ETH zürich

ETH Library

AlcoR: alignment-free simulation, mapping, and visualization of low-complexity regions in biological data

Journal Article

Author(s):
Silva, Jorge M.; Qi, Weihong; Pinho, Armando J.; Pratas, Diogo

Publication date:
2023

Permanent link:
https://doi.org/https://doi.org/10.3929/ethz-b-000655057

Rights / license:
Creative Commons Attribution 4.0 International

Originally published in:
GigaScience 12(1), https://doi.org/10.1093/gigascience/giad101

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

OXFORD (GIGA)n SCIENCE

GigaScience, 2023, **12**, 1–16
DOI: 10.1093/gigascience/giad101
Technical Note

AlcoR: alignment-free simulation, mapping, and visualization of low-complexity regions in biological data

Jorge M. Silva [1,2], Weihong Qi [3,4], Armando J. Pinho [1,2], and Diogo Pratas [1,2,5,*]

[1]IEETA, Institute of Electronics and Informatics Engineering of Aveiro, and LASI, Intelligent Systems Associate Laboratory, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
[2]Department of Electronics Telecommunications and Informatics, University of Aveiro, Campus Universitario de Santiago, 3810-193, Aveiro, Portugal
[3]Functional Genomics Center Zurich, ETH Zurich and University of Zurich, Winterthurerstrasse, 190, 8057, Zurich, Switzerland
[4]SIB, Swiss Institute of Bioinformatics, 1202, Geneva, Switzerland
[5]Department of Virology, University of Helsinki, Haartmaninkatu, 3, 00014 Helsinki, Finland
*Correspondence address. Diogo Pratas. Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal. E-mail: pratas@ua.pt

Abstract

Background: Low-complexity data analysis is the area that addresses the search and quantification of regions in sequences of elements that contain low-complexity or repetitive elements. For example, these can be tandem repeats, inverted repeats, homopolymer tails, GC-biased regions, similar genes, and hairpins, among many others. Identifying these regions is crucial because of their association with regulatory and structural characteristics. Moreover, their identification provides positional and quantity information where standard assembly methodologies face significant difficulties because of substantial higher depth coverage (mountains), ambiguous read mapping, or where sequencing or reconstruction defects may occur. However, the capability to distinguish low-complexity regions (LCRs) in genomic and proteomic sequences is a challenge that depends on the model's ability to find them automatically. Low-complexity patterns can be implicit through specific or combined sources, such as algorithmic or probabilistic, and recurring to different spatial distances—namely, local, medium, or distant associations.

Findings: This article addresses the challenge of automatically modeling and distinguishing LCRs, providing a new method and tool (AlcoR) for efficient and accurate segmentation and visualization of these regions in genomic and proteomic sequences. The method enables the use of models with different memories, providing the ability to distinguish local from distant low-complexity patterns. The method is reference and alignment free, providing additional methodologies for testing, including a highly flexible simulation method for generating biological sequences (DNA or protein) with different complexity levels, sequence masking, and a visualization tool for automatic computation of the LCR maps into an ideogram style. We provide illustrative demonstrations using synthetic, nearly synthetic, and natural sequences showing the high efficiency and accuracy of AlcoR. As large-scale results, we use AlcoR to unprecedentedly provide a whole-chromosome low-complexity map of a recent complete human genome and the haplotype-resolved chromosome pairs of a heterozygous diploid African cassava cultivar.

Conclusions: The AlcoR method provides the ability of fast sequence characterization through data complexity analysis, ideally for scenarios entangling the presence of new or unknown sequences. AlcoR is implemented in C language using multithreading to increase the computational speed, is flexible for multiple applications, and does not contain external dependencies. The tool accepts any sequence in FASTA format. The source code is freely provided at https://github.com/cobilab/alcor.

Keywords: low-complexity analysis, alignment-free method, data compression, genomes, proteomes, FASTA, simulation

Introduction

With the current unprecedented developments in sequencing techniques and reconstruction methodologies, the availability of complete sequences with high quality and without gaps is steadily increasing [1–3], providing direct applicability to the accurate analysis of these sequences, specifically large-scale data complexity studies. However, the current methodologies are not fully prepared to deal with the scale and characteristics of the reconstructed data, becoming prohibitive regarding computational resources and without the desirable precision and accuracy.

