish a special class of right-extensions that are not suffixes of any other right-extension.

**Definition 2 (Super-maximal Extensions [4,5]).** *The set of super-maximal extensions of  $w$  is  $\mathcal{S}_r(w) = \{x \in E_r(w) \mid \forall y \in E_r, y = zx \Rightarrow z = \varepsilon\}$ . Moreover, we let  $\mathbf{sre}(w) = |\mathcal{S}_r(w)|$ .*

We now define suffixient sets for strings not necessarily  $\$$ -terminated; we introduce later the special terminator  $\$$ .

**Definition 3 (Suffixient Set [4,5]).** *Let  $w[1..n] \in \Sigma^*$ . A set  $S \subseteq [1..n]$  is a suffixient set for  $w$  if for every right-extension  $x \in E_r(w)$  there exists  $j \in S$  such that  $x$  is a suffix of  $w[1..j]$ .*

Intuitively, a suffixient set is a collection of positions of  $[1..|w|]$  capturing all the right-extensions appearing in  $w$ . The smallest suffixient sets, which are suffixient sets of minimum size, have also been characterized in terms of super-maximal right-extensions. The next definition simplifies the original one [4,5].

**Definition 4 (Smallest Suffixient Set).** *Let  $w[1..n] \in \Sigma^*$ . A suffixient set  $S \subseteq [1..n]$  is a smallest suffixient set for  $w$  if there is a bijection  $\text{pos} : \mathcal{S}_r \rightarrow S$  such that every  $x \in \mathcal{S}_r$  is a suffix of  $w[1..\text{pos}(x)]$ .*

In its original formulation, the measure  $\chi$  is defined over  $\$$ -terminated strings. Here, we define  $\chi(w)$  with the  $\$$  being implicit, not being part of  $w$ .

**Definition 5 (Measure  $\chi$  [4,5]).** *Let  $w \in \Sigma^*$  and assume  $\$ \notin \mathcal{F}_w$ . Then,  $\chi(w) = |\mathcal{S}|$ , where  $\mathcal{S}$  is a smallest suffixient set for  $w\$$ .*

One can see from the above definitions that  $\chi$  is well-defined because  $\chi(w) = \mathbf{sre}(w\$)$ . We will use this relation to prove results on  $\chi$  via  $\mathbf{sre}$ .

## 4 Sensitivity of $\chi$ to String Operations

The sensitivity to string operations has been studied for many repetitiveness measures [1,9,10,14,15,24,29,30]. It is desirable for a repetitiveness measure to not change much upon small changes in the sequence. Some repetitiveness measures are resistant to edit operations. For instance,  $b$ ,  $z$  and  $g$  can only increase by a multiplicative constant after an edit operation [1], though they can increase only by a  $O(1)$  additive factor when prepending or appending a character. On the other hand,  $r$  can increase by a  $\Theta(\log n)$  factor when appending a character [15, Prop. 37]. Other results have been obtained concerning more complex string operations, like reversing a string [14], or applying a string morphism [9,10].

In this section we study how  $\mathbf{sre}$  and  $\chi$  behave in this respect. We start by proving the following useful lemma.

**Lemma 1.** *If  $E_r(w_1) \subseteq E_r(w_2)$ , then  $\mathbf{sre}(w_1) \leq \mathbf{sre}(w_2)$ .*

*Proof.* Let  $x, y \in \mathcal{S}_r(w_1)$  with  $x \neq y$ . Because  $x \in E_r(w_2)$ , there exists  $z \in \mathcal{S}_r(w_2)$  with  $x$  a suffix of  $z$ . Because  $y$  is not a suffix of  $x$  and vice versa,  $y$  cannot be a suffix of  $z$ . Therefore, the map  $x \mapsto z$  with  $x \in \mathcal{S}_r(w_1)$ ,  $z \in \mathcal{S}_r(w_2)$ , and  $z = z'x$  for some  $z' \in \Sigma^*$  is injective and then  $\mathbf{sre}(w_1) \leq \mathbf{sre}(w_2)$ .  $\square$

We now prove that  $\mathbf{sre}(w)$  grows only by  $O(1)$  when prepending or appending characters.

**Lemma 2.** *Let  $w \in \Sigma^*$ , and  $c \in \Sigma$ . It holds  $\mathbf{sre}(w) \leq \mathbf{sre}(wc) \leq \mathbf{sre}(w) + 2$ .*

*Proof.* The lower bound follows from Lemma 1. For the upper bound, we analyse the new right-extensions that may arise due to appending  $c$  to  $w$ . For any fixed suffix  $xc$  of  $wc$ :

1. if  $xa$  does not appear in  $w$  for any  $a \neq c$ , then  $xc$  induces no new right-extensions in  $wc$ ;
2. if for some  $a \neq b$ ,  $xa$  and  $xb$  were both substrings of  $w$ , and  $c \neq a$  and  $c \neq b$ , then  $xc$  is a new right-extension of  $wc$ ;
3. if  $x$  is always followed by  $a \neq c$  in  $w$  (hence,  $xa$  is not a right-extension of  $w$ ), then both  $xa$  and  $xc$  are new right-extensions of  $wc$ .

Cases 1 and 2 induce at most one new super-maximal right-extension in total for all possible  $xc$ , namely the longest right-extension in  $wc$  that is a suffix of  $wc$ . For Case 3, consider a fixed  $a \in \Sigma$ . For all the increasing-length suffixes  $x_1c, x_2c, \dots, x_tc$  of  $wc$  that became right-extensions together with  $x_1a, x_2a, \dots, x_ta$ , one can see that the latter form a chain of suffixes of  $x_ta$ . Hence, we only have one possible new super-maximal right-extension ending with  $a$ , namely  $x_ta$ . Observe that the chain of suffixes  $x_1a, x_2a, \dots, x_ta$  is unique: if the suffix  $x$  is always followed by  $a$ , any suffix  $y$  of  $x$  is either right-maximal in  $w$  (and  $y$  falls within Case 2), or it is always followed by an  $a$  (because  $x$  is always followed by an  $a$ ), i.e.  $y = x_i$  for some  $i \in [1..t]$ .  $\square$

