Can Formal Languages help Pangenomics to represent and analyze multiple genomes? *

Paola Bonizzoni[1], Clelia De Felice[2], Yuri Pirola[1], Raffaella Rizzi[1], Rocco Zaccagnino[2], and Rosalba Zizza[2]

[1] Dip. di Informatica, Sistemistica e Comunicazione, University of Milano-Bicocca, viale Sarca 336, 20126 Milan, Italy
{paola.bonizzoni,yuri.pirola,raffaella.rizzi}@unimib.it
[2] Dip. di Informatica, University of Salerno, via Giovanni Paolo II 132, 84084 Fisciano, Italy
{rzaccagnino,rzizza,cdefelice}@unisa.it

Abstract. Graph pangenomics is a new emerging field in computational biology that is changing the traditional view of a reference genome from a linear sequence to a new paradigm: a sequence graph (pangenome graph or simply pangenome) that represents the main similarities and differences in multiple evolutionary related genomes. The speed in producing large amounts of genome data, driven by advances in sequencing technologies, is far from the slow progress in developing new methods for constructing and analyzing a pangenome. Most recent advances in the field are still based on notions rooted in established and quite old literature on combinatorics on words, formal languages and space efficient data structures. In this paper we discuss two novel notions that may help in managing and analyzing multiple genomes by addressing a relevant question: how can we summarize sequence similarities and dissimilarities in large sequence data? The first notion is related to variants of the *Lyndon factorization* and allows to represent sequence similarities for a sample of reads, while the second one is that of *sample specific string* as a tool to detect differences in a sample of reads. New perspectives opened by these two notions are discussed.

1 Introduction

The 1000 Genomes Project [16] marks the beginning of new computational approaches to genomic studies involving the use of efficient data structures to represent the high variation rate among multiple genomes. Indeed, a main result of the project has been the characterization of a broad spectrum of genetic variations in the human genome, including the discovery of novel variations in the

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 872539.
The final authenticated version is available online at https://doi.org/10.1007/978-3-031-05578-2_1.

South Asian, African and European populations—thus enhancing the catalogue of variability within the human individuals. In particular, the question "what is an ideal human reference genome?" is becoming the focus of investigations that also involve theoreticians in the computer science community. While the literature in computational biology presents experimental evidence of the advantages of the idea of replacing a linear reference with a pangenome graph [23, 37, 39], still theoretical foundations of computational pangenomics is missing. A recent tutorial introduces the main theoretical background in graph pangenomics [1]. It is interesting to note that formal language theory has again played a crucial role in suggesting novel approaches to this new emerging field. The first main representation of a graph pangenome is based on building a prefix language from the interpretation of the graph as an automaton [33], while *Wheeler graphs* [22] establish an interesting connection between regular languages and compressed data structures which are fundamental in the indexing of pangenomes. Language theoretic notions that have been recently rediscovered in Bioinformatics are those of Lyndon words and of the Lyndon factorization of a word [15, 21, 32]. Indeed, these well-known notions intervene in a bijective transformation [29] alternative to the Burrows-Wheeler Transform for compressing sequences and in new measures of similarities between sequences [7]. The investigation of sequence similarity and dissimilarity measures is a crucial topic in Bioinformatics for comparing sequences. For example, sequence alignment is the oldest standard procedure performed to measure the distance between sequences. However, the search for alignment-free approaches to sequence comparison is the focus of deep investigations in the framework of pangenomics, since there is the need to cope with the high computational cost of the alignment and have fast approaches to compute genetic variations in a pangenome [1]. In this direction, a possible alignment-free approach may consist in applying mathematical transformations on sequences that lead easily to a fast sequence comparison. In particular, summarizing sequences by alternative representations is becoming a new paradigm for facing the huge amount of sequencing data. Factorizing a word is intuitively a way to give an alternative representation of it: thus, a main question is whether there exists a way to factorize sequences so that it may lead to a more compact representation to detect shared regions between sequences. This work is focused on the Lyndon factorization as a factorization preserving similarities among sequences. Since being able to detect dissimilarities is also important in sequence comparison, here we also present a novel notion aiming at discovering differences among similar sequences. This is the notion of *sample specific string* (SFS) [27]. We show applications of both notions in facing problems motivated by computational pangenomics [13, 20].