Several considerations must be inquired to address the problem adequately; for example, consider the first genomic or proteomic sequence that comes to mind. Does a method to accurately distinguish low-complexity regions (LCRs) from the remaining sequence exist? Does this method provide the capability to address different characteristics and proximity within these patterns? If all yes, can this method be implemented as an efficient tool? Can it be implemented to provide visualization of the localized regions? How can this method be appropriately benchmarked? These are the 5 main questions raised and addressed in this article.

We refer to LCRs as the subsequences from a sequence of symbols that are considered to contain lower complexity or a higher number of repetitive characteristics measured above a certain threshold. As an example, consider the following sequence: "ATGCTCGAAAAAAAAAACGAGCAT." From this sequence, is there any repetitive or low-complexity region? One can immediately identify the low-complexity region as the repetition of

Received: June 28, 2023. **Revised:** September 29, 2023. **Accepted:** November 7, 2023
© The Author(s) 2023. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

"AAAAAAAAAA" and the corresponding region coordinates of 8 to 17. However, although not immediately recognizable, there are other LCRs. Specifically, the "ATGCTCG" subsequence from positions 1 to 7 is an inversion of the "CGAGCAT" subsequence at positions 18 to 24. Inversions are subsequences from an original one where instead of a normal order, they are in reverse order (from the end to the beginning or the opposite), followed by a substitution where each element becomes exactly another one (A becomes T, C becomes G, G becomes C, and T becomes A). Despite the simplicity of this transformation, if the method that performs this type of detection is not prepared to find this algorithmic characteristic, the low-complexity region will not be classified as such. Therefore, the better the model, the higher the accuracy and the cumulative amplitude of these regions.

Many other types of low-complexity patterns may be contained in sequences, for example, tandem repeats, exact or similar genes, poly-tales, GC regions, and hairpins, among others [4]. However, low-complexity sequences can cause serious challenges for search and clustering algorithms, for example, based on matching words or patterns. Detecting these regions in biological sequences is also related to differences in the accuracy of sequencing technologies [5] and assembly strategies [3] to reconstruct the original biological sequence. In fact, there are many highly important applications that depend on the correct and non-underestimation of these regions. For example, the recent state-of-the-art viral integration caller uses LCRs as part of fundamental analysis in integration classification [6]. Specifically, inaccuracy in previous integration callers is mainly caused by the incorrect alignment of reads in repetitive regions, leading to false positives. This classification is of extreme importance because predicting virus integration can help uncover the mechanisms that lead to many devastating diseases [7–10].

However, building a model that can accurately identify all these LCRs is not a trivial process because, generically, biological sequences contain LCRs that may be characterized by mutations, heterogeneity, and nonstationary content, and additionally, noise and errors that may be inserted through the sequencing and reconstruction (assembly) process, analogous to predict outcomes based on imperfect data [11]. Moreover, the size of the sequence also matters. If we consider now that there were 20 million random symbols between the identified inverted regions of the above example, would the model still have enough memory and precision to identify it as a low-complexity region? What about the number of sequences that would have exact or similar sequences? There is a high probability of having more 7-nucleotide sequences exactly as "ATGCTCG" or "CGAGCAT" (assuming a uniform distribution) for approximately 20 million symbols. Therefore, is it considered an underestimation if the model considers these subsequences of higher redundancy? Probably, yes. So, where to draw the line of underestimation? Fortunately, models that are upper-bounded approximations to the Kolmogorov complexity implicitly respect this property [12].

Specifically, the Kolmogorov complexity is defined as the size of a shortest program that represents a particular sequence and halts [13]. Therefore, the Kolmogorov complexity is a shortest quantity measure but noncomputable that can only be upper-bounded by an efficient model. Such a model can be a lossless data compressor because its main aim is to minimize the data representation without loss of information. However, the quality and efficiency of the compressor dramatically influence the accuracy and precision of the predictions. Fortunately, data compressors that are prepared to deal with specific properties of the data and consistently provide higher compression ratios are better candidates for predicting LCRs through the generation of complexity profiles.