**Lemma 3.** *Let  $w \in \Sigma^*$  and  $c \in \Sigma$ . It holds  $\mathbf{sre}(w) \leq \mathbf{sre}(cw) \leq \mathbf{sre}(w) + 2$ .*

*Proof.* The lower bound follows from Lemma 1. For the upper bound, let  $cxa$  be the smallest prefix of  $cw$  that is not a right-extension of  $w$ , but is a right-extension of  $cw$  (if it exists). This means that  $cxa$  does not appear in  $w$  (otherwise, it would be a right-extension of  $w$ ), so no prefix of  $cw$  of length  $|cxa|$  or more is right-maximal. Hence, all prefixes of  $cw$  shorter than  $cxa$  were already right-extensions, and all prefixes longer than  $cxa$  cannot be right-extensions. Therefore,  $cxa$  together with some  $cxb$  appearing in  $w$  are the only possible new right-extensions in  $cw$  with respect to  $w$ .  $\square$

By letting  $c = \$ \notin \mathcal{F}_w$  in Lemma 2, we relate  $\chi$  to  $\mathbf{sre}$  (note that  $\chi$  is always at least  $\mathbf{sre} + 1$  because of the new super-maximal extension ending with  $\$$ ). This makes clear the relation between Combinatorics on words [21] with suffixient sets, via the common notion of *right-special factors* (what we call here right-maximal substrings).

**Corollary 1.** *Let  $w \in \Sigma^*$ . It holds  $\mathbf{sre}(w) + 1 \leq \chi(w) \leq \mathbf{sre}(w) + 2$ .*

Note that, while the value  $\mathbf{sre}(w)$  is non-decreasing after appending a character, this is not the case for the measure  $\chi$ .

**Lemma 4.** *The measure  $\chi$  is not monotone to appending a character.*

*Proof.* Let  $w = \text{abaab}$ . It holds  $\mathcal{S}_r(w\$) = \{\text{aa}, \text{ab}, \text{ab\$}, \text{aba}\}$  and  $\mathcal{S}_r(wa\$) = \mathcal{S}_r(\text{abaaba\$}) = \{\text{ab}, \text{aba\$}, \text{abaa}\}$ . Hence,  $\chi(w) = 4$  and  $\chi(wa) = 3$ .  $\square$

Now we study how much  $\mathbf{sre}(w)$  can vary upon edit operations in arbitrary positions, rotations, and reversals. We will use the following famous string family.

**Definition 6.** *A binary de Bruijn sequence of order  $k > 0$  [2] contains every binary string in  $\{\mathbf{a}, \mathbf{b}\}^k$  as a substring exactly once. The length of these strings is  $n = 2^k + (k - 1)$ . The set of binary de Bruijn sequences of order  $k$  is  $\mathbf{dB}(k)$ .*

**Lemma 5.** *It holds  $\mathbf{sre}(w) = 2^k = \Omega(n)$  for any  $w[1..n] \in \mathbf{dB}(k)$ .*

*Proof.* Let  $w[1..n]$  be a binary de Bruijn string of order  $k$ . By definition,  $w$  contains every binary string of length  $k$  as a substring exactly once. As all the possible pairs of strings  $x\mathbf{a}$  and  $x\mathbf{b}$  of length  $k$  appear in  $w$ , it follows that all the strings in  $\mathcal{F}_w(k)$  are right-extensions. Moreover, each  $x\mathbf{a}$  and  $x\mathbf{b}$  of length  $k$  are super-maximal right-extensions: otherwise, there would exist some  $c \in \{\mathbf{a}, \mathbf{b}\}$  such that  $cx\mathbf{a}$  and  $cx\mathbf{b}$  are both substrings of  $w$ , which raises a contradiction since the  $k$ -length string  $cx$  cannot appear twice in  $w$ . Moreover, there are no right-maximal strings of length  $k$  or greater; hence, there are no right-extensions of length greater than  $k$ . It follows that  $\mathbf{sre}(w) = |\mathcal{F}_w(k)| = 2^k = \Omega(n)$ .  $\square$

The following lemma uses the de Bruijn family to show that  $\mathbf{sre}$  can grow by  $\Omega(\log n)$  upon arbitrary edit operations and rotations.

**Lemma 6.** *Let  $w = \mathbf{a}^k \mathbf{b} \mathbf{a}^{k-2} \mathbf{b} x \mathbf{a} \mathbf{b}^k \mathbf{a}^{k-1} \in \mathbf{dB}(k)$  be the lexicographically smallest binary de Bruijn sequence of order  $k$  [11, 12]. It holds:*

1. (Ins)  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k - 2$  if  $w' = \mathbf{a}^{2k-2} \mathbf{b} x \mathbf{a} \mathbf{b}^k \mathbf{a}^{k-1}$ ,
2. (Sub)  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k - 3$  if  $w' = \mathbf{a}^k \mathbf{b} \mathbf{a}^{k-2} \mathbf{b} x \mathbf{a} \mathbf{b}^{k-1} c \mathbf{a}^{k-1}$ ,
3. (Del)  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k - 4$  if  $w' = \mathbf{a}^k \mathbf{b} \mathbf{a}^{k-2} \mathbf{b} x \mathbf{a} \mathbf{b}^k c \mathbf{a}^{k-1}$ ,
4. (Rot)  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k - 2$  if  $w' = \mathbf{b} \mathbf{a}^{k-2} \mathbf{b} x \mathbf{a} \mathbf{b}^k \mathbf{a}^{2k-1}$ .

*Proof.* Observe that in each claim,  $w$  is obtained after performing a string operation on the corresponding  $w'$ : in Claim 1,  $w \in \mathbf{ins}_\Sigma(w')$ ; in Claim 2,  $w \in \mathbf{sub}_\Sigma(w')$ ; in Claim 3,  $w \in \mathbf{del}_\Sigma(w')$ ; in Claim 4,  $w \in \mathcal{R}(w')$ . We prove each claim separately by comparing the super-maximal extensions of  $w'$  before and after performing the string operation on  $w'$  that yields  $w$ , for which  $\mathbf{sre}(w) = 2^k$  by Lemma 5.