This paper is structured as follows. After introducing preliminaries on sequences, Lyndon words and Lyndon factorization, we survey some main theoretical results on Lyndon-based factorizations motivated by Bioinformatics applications. Then, we discuss preliminary results on their application. In Section 4 we discuss the theoretical background of the notion of sample specific strings and then, we

present its application in structural variant detection. We conclude with some open problems related to Lyndon-based factorizations.

2 Preliminaries

Throughout this paper we follow [31] for the notations. Let $w = a_1 \cdots a_m$ be a string (or word) over a finite alphabet Σ. The empty word is denoted by 1. The *length* of w (that is, the number m of its characters) will be denoted by $|w|$. A word $x \in \Sigma^*$ is a *factor* of $w \in \Sigma^*$ if there are $u_1, u_2 \in \Sigma^*$ such that $w = u_1 x u_2$. If $u_1 = 1$ (resp. $u_2 = 1$), then x is a *prefix* (resp. *suffix*) of w. A factor (resp. prefix, suffix) x of w is *proper* if $x \neq w$. We recall that, given a nonempty word w, *a border* of w is a word which is both a proper prefix and a suffix of w. The longest border is also called *the border* of w. A word $w \in \Sigma^+$ is bordered if it has a nonempty border. Otherwise, w is *unbordered*. A nonempty word w is *primitive* if $w = x^k$ implies $k = 1$. An unbordered word is primitive. Given $w, w' \in \Sigma^*$, we denote by $w < w'$ (resp. $w \leq w'$) if w is lexicographically smaller than w' (resp. smaller than or equal to w'). Furthermore, for two nonempty words w, w', we write $w \ll w'$ if $w < w'$ and additionally w is not a proper prefix of w' [4]. We recall that a *factorization* of a string w is a sequence $F(w) = (f_1, f_2, \ldots, f_n)$ of factors such that $w = f_1 f_2 \cdots f_n$.

In [6] a numeric representation of a factorization of a string is defined, named the *fingerprint* of w with respect to $F(w)$, i.e., the sequence $\mathcal{L}(w) = (|f_1|, |f_2|, \ldots, |f_n|)$ of the lengths of the factors of $F(w)$. In addition, a *k-finger* is a k-mer of $\mathcal{L}(w)$, that is, any substring $(l_i, l_{i+1}, \ldots, l_{i+k-1})$ composed of k consecutive elements of $\mathcal{L}(w)$.

In this framework, we consider strings over a the DNA alphabet and they will be simply called sequences, meaning to represent genomes or fragments of genome sequences. For preliminaries to computational pangenomics and some basic notions, we address the reader to a recent tutorial [1].

3 Lyndon words and Lyndon-based factorization

The Lyndon Factorization CFL. In order to obtain read fingerprints, in [7] some special kinds of factorizations are proposed, named *Lyndon-based factorizations*, since they are defined starting from the well-known *Lyndon factorization* of a string w [32]. Each string w can be uniquely factorized into a non-increasing product (w.r.t. the lexicographic order) of *Lyndon words* [32]. A Lyndon word is a string which is strictly smaller than any of its nonempty proper suffixes. Lyndon words are primitive and unbordered. For example, suppose that $\Sigma = \{a, c, g, t\}$ and $a < c < g < t$ (in next examples, we always suppose this ordering on the alphabet). Thus, *accgctct* is a Lyndon word, whereas *cac* is not a Lyndon word,

Formally, given a string w, its Lyndon factorization $\mathsf{CFL}(w)$ is a sequence $\mathsf{CFL}(w) = (f_1, f_2, \ldots, f_n)$ of words such that $f_1 \geq f_2 \geq \cdots \geq f_n$ and each f_i is a Lyndon word. For example, given $w_1 = gcatcaccgctctacagaac$, we have that $\mathsf{CFL}(w_1) = (g, c, atc, accgctct, acag, aac)$. The notation CFL is due to the fact

that stating the uniqueness of this factorization is usually credited to Chen, Fox and Lyndon [15]. We recall that CFL can be computed in linear time and constant space [21].