In practice, a complexity profile is a numeric sequence that estimates the amount of information (e.g., in bits) to compress or represent each symbol of the sequence. The size of the complexity profile is equal to the size of the sequence and, through vertical alignment, provides immediate localization of complexity differences between the sequence regions found by the data compressor in use [14–16]. Regions below a certain threshold are considered of low-complexity or to contain higher redundancy.

The literature provides several examples of the computation of complexity (or entropy) profiles or equivalent approaches for many applications [17, 18]. For example, in the area of computational security [19], these profiles have been used for mimicking antiviruses [20] and detecting malware [21]. Although there has been success in the application of other distinct areas—namely, with the classification of galaxy clusters [22] and finding characteristic fingerprints in lithium-ion cells [23]—the use of these profiles has been widely applied to biological sequence analysis [24–26]. Examples of these biological applications are regulatory region prediction [14, 27–29], identification of inversions [30, 31], predicting coding regions [32], diversity estimation [33], predicting messenger RNA (mRNA) and long noncoding RNA (lncRNA) folding [34, 35], rearrangement detection [36, 37], improving genome reconstruction [38], predicting local LCRs [39], providing biological studies based on LCRs [40, 41], and incorporating into important pipelines [42].

The method proposed in this article (AlcoR), with description and formalization provided in the next section, uses a bidirectional compression scheme of an input string assuming 2 causal directions, from the sequence's beginning to end and the opposite, followed by its minimum, smoothing, segmentation, and visualization operations. The method uses a compression scheme that combines multiple context models with specific memory capacities to consider different distances between patterns. Moreover, a free and efficient standalone implementation of the method is provided in C language with the additional flexibility to address challenges in other areas.

To benchmark the method, we use synthetic sequences as one of the most effective practices in benchmark test methods and data analysis [43, 44]. Here, synthetic sequences are defined as computer-generated sequences that resemble one or multiple features of natural sequences. Synthetic sequences provide a controlled environment with complete or partial expected outcomes, similar to a gold standard. Additionally, they can be used to create or identify protein sequences with optimized properties [45–47]. Ideally, they should be used as the first benchmark test for most existing computational biological tools, especially in low-complexity analysis, given structural complexity differences.

The literature contains several methodologies to generate synthetic sequences using models with low to high simplicity for genomic and proteomic sequences [47–55]. This generation has been augmented to specific file formats that additionally simulate the process of next-generation sequencing (NGS) methodologies (e.g., [56]), providing a critical application to compare existing and new NGS tools or analytical pipelines [57].

Specifically, synthetic sequences can contain different characteristics and generation sources. For example, a synthetic sequence can be generated according to a specific distribution, assuming statistical independence between the outcome of the symbols, or generated according to a model that has been trained in 1 or multiple *natural* DNA sequences. These sequences can also contain deliberated rearrangements (e.g., translocation, in-

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

versions, fusions) or a small variation, substitutions, additions, or deletions of 1 or multiple bases. Another feature that synthetic sequences can contain is the heterogeneity created by multiple sources, for example, through the alternated combination of segments produced by a simple independent model and a model trained with a particular sequence. As an analogy, it can be seen as a book independently written by different authors and combined using a conjunction of pages.

However, measuring sequence complexity approximates Kolmogorov complexity, an uncomputable measure, yielding only an upper bound. LCR mappers detect low-complexity regions (if they do not underestimate), but unmapped areas may also be of low complexity. No computable gold standard exists for benchmarking LCR detection due to unknown reference values. Simulations based on linear functions may serve as an indicative benchmark, but they also create inherently low-complex sequences because there is a simple program to explain them, offering only indicative results as mapper models typically cannot identify their whole origin.

Therefore, to test the method for locating LCRs, we also provide a flexible method that enables the simulation of synthetic sequences. In the next section, we describe this methodology along with the details of its implementation into an efficient tool. Moreover, several illustrative demonstrations are provided using synthetic, nearly synthetic, and natural sequences with different characteristics. Although the AlcoR method is able to provide additional statistical and algorithmic analysis than common alignment mapper tools, we provide several illustrative comparisons for validation and benchmark purposes.