For Claim 1, note that  $\mathbf{sre}(w')$  is the same as  $\mathbf{sre}(\mathbf{a}^k \mathbf{b} x \mathbf{a} \mathbf{b}^k \mathbf{a}^{k-1})$ , as prepending the character  $\mathbf{a}$  multiple times to this string to obtain  $w'$  never increases  $\mathbf{sre}$ ; it only updates the super-maximal extension  $\mathbf{a}^k$  to  $\mathbf{a}^{k+1}$  and  $\mathbf{a}^{k-1} \mathbf{b}$  to  $\mathbf{a}^k \mathbf{b}$ , and

so on. For simplicity, we let  $w' = \mathbf{a}^k \mathbf{b} x \mathbf{a} \mathbf{b}^k \mathbf{a}^{k-1}$ . The string  $w'$  does not contain substrings of length  $k$  of the form  $\mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1}$  for  $i \in [1 \dots k-2]$ , nor the substring  $\mathbf{b} \mathbf{a}^{k-2} \mathbf{b}$ . Note that for each of these substrings  $y \in \mathcal{F}_w(k)$  with  $y \notin \mathcal{F}_{w'}(k)$ , the other corresponding right-extension  $y'$  in  $w$  sharing a length  $k-1$  prefix with  $y$  is not a right-extension in  $w'$ . Moreover, note that all the suffixes of length  $k-1$  of these  $y$  are not suffixes of one another, nor of the length  $k-1$  suffixes of any of the substrings  $y'$  in  $w'$ . Hence, all  $k-1$  length binary strings still appear in  $w'$  as the suffix of some length  $k$  substring that remains a right-extension in  $w'$ , and hence, super-maximal extensions of  $w'$  have to be of length at least  $k$ . As each string of length  $k$  appearing in  $w'$  is unique, there are no super-maximal extensions of length greater than  $k$ . Thus,  $\mathbf{sre}(w') = 2^k - 2(k-1)$  because we are losing  $k-1$  pairs of super-maximal extensions of length  $k$  with respect to  $w$ . It follows that by inserting the  $\mathbf{b}$  in  $w'$  to yield  $w$ ,  $\mathbf{sre}$  increases by  $2k-2$ .

For Claim 2, note that exactly  $k$  substrings of length  $k$  are lost when substituting the last  $\mathbf{b}$  of  $w$  by  $\mathbf{c}$ : those of the form  $\mathbf{b}^i \mathbf{a}^{k-i}$  with  $i > 0$ . This means that substrings ending in  $\mathbf{b}^i \mathbf{a}^{k-i-1}$  with  $0 < i < k$  are not right-maximal in  $w'$ , hence,  $2(k-1)$  super-maximal extensions are lost. Moreover,  $\mathbf{b}^{k-2}$  is still a right-maximal substring, since  $\mathbf{b}^{k-1}$  and  $\mathbf{b}^{k-2} \mathbf{c}$  occur in  $w'$ . Observe that only  $\mathbf{b}^{k-2} \mathbf{c}$  is a super-maximal extension, while  $\mathbf{b}^{k-1}$  is a suffix of  $\mathbf{a} \mathbf{b}^{k-1}$ . Thus,  $\mathbf{sre}(w') = 2^k - 2(k-1) + 1$  and  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k-3$ .

For Claim 3, the analysis is similar to Claim 2, but in  $w'$ ,  $\mathbf{b}^{k-1}$  remains as a super-maximal extension. Thus,  $\mathbf{sre}(w') = 2^k - 2(k-1) + 2$  and  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k-4$ .

For Claim 4, the analysis is similar to Claim 1, but in  $w'$ ,  $\mathbf{b} \mathbf{a}^{k-2} \mathbf{b}$  appears, while  $\mathbf{a}^{k-1} \mathbf{b}$  does not. Thus,  $\mathbf{sre}(w') = 2^k - 2(k-1)$  and  $\mathbf{sre}(w) - \mathbf{sre}(w') = 2k-2$ .  $\square$

We now show that  $\mathbf{sre}$  can grow by  $\Omega(\sqrt{n})$  upon string reversals.

**Lemma 7.** *Let  $k > 0$ . Let  $w_k = \prod_{i=0}^{k-1} \mathbf{c} \mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \#_i \mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \$ _i$  on the alphabet  $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\} \cup \bigcup_{i \in [0 \dots k-1]} \{\#_i, \$ _i\}$ . It holds  $\mathbf{sre}(w_k) - \mathbf{sre}(w_k^R) = k-1$ .*

*Proof.* Observe that by construction, any substring of  $w_k$  containing  $\#_i$  or  $\$ _i$  is not right-maximal, as these symbols are unique. Hence, the right-extensions of  $w_k$  cannot cross from one side to the other side of those special delimiters. Moreover, substrings of the form  $\mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1}$  for  $i \in [0 \dots k-1]$  appear exactly twice in  $w_k$  and their right-extensions are super-maximal. By looking at the structure of the string  $w_k$  and carefully analyzing its right-extensions, one can verify that the super-maximal right-extensions of  $w_k$  are the following:

1.  $\mathbf{b} \mathbf{a}^{k-1}$  and  $\mathbf{c}$
2.  $\mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \#_i$  and  $\mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \$ _i$  for  $i \in [0 \dots k-1]$ ,
3.  $\mathbf{c} \mathbf{a}^i$  and  $\mathbf{c} \mathbf{a}^{i-1} \mathbf{b}$  for  $i \in [1 \dots k-1]$ ,
4.  $\mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1}$  for  $i \in [1 \dots k-1]$ .

This sums to a total of  $5k-1$  super-maximal extensions in  $w_k$ . In the reversed string  $w_k^R = \prod_{i=0}^{k-1} \$ _{k-i-1} \mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \#_{k-i-1} \mathbf{a}^i \mathbf{b} \mathbf{a}^{k-i-1} \mathbf{c}$ , we have instead:



1.  $\mathbf{ba}^{k-1}$  and  $\mathbf{\$}_{k-1}$ ,
2.  $\mathbf{a}^i \mathbf{ba}^{k-i-1} \#_{k-i-1}$  and  $\mathbf{a}^i \mathbf{ba}^{k-i-1} \mathbf{c}$  for  $i \in [0 \dots k-1]$ ,
3.  $\mathbf{a}^{k-i-1} \mathbf{c} \mathbf{\$}_{k-i-2}$  for  $i \in [1 \dots k-2]$ , and  $\mathbf{a}^{k-2} \mathbf{c} \mathbf{\$}_{k-2}$ ,
4.  $\mathbf{a}^i \mathbf{ba}^{k-i-1}$  for  $i \in [1 \dots k-1]$ .