The notion of Lyndon words has been shown to be useful in theoretical applications, such as the well-known proof of the *Runs Theorem* [2], as well as in string compression analysis. Furthermore, the Lyndon factorization has recently revealed to be a useful tool also in investigating queries on suffixes of a word and sorting such suffixes with strong potentialities for string comparison that have not been completely explored and understood. Relations between Lyndon words and the Burrows-Wheeler Transform (BWT) have also been discovered first in [18,34] and, more recently, in [3,28,29]. A connection is found between the Lyndon factorization CFL and the Lempel-Ziv (LZ) factorization [26], where it is shown that in general the size of the LZ factorization is larger than the size of the Lyndon factorization, and in any case the size of the Lyndon factorization cannot be larger than a factor of 2 with respect to the size of LZ. This result has been further extended in [40] to overlapping LZ factorizations.

Conservation Property of CFL. In [10] a new property of the Lyndon factorization, named *Conservation Property* [6,7,13], has been proved, which is crucial in our framework, and here reported. More precisely, let $\mathsf{CFL}(w) = (\ell_1, \ell_2, \ldots, \ell_n)$. We firstly recall that x is a *simple* factor of w if, for each occurrence of x as a factor of w, there is j, with $1 \leq j \leq n$, such that x is a factor of ℓ_j. So, let $x = \ell_i'' \ell_{i+1} \cdots \ell_{j-1} \ell_j'$ be a non simple factor of w, for some indexes i, j with $1 \leq i < j \leq n$, and $\ell_i = \ell_i' \ell_i''$, $\ell_j = \ell_j' \ell_j''$.

The above-mentioned Conservation Property is stated below and it compares the Lyndon factorization of w and that of its non-simple factors.

Lemma 1. *[9, 10] Let $w \in \Sigma^+$ be a word and let $\mathsf{CFL}(w) = (\ell_1, \ldots, \ell_n)$ be its Lyndon factorization. For any i, j, with $1 \leq i \leq j \leq n$, one has $\mathsf{CFL}(\ell_i \ell_{i+1} \cdots \ell_j) = (\ell_i, \ell_{i+1}, \ldots, \ell_j)$. In addition, let x be a non-simple factor of w such that x is not a concatenation of consecutive factors of $\mathsf{CFL}(w)$ and let $\ell_i'', \ell_{i+1}, \ldots, \ell_{j-1}, \ell_j'$ be such that $x = \ell_i'' \ell_{i+1} \cdots \ell_{j-1} \ell_j'$, with $1 \leq i < j \leq n$.*

Let $\mathsf{CFL}(\ell_i'') = (m_1, \ldots, m_h)$ and $\mathsf{CFL}(\ell_j') = (v_1, \ldots, v_t)$. We have

$$\mathsf{CFL}(x) = (m_1, \ldots, m_h, \ell_{i+1}, \ldots, \ell_{j-1}, v_1, \ldots, v_t)$$

where it is understood that if $\ell_i'' = 1$ (resp. $\ell_j' = 1$), then the first h terms (resp. last t terms) in $\mathsf{CFL}(x)$ vanish.