After the benchmark, we apply the LCR mapper to large-scale biological data, providing the complete low-complexity maps for a complete human genome and a complete diploid cassava genome, including a comparison with RepeatModeler and RepeatMasker [58, 59]. Using both synthetic and natural sequences, we provide evidence that the AlcoR tool can efficiently and accurately model probabilistic and several algorithmic patterns, for example, with a considerable degree of substitutions and inversions, unveiling several structural characteristics of sequences without prior knowledge or references.

Methods

This section describes the methods used to map, extract, and visualize LCRs in biological sequences. Additionally, we develop and provide a method in this section to simulate synthetic or nearly-synthetic sequences for the main method benchmark purpose. Finally, we provide the respective implementation details of the whole set of tools. The details of the parameters and how to reproduce the methods are available in Supplementary Sections 2 and 3.

LCR mapper

The proposed method in this article uses lossless data compression for estimating the complexity profiles for further segmentation of LCRs. The biological sequences data compression field began 3 decades ago, starting with Biocompress [60]. Afterward, many algorithms emerged, mostly modeling the existence of exact or approximate repeated and inverted repeated regions, through the usage of simple bit encoding, dictionary approaches, or context modeling; a few examples are [61–75].

Although any efficient lossless data compression method that can provide the number of bits to compress a given symbol is a candidate, a compression scheme that provides state-of-the-art results in genomic sequences in balance with affordable computational resources trade-off is used [76–78]. This compression method uses soft-blending between finite-context models (FCMs) of several context depths and substitution context models with specific subprograms, in the case of genomic sequences, to handle inversions [79]. Additionally, we enhance these models with locality by incorporating several memory mechanisms.

The compression method already provides a setup with several parameterized models for nonexpert usage. Nevertheless, it permits customization of the number of models, context depths, mixture approach, and memory capacity, among others.

Figure 1 illustrates the main steps in this methodology. Accordingly, the complexity profiles are generated using a bidirectional compression scheme of an input sequence assuming 2 causal directions, from the sequence's beginning to the end (left–right) and the opposite (right–left), followed by the minimum of both directions, as well as smoothing and segmentation of the regions below a certain threshold. Finally, the visualization of the regions through the generation of a map automatically occurs. Although we use sequence simulation (generation) to benchmark this methodology, the simulation and the extraction of the sequences are optional operations.

The bidirectional computation is key in this process, ensuring that even an *original* subsequence that is a copy of another is efficiently identified as an LCR. If the previous section's exemplifying sequence was used to generate the complexity profile assuming only the beginning to the end order, the first 7 symbols would not be identified as a low-complexity region. This feature is a characteristic of models that do not track the positions where copies occur, mainly to save computation time and memory. Therefore, the 2 directions are processed in parallel to reduce the computational resources substantially.

Formalization and parameters

Consider a source, X, that has generated n symbols from a finite alphabet Θ with size $|\Theta|$. The nature of the source, X, is unknown, but the sequence of symbols $x_0, x_1, \ldots, x_{n-1}$ is known. The computation $\overrightarrow{C}(x_i)$ delimits the compression of left-to-right symbol access (i.e., from x_0 to x_{n-1} by numeric increasing order), and $\overleftarrow{C}(x_i)$ delimits the compression of right-to-left access (i.e,. from x_{n-1} to x_0 by numeric decreasing order).

The employed data compressor, $C(x)$, is characterized by a combination of multiple FCMs [71, 80]. An FCM has the Markov property, in which the conditional probability distribution of observing a symbol, R, depends only on the state of the preceding k-mer, respecting the following equality:

$$C(x_i) = \frac{R(x_i|x_{i-1}^{i-k}) + \frac{1}{b}}{\frac{|\Theta|}{b}}. \quad (1)$$

Notice that $b \geq 1$ corresponds to a value that is associated with a bet, where the higher the number, the higher the probabilistic confidence (higher bet). Therefore, when $b = 1$ turns into the Laplace estimator, otherwise behaves progressively as a likelihood estimator.