This sums to a total of  $4k$  super-maximal extensions in  $w_k^R$ . Thus,  $\mathbf{sre}(w_k) - \mathbf{sre}(w_k^R) = (5k-1) - 4k = k-1$ .  $\square$

We give an example of the words  $w_k$  and  $w_k^R$  of Lemma 7, and their super-maximal right-extensions.

*Example 1.* Let  $w_3 = \mathbf{cbaa\#_0baa\$_0caba\#_1aba\$_1caab\#_2aab\$_2}$ . It can be verified that the super-maximal right-extensions of  $w_3$  are:

1.  $\mathbf{baa}$  and  $\mathbf{c}$ ;
2.  $\mathbf{baa\#_0}$  and  $\mathbf{baa\$_0}$ ;  $\mathbf{aba\#_1}$  and  $\mathbf{aba\$_1}$ ;  $\mathbf{aab\#_2}$  and  $\mathbf{aab\$_2}$ ;
3.  $\mathbf{ca}$  and  $\mathbf{cb}$ ;  $\mathbf{caa}$  and  $\mathbf{cab}$ ;
4.  $\mathbf{aba}$  and  $\mathbf{aab}$ .

Similarly, let  $w_3^R = \mathbf{\$ _2baa\#_2baac\$_1aba\#_1abac\$_0aab\#_0aabc}$ . The super-maximal right-extensions of  $w_3^R$  are:

1.  $\mathbf{baa}$  and  $\mathbf{\$ _2}$ ;
2.  $\mathbf{baa\#_2}$  and  $\mathbf{baac}$ ;  $\mathbf{aba\#_1}$  and  $\mathbf{abac}$ ;  $\mathbf{aab\#_0}$  and  $\mathbf{aabc}$ ;
3.  $\mathbf{ac\$_0}$ ;  $\mathbf{ac\$_1}$ ;
4.  $\mathbf{aba}$  and  $\mathbf{aab}$ .

One can see that  $\mathbf{sre}(w_3) = 14$ ,  $\mathbf{sre}(w_3^R) = 12$ , and hence,  $\mathbf{sre}(w_3) - \mathbf{sre}(w_3^R) = 2$ , as stated in Lemma 7.

Formally, the additive sensitivity of a measure of repetitiveness  $\mu$  to a string operation  $\rho$  can be defined as a function  $AS_{\mu,\rho} : \mathbb{Z}^+ \rightarrow \mathbb{R}$ , where  $AS_{\mu,\rho}(n) = \max_{w \in \Sigma^n} (\max_{w' \in \rho(w)} (\mu(w') - \mu(w)))$ , that is the maximum achievable difference among all the strings. Overall, we obtain the following result on the additive sensitivity of  $\mathbf{sre}$ , which, by Corollary 1, can be written in terms of  $\chi$ .

**Corollary 2.** *The following bounds on the additive sensitivity of the measure  $\chi$  to string operations hold:*

1.  $AS_{\chi,\rho}(n) = \Omega(\log n)$  for  $\rho \in \{\mathbf{ins}, \mathbf{del}, \mathbf{sub}, \mathcal{R}(\cdot)\}$ ;
2.  $AS_{\chi,\mathbf{rev}}(n) = \Omega(\sqrt{n})$ , where  $\mathbf{rev}(w) = \{w^R\}$ .

*Proof.* Claim 1 follows by Lemma 6, where  $n = |w| = \Theta(2^k)$  and  $AS_{\chi,\rho}(n) = \Omega(k) = \Omega(\log n)$ , for all  $\rho \in \{\mathbf{ins}, \mathbf{del}, \mathbf{sub}, \mathcal{R}(\cdot)\}$ . Claim 2 follows by Lemma 7, where  $n = |w_k| = \Theta(k^2)$  and  $AS_{\chi,\mathbf{rev}}(n) = \Omega(k) = \Omega(\sqrt{n})$ .  $\square$

Finally, we show upper bounds on the sensitivity of  $\chi$  to string operations.

**Lemma 8.** *Let  $w \in \Sigma^*$  and  $w' \in \text{ins}_\Sigma(w) \cup \text{del}_\Sigma(w) \cup \text{sub}_\Sigma(w) \cup \mathcal{R}(w) \cup \{w^R\}$ . It holds*

$$\begin{aligned}\chi(w') - \chi(w) &= O(\delta \max(1, \log(n/\delta \log \delta)) \log \delta) \text{ and} \\ \chi(w') / \chi(w) &= O(\max(1, \log(n/\delta \log \delta)) \log \delta).\end{aligned}$$

*Proof.* To prove our thesis, we rely on the relations  $\delta \leq \chi \leq 2\bar{r}$  [4] and  $r = O(\delta \max(1, \log(n/\delta \log \delta)) \log \delta)$  [17]. Moreover, since the multiplicative sensitivity of the measure  $\delta$  to any of the string operations is  $O(1)$  [1], for any  $w \in \Sigma^*$  it holds  $\bar{r}(w) = r(w^R) = O(\delta \max(1, \log(n/\delta \log \delta)) \log \delta)$ . The thesis follows by considering the worst case, that is  $\chi(w) = \Theta(\delta)$  and  $\chi(w') = \Theta(\delta \max(1, \log(n/\delta \log \delta)) \log \delta)$ .  $\square$

## 5 Relating $\chi$ to Other Repetitiveness Measures

Previous work [4] established that  $\gamma = O(\chi)$  and  $\chi = O(\bar{r})$  on every string family. In this section we obtain the more natural result that  $\chi$  is always  $O(r)$ , and that it can be asymptotically strictly smaller,  $\chi = o(r)$ , on some string families (we actually prove  $\chi = o(v)$ ). We also show that  $\chi$  is incomparable with all the copy-paste measures except  $b$ , in the sense that there are string families where  $\chi$  is asymptotically strictly smaller than each other, and vice versa.