According to Lemma 1, given two strings w and w' sharing a common overlap x, under some hypothesis, there exist factors that are in common between $\mathsf{CFL}(w)$ and $\mathsf{CFL}(w')$. Thus w and w' will have fingerprints sharing k-fingers for a suitable size k. For example, consider again $w_1 = gcatcaccgctctacagaac$ and let $w_2 = ccaccgctctacagaagcatc$. Then, $\mathsf{CFL}(w_1) = (g, c, atc, accgctct, acag, aac)$ and we have that $\mathsf{CFL}(w_2) = (c, c, accgctct, acag, aagcatc)$. Hence, we have $\mathcal{L}(w_1) = (1, 1, 3, 8, 4, 3)$ and $\mathcal{L}(w_2) = (1, 1, 8, 4, 7)$. The two common consecutive elements $(8, 4)$ are related to the same factors in the two words (8 is related to

accgctct and 4 is related to *acag*) and capture the common substring *accgctctacag* given by their concatenation.

Even though the hypothesis that x is not simple with respect to $\mathsf{CFL}(w)$ cannot be dropped (see [6]), it is worthy of note that in real data this hypothesis is always satisfied. Such an interesting property suggests the possibility of using directly k-fingers as features. Indeed, in [6] it is presented an approach in which k-fingers are used for classifying sequencing reads (Section 3.1).

Canonical Inverse Lyndon factorization ICFL**.** The *Canonical Inverse Lyndon factorization* $\mathsf{ICFL}(w) = (f_1, f_2, \ldots, f_n)$ has been introduced in [8] as a factorization of w such that $f_1 \ll f_2 \ll \cdots \ll f_n$ and each f_i is an *inverse Lyndon word*, that is, each nonempty proper suffix of f_i is strictly smaller than f_i [8]. For example, *cac*, *tcaccgc* are inverse Lyndon words. Let us consider again $w_1 = gcatcaccgctctacagaac$. We have that $\mathsf{ICFL}(w_1) = (gca, tcaccgc, tctacagaac)$. Observe that, differently from Lyndon words, inverse Lyndon words may be bordered. Furthermore, this factorization is also unique and can be computed in linear time [8].

What is the motivation of introducing a new factorization? In [10] two main results are proved: *(i)* an upper bound on the length of the longest common prefix of two factors of w starting from different positions on w is provided, and *(ii)* a relation among sorting of global suffixes, i.e., suffixes of the word w, and sorting of local suffixes, i.e., suffixes of the products of factors in $\mathsf{ICFL}(w)$ is given. The latter result is the counterpart for $\mathsf{ICFL}(w)$ of the compatibility property, proved in [35] for the Lyndon factorization. However, *(ii)* is in some sense stronger than that one in [35], as we explain below. Indeed, as a preliminary result, in [10] it is proved that that the longest common prefix between f_i and f_{i+1} is shorter than the border of f_i, when w is not an inverse Lyndon word. This result is obtained thanks to the *grouping* property of ICFL proved in [8]: given a word w, the factors in $\mathsf{ICFL}(w)$ are obtained by grouping together consecutive factors of the anti-Lyndon factorization of w that form a non-increasing chain for the prefix order (the anti-Lyndon factorization of w is the Lyndon factorization w.r.t. the inverse lexicographic order).

In this framework, a natural question is whether and how the longest common extensions for arbitrary positions on w are related to the size of the factors in $\mathsf{ICFL}(w)$. It is proved that there are relations between the length of the longest common prefix $\mathrm{lcp}(x, y)$ of two factors x, y of a word w starting from different positions on w and the maximum length $\mathcal{M}$ of two consecutive factors of the inverse Lyndon factorization of w. More precisely, $\mathcal{M}$ is an upper bound on the length of $\mathrm{lcp}(x, y)$. Thus, this result is in some sense stronger than the compatibility property, proved in [35] for the Lyndon factorization and in [10] for the inverse Lyndon factorization. Roughly, the compatibility property allows us to extend to the suffixes of the whole word the mutual order between suffixes of the concatenation of (inverse) Lyndon factors.