The FCM models can be combined with other models, such as extended FCMs [81] or the substitution tolerant context models [82]. The combination of these models can only access specific memory capacities and, hence, consider different distances between patterns. The access to the sequence is given by

$$\overrightarrow{C}(x_i) = C(x_i|x_{i-1}^{i-p}),\ \overleftarrow{C}(x_i) = C(x_i|x_{i+1}^{i+p+1}). \quad (2)$$

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

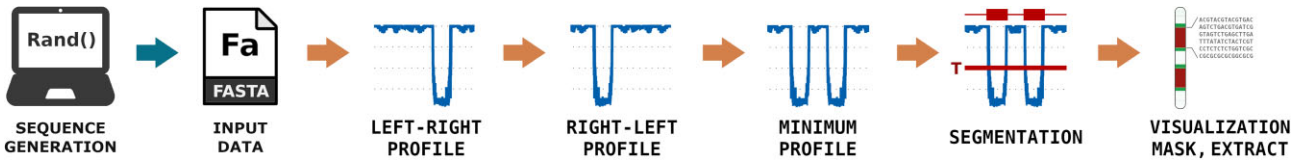

Figure 1: AlcoR pipeline for mapping and visualization of LCRs of a sequence using the bidirectional complexity profiles followed by the minimum, segmentation, and automatic low-complexity map generation (visualization; SVG format). The mask and extract operations of the LCRs and the simulation of synthetic sequences are optional.

Accordingly, the memory used to independently compute $\overrightarrow{C}(x_i)$ and $\overleftarrow{C}(x_i)$ is characterized by a cache history of p symbols. This means that the models do not consider older outcomes from the sequence that are far from p symbols. Accordingly, at symbol position i, the model of $\overrightarrow{C}(x_i)$ can predict the information content having access exclusively to x_{i-1}^{i-p} as $x_{i-1},\ldots, x_{i-p}$, where for $i < p$, the lowest $i = 0$. For predicting the $\overleftarrow{C}(x_i)$, the model has exclusive access to x_{i+1}^{i+p+1} as $x_{i+1},\ldots, x_{i+p}$, where if $i > p$, then $i = n - 1$. Both computations run in parallel because the models are independent, substantially improving the real processing time.

To compute the minimal bidirectional complexity profiles, both left to right as $\overrightarrow{C}(x_i)$ and right to left as $\overleftarrow{C}(x_i)$, computations are applied according to

$$\mathcal{N}(x_i) = \min\{\overrightarrow{C}(x_i), \overleftarrow{C}(x_i)\}. \quad (3)$$

Specifically, both $\overrightarrow{C}(x_i)$ and $\overleftarrow{C}(x_i)$ computations run in a different thread, and then the minimum of both is computed when both threads end the computation. After, the $\mathcal{N}(x_i)$ output is averaged according to the following moving average

$$\mathcal{F}(x_i) = \frac{1}{w}\sum_{j=0}^{w-1} x_{i+j} \quad (4)$$

where w is the window size parameter $\in \mathbb{N}^+$.

The initial and final positions of each region below the threshold, $\mathcal{T}$, are provided as standard output. By default, the threshold is

$$\mathcal{T} = \frac{log_2(\Theta)}{2}, \quad (5)$$

but $\mathcal{T}$ can also be set to a custom value, where $\mathcal{T} \in \mathbb{R}^+$.

Finally, the method outputs the LCRs below the threshold, where regions shorter than a particular size (s) can be automatically discarded from the final set.

Extraction, information, masking, and visualization of LCRs

The AlcoR methodology enables one to extract LCRs of the input FASTA file using the coordinates generated from the mapper. This characteristic allows further analysis using other approaches—namely, comparing or classifying the sequence content within multiple regions.

Another feature that AlcoR provides is to compute the length (nucleotide or amino acid sequence) and the percentage of G and C bases that are registered along with the coordinates of each FASTA read. This is a fast and simple implementation that allows extended analysis for comparative purposes (e.g., to understand the trends of LCRs).

A very useful feature of AlcoR is the ability to soft mask the sequences contained in the FASTA file, where each sequence corresponding to an LCR previously mapped is transformed into lowercase symbols.

Moreover, AlcoR enables the automatic generation of an image file (SVG format) depicting the LCRs in scale with the amplest sequence using an ideogram style. We call this functionality the visualization; it enables reading multiple LCRs from multiple sequence files, additionally providing the capability to combine different levels of low complexity in a single image through the concatenation of the output mapped regions. Additionally, the visualization method enables setting the thickness and distance between the bars and the color of the regions, background, and border. Strict or round corners are also an option. Finally, it provides the functionality to synthetically enlarge the regions to increase visibility when the number of regions is low or the length is small.