### 5.1 Proving $\chi = O(r)$

We first prove that  $\chi$  is asymptotically upper-bounded by the number  $r$  of runs in the BWT of the sequence. As for the measure  $\chi$ , we assume that the BWT is computed after appending the  $\$$  symbol.

**Lemma 9.** *It always holds that  $\chi \leq 2r$ .*

*Proof.* Let  $x_i$  denotes the  $i$ th rotation of  $w\$$  in lexicographic order, for each  $i \in [1..|w|+1]$ , and let  $u_i$  be the longest common prefix between the rotations  $x_i, x_{i+1}$ , for each  $i \in [1..|w|]$ . We further define  $s : [1..n+1] \rightarrow [0..n]$  as  $s(i) = j$  if  $x_i = w[j+1..|w|]\$w[1..j]$ , i.e., the number of cyclic shift to the right required to transform  $x_i$  into  $w\$$ .<sup>5</sup> As the symbol  $\$$  occurs only once in  $w\$$ , the function  $s$  is bijective.

Note that each right-extension of  $w\$$  can be written as  $u_i c$ , for some  $i \in [1..|w|]$  and  $c \in \Sigma$ . Consider now the set

$$S = \bigcup_{i \in [1..|w|]} \{s(i) + |u_i| + 1, s(i+1) + |u_i| + 1\},$$

that is the set of positions where the occurrences of the right-extensions  $u_i c_1$  and  $u_i c_2$  end in  $w\$$ , where  $u_i c_1$  and  $u_i c_2$  are the prefix of  $x_i$  and  $x_{i+1}$  respectively,

<sup>5</sup> The function  $s$  mimics the well-known Suffix Array [23], here omitted for simplicity of exposition.

for some  $c_1, c_2 \in \Sigma$  such that  $c_1 < c_2$ . It follows by construction that the set  $S$  is a suffixient set of  $w\$$ .

We now show that  $|S| \leq 2r$ . Let us factorize each pair of consecutive rotations in the BWT-matrix as  $x_i = u_i v_i c_i$  and  $x_{i+1} = u_i v'_i c_{i+1}$ . Observe that  $v_i, v'_i \neq \varepsilon$  [9, Corollary 8],  $v_i[1] \neq v'_i[1]$ , and  $c_i = \text{BWT}(w\$)[i]$  for all  $i \in [1..|w| + 1]$ . A well-known property of the BWT-matrix is that if  $c_i = c_{i+1} = c \in \Sigma$ , then there exists  $j \in [1..|w|]$  such that  $x_j = cu_i v_i$  and  $x_{j+1} = cu_i v'_i$  [3]. As a consequence, one has that  $s(j) + |u_j| + 1 = (s(i) - 1) + (|u_i| + 1) + 1 = s(i) + |u_i| + 1$  and  $s(j+1) + |u_j| + 1 = (s(i+1) - 1) + (|u_i| + 1) + 1 = s(i+1) + |u_i| + 1$ , and the procedure can be reiterated as long as  $x_j$  and  $x_{j+1}$  end with the same symbol. It follows that the same set can be written as

$$S = \{s(i) + |u_i| + 1, s(i+1) + |u_i| + 1 \mid i \in [1..|w|] \wedge \text{BWT}[i] \neq \text{BWT}[i+1]\},$$

i.e., the size of  $S$  is at most twice the number of equal-letter runs in  $\text{BWT}(w\$)$ , and the thesis follows.  $\square$

## 5.2 A Family with $\chi = o(v)$ (and thus $o(r)$ )

We will now show that  $\chi = o(v)$  on the so-called Fibonacci words, which also implies  $\chi = o(r)$  in that string family because  $v = O(r)$  [28]. Combined with Lemma 9, this implies that  $\chi$  is a strictly smaller measure than  $r$ . In contrast,  $\chi$  is incomparable with  $v$ , as we show later. On our way, we obtain some relevant byproducts about the structure of suffixient sets on Fibonacci, and more generally, episturmian words.

**Definition 7 ([8,16]).** An infinite string  $\mathbf{w}$  is episturmian if it has at most one right-maximal substring of each length and its set of substrings is closed under reversal, that is,  $\mathcal{F}_{\mathbf{w}} = \mathcal{F}_{\mathbf{w}}^R$ . It is standard episturmian (or epistandard) if, in addition, all the right-maximal substrings of  $\mathbf{w}$  are of the form  $\mathbf{w}[1..i]^R$  with  $i \geq 0$ , i.e., they are the reverse of some prefix of  $\mathbf{w}$ .

**Lemma 10.** Let  $\mathbf{w} \in \Sigma^\omega$  be an episturmian word with  $\sigma \geq 2$ . Then,  $\text{sre}(\mathbf{w}[i..j]) \leq \sigma$  for  $i, j \geq 0$ .

*Proof.* Let  $\mathbf{w}$  be an epistandard word. The right-extensions  $x_1, x_2, \dots$  ending with  $a \in \Sigma$  form a *suffix-chain* where each  $x_i$  is a suffix of  $x_{i+1}$ . There is one of those suffix-chains for each character  $a \in \Sigma$ .

Let  $\mathbf{w}$  be episturmian but not necessarily epistandard. There exists some epistandard word  $\mathbf{s}$  with the same set of substrings, i.e.,  $\mathcal{F}_{\mathbf{w}} = \mathcal{F}_{\mathbf{s}}$  [8]. Therefore, for any episturmian word  $\mathbf{w}$ , there exist exactly  $\sigma$  suffix-chains of right-extensions.

When considering substrings of  $\mathbf{w}$ , the super-maximal right-extension in  $\mathbf{w}[i..j]$  ending with  $a \in \Sigma$  is the longest right-extension of  $\mathbf{w}$  ending with  $a$  that remains a right-extension in  $\mathbf{w}[i..j]$ . It follows that for any substring  $\mathbf{w}[i..j]$  of any episturmian word  $\mathbf{w}$ ,  $\text{sre} \leq \sigma$ .  $\square$

Combining this result with Corollary 1, we obtain the following bound.