3.1 Some applications: representing and querying read sequences

Sequencing technologies produce the main input data for a vast majority of algorithms in Bioinformatics. For example, the only way to get the whole sequence of the genome of a single individual is to produce (by sequencing) fragmented multiple copies of the genome sequence (called *reads*), that are computationally assembled into the original sequence. The extraordinary improvements in the sequencing technologies has allowed to obtain long enough fragments w.r.t. to the original massive sequencing consisting of reads of an average length of around 100 base pairs. In this section we touch upon two applications of the notion of *fingerprint*, presented in the previous sections, related to two traditionally difficult Bioinformatics tasks: genome assembly and transcript read classification. Indeed, read fingerprints provide a compact representation of the reads and, thanks to the Conservation Property, they are effective in preserving sequence similarities. In fact, the k-fingers (sub-pieces of a fingerprint) are able to capture the similarity regions between two reads in a more flexible way with respect to the k-mers of a sequence: the length k of a k-mer is fixed, whereas the length of the read substring, undergoing a k-finger, is variable. Furthermore, fingerprints are numerical sequences shorter than the represented character sequences and we also expect that they are resilient to errors occurring in the reads (especially in long reads). The first application is related to genome assembly based on the use of an overlap graph which is constructed by detecting the overlaps between genomic reads [11, 12]. When dealing with long reads this task is further complicated by the length of the reads and the high sequencing error rate. In [13] a novel alignment-free algorithm for discovering the overlaps in a set of noisy long reads is presented, which exploits the fingerprints of the reads. Indeed, the k-fingers provide anchors for computing the overlaps between reads. The algorithm takes as input a set S of genomic reads and, after factorizing them, builds a hash table of the k-fingers by performing a linear scanning of the fingerprints. Next, the hash table is used in order to compute in $O(LN)$ time the common regions between each read s and the reads previously processed, assuming that the read length is L and N is the maximum number of occurrences of a unique (that is, occurring once) k-finger of s in the reads considered at the previous iterations. At the end, the algorithm obtains the read overlaps from all the detected common regions. Observe that comparing reads in a reference-free framework often requires a pairwise comparison and is computationally demanding (refer for example to the problem of the identification of the relationships between metagenomic reads [25]). The second application of the read fingerprints is related to the problem of assigning transcriptomic reads (that is, reads sequenced from gene transcripts of RNA-Seq reads) to their origin gene. In [6] fingerprints are used as a machine-interpretable representation of sequencing data in order to define an effective feature embedding method for assigning RNA-Seq reads to the origin gene. Indeed, a fingerprint (and the sequence of k-fingers) is used to produce an embedded representation of the read. Moreover, the machine learning classifier proposed in [6] was also extended for detecting chimeric RNA-Seq reads, which is a subtask of gene-fusion finding methods [19,30,38]. In fact, the chimeric reads

detection problem can be seen as a variant of the read-gene classification problem since it requires to assign a chimeric read to two genes (instead of a single gene), which have been fused after genomic rearrangement.

4 Sample Specific strings and structural variations in human genome

A classical example of how combinatorics on words is helping comparative genomics to analyze sequences, is given by the notion of *minimal absent word* [5]: this is a word absent from y whose all proper factors occur in y. It has several applications in Bioinformatics [14,36]. Here we consider a slightly different variant based on the idea of considering minimal words that are absent in a sequence but present in another sequence: we call them *specific strings*. Recently, in [27] the notion of sample specific string has been proposed to detect signatures of variations between a reference genome R and a sample T of reads from a target individual. A sample of reads is the typical output of the sequencing of an individual and consists of a collection of strings or reads.

Let us formally recall the notion of specific strings introduced in [27].