Simulation of sequences

The simulation of synthetic sequences provides a relatively controlled environment to benchmark-test the AlcoR mapper or any other tool that uses FASTA data. Accordingly, we developed a FASTA simulation tool to deal with 3 main types of submodels: file extraction, pseudo-random generation, and modeling generation. The file extraction simulation mode considers a subsequence from a FASTA file containing a sequence using the initial and ending positions. The pseudo-random generation simulates custom sequences using a linear congruential generator (LCG) with seed and size as the main parameters. The modeling generation learns from a FASTA file and further generates a sequence with a custom size using an FCM of a given context order and bet parameter. This model will store the counters seen in a particular sequence (or set of sequences) and then compute equation (1), but instead, it will generate a y using $R(y_i|y_{i-1}^{i-k}, x_1^n)$. For each simulation model, a specific FASTA file and FCM are used, providing flexibility to include modeling using different FASTA files, context orders, and parameters.

In any of the subsequences simulated by the previous models, specific transformations can be applied—namely, inversions and mutations. The inversion considers the input sequence is reversed and, in the case of DNA, complemented. The mutations applies single-nucleotide polymorphism (SNP) mutations using specific intensities, according to a probability of specific mutation, through the generation of pseudo-random symbols (LCG). The SNP mutations can be independently applied for substitution, deletion, and addition types.

A very important feature of this method is the capability to generate multiple sequences with different characteristics in a single FASTA file. Therefore, the content of the file can have multiple combinations of the previously reported functionalities through the conjunction (automatic concatenation of the content), enabling it to provide extremely high flexibility.

Additionally, this method enables the application of protein sequences and other symbolic sequences as long as they respect the FASTA file format. Moreover, the alphabet of the sequences can be selected or reduced to ignore certain symbols and provide custom synthetic sequences.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

Implementation

The AlcoR method is implemented in C language and does not contain any external dependencies, including the SVG map generation. The source code and benchmark scripts are freely provided at the repository [83]. Additionally, we provide a website with several pipelines, including a video to better understand AlcoR, available at [84].

The AlcoR tool contains 1 main menu (command: AlcoR) with 5 submenus for computing the features that it provides—namely, info, extract, mapper, simulation, and visual. The info tool retrieves information of the length and GC percentage for each FASTA read. The extract tool extracts a sequence of a FASTA file using positional coordinates (independent from the existing headers of the FASTA files). The mapper tool computes the LCRs of a FASTA read while providing bidirectional complexity profiles and further structural similarity analysis—this option includes soft masking the LCRs of a FASTA file. The simulation tools provides FASTA sequence simulation with features: file extraction, random generation, and sequence modeling (with SNPs specific mutations). Finally, the visual tool computes an SVG file with the respective map containing the LCRs.

For each of the submenu tools, Supplementary Section 3 provides details on each feature, with special focus on the most important parameters.

Results

Benchmarking methods for unsupervised identification of LCRs are challenging because comparing existing methods requires equivalent characteristics and objectives. Specifically, the number of existing methods for biological sequences is low, and only a few provide computational tools that can be effectively installed and run. Some examples of these tools are Cafefilter [85] and DUST [86] for DNA sequences and SEG [87] and CARD [88] for protein sequences. For specific purposes, there are several tools, for example, RepeatModeler for the automated genomic discovery of transposable element families [58] and RepeatMasker for masking LCRs [59].

From this set, some are not prepared to deal with FASTA data, the computational resources of the methods needed to handle these sequences are prohibitive, or the current data size is not appropriate for the computation or visualization. Also, since some of them have been built for specific uses, their comparison becomes highly subjective. Moreover, the AlcoR method contains a low-complexity analysis that comprises local and distant regions at the same time, currently providing singular characteristics.