**Corollary 3.** *For any episturmian word  $w \in \Sigma^\omega$  it holds  $\chi(w[i..j]) \leq \sigma + 2$ .*

The next lemma precisely characterizes the suffixient sets of Fibonacci words, a particular case of epistandard words that will be useful to relate  $\chi$  with  $v$ .

**Definition 8.** *Let  $F_1 = \mathbf{b}$ ,  $F_2 = \mathbf{a}$ , and  $F_k = F_{k-1}F_{k-2}$  for  $k \geq 3$  be the Fibonacci family of strings. Their lengths,  $f_k = |F_k|$ , form the Fibonacci sequence.*

**Lemma 11.** *Every Fibonacci word  $F_k\$$  has a suffixient set of size at most 4. For  $k \geq 6$ , the only smallest suffixient sets for  $F_k\$$  are  $\{f_k+1, f_k-1, f_{k-1}-1, p\}$ , where  $p \in \{f_{k-2}+1, 2f_{k-2}+1\}$ .*

*Proof.* The upper bound of 4 stems directly from Corollary 3, because the infinite Fibonacci word is binary epistandard. For  $k \geq 3$ , there exist strings  $H_k$  such that  $F_k = F_{k-1}F_{k-2} = H_kcd$  and  $F_{k-2}F_{k-1} = H_kdc$ , for  $cd = \mathbf{ab}$  or  $cd = \mathbf{ba}$  depending on the parity of  $k$  [22]. Let us call  $F'_k = H_kdc = F_{k-2}F_{k-1}$ , that is,  $F_k$  with the last two letters exchanged; thus  $F_k = F_{k-1}F_{k-2} = F_{k-2}F'_k$ .

Note that  $F_{k-1} = H_{k-1}dc$  prefixes  $F_k$ . On the other hand, we can write  $F_k = F_{k-1}F_{k-2} = F_{k-2}F_{k-3}F_{k-2} = F_{k-2}F'_{k-1} = F_{k-2}H_{k-1}cd$ . Therefore, string  $H_{k-1}$  is right-maximal in  $F_k$ . Its extensions,  $H_{k-1}d$  and  $H_{k-1}c$ , are super-maximal because there are no other occurrences of  $H_{k-1}$  in  $F_k$ : (i)  $H_{k-1}$  cannot occur starting at positions  $f_{k-2}+2$  or  $f_{k-2}+3$  because it occurs at  $f_{k-2}+1$ , so  $H_{k-1}$  should match itself with an offset of 1 or 2, which is impossible because it prefixes  $F_{k-1}$  and all  $F_{k-1}$  for  $k-1 \geq 5$  start with  $\mathbf{abaab}$ ; (ii)  $H_{k-1}$  cannot occur starting at positions 2 to  $f_{k-2}$  because its prefix  $F_{k-2}$  should occur inside the prefix  $F_{k-2}F_{k-2}$  of  $F_k = F_{k-2}F'_{k-1} = F_{k-2}F_{k-2}F'_{k-3}$ , and so  $F_{k-2}$  should equal a rotation of itself, which is impossible [7, Cor. 3.2]. The two positions following  $H_{k-1}$ ,  $f_{k-1}-1$  and  $f_k-1$ , then appear in any suffixient set.

On the other hand,  $F_{k-2}$  is followed by  $\$$  in  $F_k\$$ , and it also prefixes  $F_k = F_{k-2}F'_{k-1}$ , therefore  $F_{k-2}$  is right-maximal. The first occurrence is preceded by  $F_{k-1}$ , and hence by  $c$ , and the second by no symbol.  $F_{k-2}$  also occurs in  $F_k$  at position  $f_{k-2}+1$ , as seen above, preceded by  $F_{k-2}$  and thus by  $d$ . There are no other occurrences of  $F_{k-2}$  in  $F_k$  because (i) it cannot occur starting at positions 2 to  $f_{k-2}$  by the same reason as point (ii) of the previous paragraph; (ii) it cannot appear starting at positions  $f_{k-2}+2$  to  $f_{k-1}-2$  because  $F_k = F_{k-2}F_{k-2}F'_{k-3}$  and  $F'_{k-3}[1, f_{k-3}-2] = F_{k-3}[1, f_{k-3}-2] = F_{k-2}[1..f_{k-3}-2]$ , thus such an occurrence would also match a rotation of  $F_{k-2}$ , which is impossible as noted above; (iii) it cannot appear starting at positions  $f_{k-1}-1$  or  $f_{k-1}$  because, since it matches at position  $f_{k-1}+1$ ,  $F_{k-2}$  would match itself with an offset of 1 or 2, which is impossible as noted in point (i) of the previous paragraph. The right-extensions of  $F_{k-2}$  are then super-maximal. The one followed by  $\$$  occurs ending at position  $f_k+1$ . The other two are followed by  $\mathbf{a}$  because they are followed by  $F_{k-2}$  and by  $F'_{k-3}$  and all  $F_k$  for  $k \geq 2$  start with  $\mathbf{a}$ . We can then choose either ending position for a suffixient set,  $f_{k-2}+1$  or  $2f_{k-2}+1$ .  $\square$

**Corollary 4.** *There exist string families where  $\chi = o(v)$ .*

*Proof.* It follows from Lemma 11 and the fact that  $v = \Omega(\log n)$  on the odd Fibonacci words [28, Thm. 28].  $\square$

### 5.3 Uncomparability of $\chi$ with Copy-Paste Measures

Finally, we show that  $\chi$  is incomparable with most copy-paste measures. This follows from  $\chi$  being  $\Theta(n)$  on de Bruijn sequences and  $O(1)$  on Fibonacci strings. Because  $g = O(n/\log n)$  on de Bruijn sequences [28] and by Lemma 5, we have:

**Corollary 5.** *There exists a string family with  $\chi = \Omega(g \log n)$ .*

This result is particularly relevant because all the copy-paste based measures  $\mu$ , with the exception of  $z_e$ , are  $O(g)$ . Corollary 5 then implies  $\mu = o(\chi)$  on de Bruijn sequences for all these measures  $\mu$ .