Definition 1. *Let R (reference) and T (target) be two strings over a finite alphabet. Then a factor s of T is a* T-specific string *w.r.t. to R (in short specific string) if the following properties hold:*

1. *s is not a factor of R,*
2. *any proper factor of s occurs in R.*

Then given a collection of strings $\mathcal{S}$ and a string R, s is a sample specific string for the collection $\mathcal{S}$, SFS in short, if s is a T-specific string for some target T in $\mathcal{S}$. A linear-time algorithm for computing T-specific string that are not overlapping on the input sequence T is given in [27], while an extensive discussion of some algorithmic properties is reported in [27]. SFSs have been proved in [20] to be effective in detecting breakpoints of structural variants (SV) i.e. medium to large size insertions and deletions in a reference genome that are present in a human sample of high quality long reads, (e.g. PACBIO HIFI). Indeed, the main idea behind the notion of SFS is that they may be of variable length w.r.t. fixed length k-mers traditionally used to identify SVs as unique k-mers occurring in a sequence. More precisely, given a substring x of a sequence R, an insertion or deletion inside x it is likely to produce a new string y that does not occur in R. Moreover, the breakpoints of the insertion or deletions (a breakpoint in x is a position of x delimiting the insertion or deletion) are likely to be associated to two factors which may be absent from R. Behind the practical interest in SFSs they are an interesting notion from the theoretical point of view. In particular, we conjecture that the SFSs could provide bounds on the classical edit distance and on the edit distance with moves, a generalization of the edit distance allowing the exchange of blocks, i.e. factors inside the sequence [17].

5 Open problems

The method given in [6] uses representation of reads obtained starting from Lyndon based factorizations. A natural question, faced in the same paper, is whether the corresponding representation produced by its fingerprint or by its k-fingers is unique, a property which is closely related to the collision phenomenon: distinct strings may have have common k-fingers. An open problem is of how the lexicographic ordering of the alphabet may affect the collision phenomenon, The properties described in [6] show that the choice of a specific ordering of the initial alphabet can have a significant impact on the collision phenomenon. However, the problem of understanding if there exists an order that minimizes this phenomenon remains open (and, if exists, which is this order) and future investigations should be devoted to it. It is worth of note that in general, the questions of finding an optimal alphabet ordering for Lyndon factorization (*i.e.*, such that number of Lyndon factors is at most, or at least, n, for a given number n) is hard [24].

As already mentioned in Section 3, it could be interesting to investigate how the bound proved for the longest common prefix between suffixes of factors in ICFL may be used for efficiently sorting suffixes. Furthermore, one challenging question is whether ICFL could be used instead of CFL for defining a new bijective version of the Burrows Wheeler Transform, as done in [29].

References

1. Baaijens, J.A., Bonizzoni, P., Boucher, C., Della Vedova, G., Pirola, Y., Rizzi, R., Sirén, J.: Computational graph pangenomics: a tutorial on data structures and their applications. Natural Computing (2022). https://doi.org/10.1007/s11047-022-09882-6
2. Bannai, H., I, T., Inenaga, S., Nakashima, Y., Takeda, M., Tsuruta, K.: The "Runs" Theorem. SIAM J. Comput. **46**(5), 1501–1514 (2017)
3. Bannai, H., Kärkkäinen, J., Köppl, D., Piatkowski, M.: Indexing the bijective BWT. In: Pisanti, N., Pissis, S.P. (eds.) 30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019, June 18-20, 2019, Pisa, Italy. LIPIcs, vol. 128, pp. 17:1–17:14 (2019)
4. Bannai, H., Tomohiro, I., Inenaga, S., Nakashima, Y., Takeda, M., Tsuruta, K.: A new characterization of maximal repetitions by Lyndon trees. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015. pp. 562–571 (2015)
5. Béal, M., Mignosi, F., Restivo, A.: Minimal forbidden words and symbolic dynamics. In: Puech, C., Reischuk, R. (eds.) STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1046, pp. 555–566. Springer (1996)
6. Bonizzoni, P., Costantini, M., De Felice, C., Petescia, A., Pirola, Y., Previtali, M., Rizzi, R., Stoye, J., Zaccagnino, R., Zizza, R.: Numeric Lyndon-based feature embedding of sequencing reads for machine learning approaches. CoRR **abs/2202.13884** (2022), https://arxiv.org/abs/2202.13884