Additionally, measuring the complexity of sequences is approximating the Kolmogorov complexity, and since it is an uncomputable measure, we can only provide an upper-bound approximation. Therefore, regions detected by LCR mappers are low-complexity regions (if mappers do not underestimate), but regions that are not mapped may be of low complexity since the models may not be able to identify them. Accordingly, there is no computable gold standard for benchmarking LCR detection because the exact reference values are not known. We could argue that simulations may provide a true gold standard for benchmarking, but the reality is that simulations based on linear functions create sequences that are all low complexity because there is a simple program to explain them. Moreover, they may insert small statistical elements that can be identified as low complexity, for example, by merging 2 synthetic sequences. Nevertheless, they can be used as indicative only since mapper models usually do not contain the ability to identify the origin from simulations.

Therefore, to fairly benchmark AlcoR, we progressively provide several demonstration tests using synthetic, nearly synthetic, and natural sequences. In the latter, we provide comparative results with the NCBI sequence viewer (NCBIv) using WindowMasker [89, 90] which is one of the most compelling and used online tools. Finally, we provide the complete maps of the recent complete human genome and the diploid Cassava chromosomes, including a comparison with RepeatModeler and RepeatMasker for the Cassava. These results can be reproducible using the repository provided at [83].

All the computations have been carried out using a laptop computer with 8 GB RAM, eight 11th Gen Intel Core i5 @ 2.40 GHz, an SSD disk with a capacity of 512 GB, and an operating system running the Linux Ubuntu 20.04 LTS.

Demonstrations

This subsection provides a set of illustrative and interactive demonstrations of the method using several synthetic, nearly synthetic, and natural sequences characterized by different types of copies and transformations—namely, inversions and mutations. These demonstrations provide the ability to change conditions to provide personalized analysis any time during these results, for example, the edition of the smoothing window size, threshold, compression models, or the simulation composition. The conjoint computation of the whole demonstrations described below took merely a few seconds using the computational resources described above. Additional information on the instructions to run the results is available in Supplementary Section 2, while detailed information on the AlcoR parameters and meaning is available in Supplementary Section 3.

Demonstration A

The first demonstration (demo) contains 2 scripts (GenSeqA.sh and DemoA.sh) that can be accessed using the folder "demos/demoA/." This demo uses AlcoR to simulate a FASTA file (sampleA.fasta) through the recursive generation of several pseudo-random (synthetic) sequences. For generating this file, the command "./GenSeqA.sh" must run. This action will compute "AlcoR simulation" and generate a synthetic sequence using a uniform distribution through a subprogram that uses LCG [91]. This sequence (sample.fasta) contains 11 segments of synthetic DNA, some of which are exact, inverted, or mutated copies. An illustration of this sequence's low-complexity coordinates (lilac color) is added to Fig. 2A as the ground truth (Truth). The details of the composition of this synthetic sequence are described at the Supplementary Subsection 2.2.

This demonstration aims to benchmark the method—namely, in the capability to find these regions and through similarity despite the transformations of inversion or mutation. For computing this demonstration, the "./DemoA.sh" script must run. This action will compute the "AlcoR mapper" for 3 thresholds (0.5, 1.0, and 1.5). The smoothing window is set to 10, and a context model of order-13, prepared to deal with inversions, is used in cooperation with a substitution-tolerant context model (allowing 5 substitutions in a context of 13) of the same order. Accordingly, the inner parameterization for each threshold is set to a model with 13:50:0:1:10:0.9/5:10:0.9, a window of 5, and a threshold of 1. Additional information on the parameters and meaning is available in Supplementary Section 3.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