While it has been said that  $z_e = O(n/\log n)$  on binary sequences as well [19], this referred to the version that adds to each phrase the next nonmatching character. Because  $z_e$  is not an optimal parse, it is not obvious that this also holds for the version studied later in the literature, which does not add the next character. We then prove next that  $z_e = o(\chi)$  holds on de Bruijn words.

**Lemma 12.** *There exists a string family with  $\chi = \Omega\left(z_e \frac{\log n \log \log \log n}{(\log \log n)^2}\right)$ .*

*Proof.* It always holds that  $z_e = O\left(z \frac{\log^2(n/z)}{\log \log(n/z)}\right)$  [13]. In de Bruijn sequences it holds that  $z = \Theta(n/\log n)$ , so  $n/z = \Theta(\log n)$ . Therefore,  $z_e = O\left(z \frac{(\log \log n)^2}{\log \log \log n}\right)$ , and replacing  $z = \Theta(n/\log n)$  we get  $z_e = O\left(n \frac{(\log \log n)^2}{\log n \log \log \log n}\right)$ . By Lemma 5, this yields  $\chi = \Omega\left(z_e \frac{\log n \log \log \log n}{(\log \log n)^2}\right) = \omega(z_e)$  on de Bruijn sequences.  $\square$

**Corollary 6.** *The measure  $\chi$  is uncomparable to  $\mu \in \{z, z_{no}, z_e, z_{end}, v, g, g_{rl}, c\}$ .*

*Proof.* From Corollary 5 and Lemma 12, and that  $z, z_{no}, z_{end}, v, g_{rl}$  and  $c$  are always  $O(g)$ , it follows that there are string families where  $\mu = o(\chi)$ , for any  $\mu \in \{z, z_{no}, z_e, z_{end}, v, g, g_{rl}, c\}$ . On the other hand, from Lemma 11 and Corollary 4, and that  $c = \Omega(\log n)$  on Fibonacci words [28, Thm. 32] and  $c = O(\mu)$  for any  $\mu \in \{z, z_{no}, z_e, z_{end}, g_{rl}, g\}$  [28, Thm. 30], it follows that there are string families where  $\chi = o(\mu)$ , for any  $\mu \in \{z, z_{no}, z_e, z_{end}, v, g, g_{rl}, c\}$ .  $\square$

## 6 Conclusions and Open Questions

We have contributed to the understanding of  $\chi$  as a new measure of repetitiveness, better finding its place among more studied ones. Figure 1 shows the (now) known relations around  $\chi$  (cf. [27]).

There are still many interesting open questions about  $\chi$ . One of the most important is whether  $\chi$  is reachable. Proving  $b = O(\chi)$  would settle this question on the affirmative, and at the same time give the first copy-paste measure that is comparable with  $\chi$ . We conjecture, instead, that  $\chi$  is not reachable, proving which would imply that  $\gamma$  is also unreachable, a long-time open question.

One consequence of Corollary 5 is that  $\chi \notin O(g \log^k(n/g))$  for any  $k > 0$ . It could be the case, though, that  $\chi = O(\delta \log n)$ , because the separation of  $\chi$  and  $\delta$  on de Bruijn sequences is a  $\Theta(\log n)$  factor.

Regarding edit operations, it seems that  $\mathbf{sre}(w')/\mathbf{sre}(w)$  is  $O(1)$  for all the string operations we considered. Showing a multiplicative constant for insertion would imply the existence of a constant for rotation and vice versa. It is also open whether  $r = O(\chi \log \chi)$ . If this were true —and provided that  $\chi$  has  $O(1)$  multiplicative sensitivity to string operations— it would imply that  $r$  has  $O(\log n)$  multiplicative sensitivity to these operations, making the already known lower bounds on multiplicative sensitivity [1,14,15] tight. If the conjecture were false, then  $\chi$  could be considerably smaller than  $r$  in some string families.

### Acknowledgements

We thank Davide Cenzato, Nicola Prezza, and Francisco Olivares for their code to compute smallest suffixient sets <https://github.com/regindex/suffixient> [5], which was helpful to propose and discard hypotheses on the behavior of  $\chi$ , and for useful discussions on suffixient sets.

### Disclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

### Funding

G.N. and C.U. were partially funded by Basal Funds FB0001 and AFB240001, ANID, Chile; and FONDECYT Project 1-230755, ANID, Chile.

G.R. was partially funded by the MUR PRIN Project “PINC, Pangenome INformatiCs: from Theory to Applications” (Grant No. 2022YRB97K), funded by Next Generation EU PNRR M4 C2, Inv. 1.1 and by the INdAM - GNCS Project CUP\_E53C24001950001.

C.U. was partially funded by ANID-Subdirección de Capital Humano/Doctorado Nacional/2021-21210580, ANID, Chile; and NIC Chile Doctoral Scholarship, NIC, Chile.

### References

1. Akagi, T., Funakoshi, M., Inenaga, S.: Sensitivity of string compressors and repetitiveness measures. *Information and Computation* **291**, 104999 (2023). <https://doi.org/10.1016/j.ic.2022.104999>
2. Bruijn, de, N.: A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam* **49**(7), 758–764 (1946)
3. Burrows, M., Wheeler, D.: A block sorting lossless data compression algorithm. Tech. Rep. 124, Digital Equipment Corporation (1994)