7. Bonizzoni, P., De Felice, C., Petescia, A., Pirola, Y., Rizzi, R., Stoye, J., Zaccagnino, R., Zizza, R.: Can we replace reads by numeric signatures? Lyndon fingerprints as representations of sequencing reads for machine learning. In: Algorithms for Computational Biology - 8th International Conference, AlCoB 2021, Missoula, MT, USA, June 7-11, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12715, pp. 16–28. Springer (2021)
8. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: Inverse Lyndon words and inverse Lyndon factorizations of words. Adv. Appl. Math. **101**, 281–319 (2018)
9. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: Lyndon words versus inverse Lyndon words: Queries on suffixes and bordered words. In: Language and Automata Theory and Applications - 14th International Conference, LATA 2020, Milan, Italy, March 4-6, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12038, pp. 385–396. Springer (2020)
10. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: On the longest common prefix of suffixes in an inverse Lyndon factorization and other properties. Theor. Comput. Sci. **862**, 24–41 (2021)
11. Bonizzoni, P., Della Vedova, G., Pirola, Y., Previtali, M., Rizzi, R.: An external-memory algorithm for string graph construction. Algorithmica **78**(2), 394–424 (2017)
12. Bonizzoni, P., Della Vedova, G., Pirola, Y., Previtali, M., Rizzi, R.: FSG: Fast String Graph Construction for De Novo Assembly. Journal of Computational Biology **24**(10), 953–968 (2017)
13. Bonizzoni, P., Petescia, A., Pirola, Y., Rizzi, R., Zaccagnino, R., Zizza, R.: Kfinger: Capturing overlaps between long reads by using Lyndon fingerprints. In: IWBBIO Conference, Gran Canaria, Spain, June 27th-30th, 2022, Proceedings. to appear (2021)
14. Chairungsee, S., Crochemore, M.: Using minimal absent words to build phylogeny. Theoretical Computer Science **450**, 109–116 (2012)
15. Chen, K.T., Fox, R.H., Lyndon, R.C.: Free Differential Calculus, IV. The quotient groups of the Lower Central Series. Ann. Math. **68**, 81–95 (1958)
16. Consortium, .G.P., et al.: A global reference for human genetic variation. Nature **526**(7571), 68 (2015)
17. Cormode, G., Muthukrishnan, S.: The string edit distance matching problem with moves. ACM Transactions on Algorithms (TALG) **3**(1), 1–19 (2007)
18. Crochemore, M., Désarménien, J., Perrin, D.: A note on the Burrows-Wheeler transformation. Theoretical Computer Science **332**(1), 567 – 572 (2005)
19. Davidson, N.M., Chen, Y., Ryland, G.L., Blombery, P., Göke, J., Oshlack, A.: JAFFAL: Detecting fusion genes with long read transcriptome sequencing. bioRxiv (2021). https://doi.org/10.1101/2021.04.26.441398
20. Denti, L., Khorsand, P., Bonizzoni, P., Hormozdiari, F., Chikhi, R.: Improved structural variant discovery in hard-to-call regions using sample-specific string detection from accurate long reads. bioRxiv (2022)
21. Duval, J.: Factorizing words over an ordered alphabet. J. Algorithms **4**(4), 363–381 (1983)
22. Gagie, T., Manzini, G., Sirén, J.: Wheeler graphs: A framework for BWT-based data structures. Theor. Comput. Sci. **698**, 67–78 (2017)
23. Garrison, E., Sirén, J., Novak, A.M., et al.: Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nature Biotechnology **36**, 875–879 (2018)