4. Cenzato, D., Depuydt, L., Gagie, T., Kim, S.H., Manzini, G., Olivares, F., Prezza, N.: Suffixient arrays: a new efficient suffix array compression technique. *CoRR* **2407.18753** (2025). <https://doi.org/10.48550/arXiv.2407.18753>
5. Cenzato, D., Olivares, F., Prezza, N.: On computing the smallest suffixient set. In: *Proc. 31st International Symposium on String Processing and Information Retrieval (SPIRE 2024)*. *Lecture Notes in Computer Science*, vol. 14899, pp. 73–87. Springer (2024). [https://doi.org/10.1007/978-3-031-72200-4\\_6](https://doi.org/10.1007/978-3-031-72200-4_6)
6. Depuydt, L., Gagie, T., Langmead, B., Manzini, G., Prezza, N.: Suffixient sets. *CoRR* **2312.01359** (2023). <https://doi.org/10.48550/arXiv.2312.01359>
7. Droubay, X.: Palindromes in the Fibonacci word. *Information Processing Letters* **55**(4), 217–221 (1995). [https://doi.org/10.1016/0020-0190\(95\)00080-V](https://doi.org/10.1016/0020-0190(95)00080-V)
8. Droubay, X., Justin, J., Pirillo, G.: Episturmian words and some constructions of de Luca and Rauzy. *Theoretical Computer Science* **255**(1), 539–553 (2001). [https://doi.org/10.1016/S0304-3975\(99\)00320-5](https://doi.org/10.1016/S0304-3975(99)00320-5)
9. Fici, G., Romana, G., Sciortino, M., Urbina, C.: On the impact of morphisms on BWT-runs. In: *Proc. 34th Annual Symposium on Combinatorial Pattern Matching (CPM 2023)*. *Leibniz International Proceedings in Informatics*, vol. 259, pp. 10:1–10:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023). <https://doi.org/10.4230/LIPIcs.CPM.2023.10>
10. Fici, G., Romana, G., Sciortino, M., Urbina, C.: Morphisms and BWT-run sensitivity. In: *Proc. 50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*. To appear (2025)
11. Fredricksen, H.: A survey of full length nonlinear shift register cycle algorithms. *SIAM Review* **24**(2), 195–221 (1982). <https://doi.org/10.1137/1024041>
12. Gabric, D., Sawada, J., Williams, A., Wong, D.: A framework for constructing de Bruijn sequences via simple successor rules. *Discrete Mathematics* **341**(11), 2977–2987 (2018). <https://doi.org/10.1016/j.disc.2018.07.010>
13. Gawrychowski, P., Kosche, M., Manea, F.: On the number of factors in the LZ-end factorization. In: *Proc. 30th International Symposium on String Processing and Information Retrieval (SPIRE 2023)*. *Lecture Notes in Computer Science*, vol. 14240, pp. 253–259. Springer (2023). [https://doi.org/10.1007/978-3-031-43980-3\\_20](https://doi.org/10.1007/978-3-031-43980-3_20)
14. Giuliani, S., Inenaga, S., Lipták, Z., Prezza, N., Sciortino, M., Toffanello, A.: Novel results on the number of runs of the Burrows-Wheeler Transform. In: *Proc. 47th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2021)*. *Lecture Notes in Computer Science*, vol. 12607, pp. 249–262. Springer (2021). [https://doi.org/10.1007/978-3-030-67731-2\\_18](https://doi.org/10.1007/978-3-030-67731-2_18)
15. Giuliani, S., Inenaga, S., Lipták, Z., Romana, G., Sciortino, M., Urbina, C.: Bit catastrophes for the Burrows-Wheeler transform. *Theory of Computing Systems* **69**(2), 19 (2025). <https://doi.org/10.1007/s00224-024-10212-9>
16. Glen, A., Justin, J.: Episturmian words: a survey. *RAIRO - Theoretical Informatics and Applications* **43**(3), 403–442 (2009). <https://doi.org/10.1051/ita/2009003>
17. Kempa, D., Kociumaka, T.: Resolution of the Burrows-Wheeler transform conjecture. *Communications of the ACM* **65**(6), 91–98 (2022). <https://doi.org/10.1145/3531445>
18. Kempa, D., Prezza, N.: At the roots of dictionary compression: String attractors. In: *Proc. 50th Annual ACM Symposium on the Theory of Computing (STOC 2018)*. pp. 827–840. ACM (2018). <https://doi.org/10.1145/3188745.3188814>
19. Kreft, S., Navarro, G.: On compressing and indexing repetitive sequences. *Theoretical Computer Science* **483**, 115–133 (2013). <https://doi.org/10.1016/j.tcs.2012.02.006>

20. Lempel, A., Ziv, J.: On the complexity of finite sequences. *IEEE Transactions on Information Theory* **22**(1), 75–81 (1976). <https://doi.org/10.1109/TIT.1976.1055501>
21. Lothaire, M.: *Algebraic Combinatorics on Words*. *Encyclopedia of Mathematics and its Applications*, Cambridge University Press, New York, NY, USA (2002). <https://doi.org/10.1017/CBO9781107326019>
22. de Luca, A.: A combinatorial property of the Fibonacci words. *Information Processing Letters* **12**(4), 193–195 (1981). [https://doi.org/10.1016/0020-0190\(81\)90099-5](https://doi.org/10.1016/0020-0190(81)90099-5)
23. Manber, U., Myers, E.W.: Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing* **22**(5), 935–948 (1993). <https://doi.org/10.1137/0222058>
24. Mantaci, S., Restivo, A., Romana, G., Rosone, G., Sciortino, M.: A combinatorial view on string attractors. *Theoretical Computer Science* **850**, 236–248 (2021). <https://doi.org/10.1016/j.tcs.2020.11.006>
25. Navarro, G.: Indexing highly repetitive string collections, part I: Repetitiveness measures. *ACM Computing Surveys* **54**(2), article 29 (2021). <https://doi.org/10.1145/3434399>
26. Navarro, G.: Indexing highly repetitive string collections, part II: Compressed indexes. *ACM Computing Surveys* **54**(2), article 26 (2021). <https://doi.org/10.1145/3432999>
27. Navarro, G.: Indexing highly repetitive string collections. *CoRR* **2004.02781** (2022). <https://doi.org/10.48550/arXiv.2004.02781>
28. Navarro, G., Ochoa, C., Prezza, N.: On the approximation ratio of ordered parsings. *IEEE Transactions on Information Theory* **67**(2), 1008–1026 (2021). <https://doi.org/10.1109/TIT.2020.3042746>
29. Navarro, G., Olivares, F., Urbina, C.: Generalized straight-line programs. *Acta Informatica* **62**(1), 14 (2025). <https://doi.org/10.1007/s00236-025-00481-3>
30. Navarro, G., Urbina, C.: Repetitiveness measures based on string morphisms. *Theoretical Computer Science* **1043**, 115259 (2025). <https://doi.org/10.1016/j.tcs.2025.115259>
31. Storer, J.A., Szymanski, T.G.: Data compression via textual substitution. *Journal of the ACM* **29**(4), 928–951 (1982). <https://doi.org/10.1145/322344.322346>