24. Gibney, D., Thankachan, S.V.: Finding an Optimal Alphabet Ordering for Lyndon factorization is Hard. In: 38th International Symposium on Theoretical Aspects of Computer Science (STACS2021). pp. 1–15. Leibniz International Proceedings in Informatics (LIPIcs) (2021)
25. Girotto, S., Pizzi, C., Comin, M.: MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures. Bioinform. **32**(17), 567–575 (2016)
26. Kärkkäinen, J., Kempa, D., Nakashima, Y., Puglisi, S.J., Shur, A.M.: On the Size of Lempel-Ziv and Lyndon Factorizations. In: 34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany. pp. 45:1–45:13 (2017)
27. Khorsand, P., Denti, L., Human Genome Structural Variant, C., Bonizzoni, P., Chikhi, R., Hormozdiari, F.: Comparative genome analysis using sample-specific string detection in accurate long reads. Bioinformatics Advances **1**(1), vbab005 (2021)
28. Köppl, D., Hashimoto, D., Hendrian, D., Shinohara, A.: In-Place Bijective Burrows-Wheeler Transforms. In: 31st Annual Symposium on Combinatorial Pattern Matching (CPM 2020). Leibniz International Proceedings in Informatics (LIPIcs), vol. 161, pp. 21:1–21:15 (2020)
29. Kufleitner, M.: On bijective variants of the Burrows-Wheeler transform. In: Proceedings of the Prague Stringology Conference 2009, Prague, Czech Republic, August 31 - September 2, 2009. pp. 65–79 (2009)
30. Liu, Q., Hu, Y., Stucky, A., Fang, L., Zhong, J.F., Wang, K.: LongGF: computational algorithm and software tool for fast and accurate detection of gene fusions by long-read transcriptome sequencing. BMC Genomics **21**, 793 (2020). https://doi.org/10.1186/s12864-020-07207-4
31. Lothaire, M.: Algebraic Combinatorics on Words, Encyclopedia Math. Appl., vol. 90. Cambridge University Press (1997)
32. Lyndon, R.: On Burnside problem i. Trans. Amer. Math. Soc. **77**, 202–215 (1954)
33. Mäkinen, V., Välimäki, N., Sirén, J.: Indexing graphs for path queries with applications in genome research. IEEE ACM Trans. Comput. Biol. Bioinform. **11**(2), 375–388 (2014)
34. Mantaci, S., Restivo, A., Rosone, G., Sciortino, M.: An extension of the Burrows-Wheeler Transform. Theor. Comput. Sci. **387**(3), 298–312 (2007)
35. Mantaci, S., Restivo, A., Rosone, G., Sciortino, M.: Suffix array and Lyndon factorization of a text. J. Discrete Algorithms **28**, 2–8 (2014)
36. Pinho, A.J., Ferreira, P.J., Garcia, S.P., Rodrigues, J.M.: On finding minimal absent words. BMC bioinformatics **10**(1), 1–11 (2009)
37. Rakocevic, G., Semenyuk, V., Lee, W.P., Spencer, J., Browning, J., Johnson, I.J., Arsenijevic, V., Nadj, J., Ghose, K., Suciu, M.C., et al.: Fast and accurate genomic analyses using genome graphs. Nature genetics **51**(2), 354–362 (2019)
38. Rautiainen, M., Durai, D.A., Chen, Y., Xin, L., Low, H.M., Göke, J., Marschall, T., Schulz, M.H.: AERON: Transcript quantification and gene-fusion detection using long reads. bioRxiv (2020). https://doi.org/10.1101/2020.01.27.921338
39. Sibbesen, J.A., Maretty, L., Krogh, A.: Accurate genotyping across variant classes and lengths using variant graphs. Nature genetics **50**(7), 1054–1059 (2018)
40. Urabe, Y., Kempa, D., Nakashima, Y., Inenaga, S., Bannai, H., Takeda, M.: On the Size of Overlapping Lempel-Ziv and Lyndon Factorizations. In: 30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019, June 18-20, 2019, Pisa, Italy. LIPIcs, vol. 128, pp. 29:1–29:11. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019)