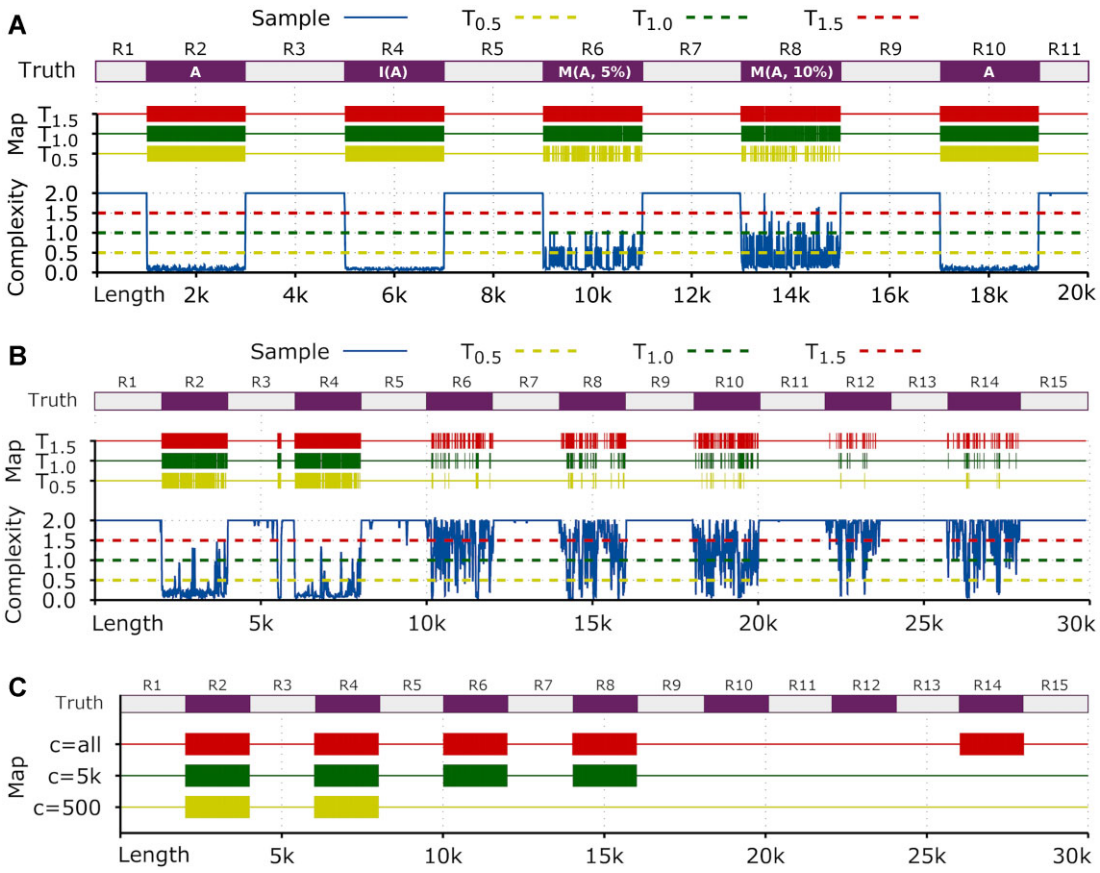

Figure 2: Low-complexity map and profiles for demonstrations A, B, and C. (A) Minimal bidirectional complexity profile of a synthetic sequence (sampleA.fasta) and the corresponding map of the segmented LCRs below the respective threshold. In the Truth panel, the A region is used as a fixed random sequence for the following transformations: inversion (I) and mutation (M) with the respective percentage. The remaining regions are probabilistic random and unique. (B) Minimal bidirectional complexity profile of the sampleB.fasta sequence and the corresponding map of the segmented LCRs below the respective threshold. In the Truth panel, the regions are according to the simulations and transformations described above. (C) Low-complexity map of the sampleC.fasta sequence with the segmented regions below the 1.0 threshold for different *c* values (yellow, 500; green, 5,000; red, whole memory). In the Truth panel, the regions are according to the simulations and transformations described above. All the Truth panels have been added to the image.

This tool will compute equation (4) and the respective segmented regions. Then, Gnuplot is used to generate the plots automatically from the script. The output plot is a PDF file with the name outA.pdf containing the minimal bidirectional complexity profile and the corresponding horizontal map of the segmented LCRs for each threshold.

As depicted in Fig. 2A, all the LCRs are accurately mapped for the thresholds $T_{1.5}$ and $T_{1.0}$, including the inverted region and both copied regions with substitutions. Identifying these regions is more challenging for the lower threshold, despite their good performance, assuming the high level of mutations that R6 and R8 contain.

Demonstration B

Although the performance of Demonstration A seems to be quite convincing, the application in natural sequences has additional challenges. One of the challenges is the presence of a nonuniform nucleotide content. Therefore, instead of a maximum average complexity for encoding each base of $\log_2(\Theta)$, this value will be approximately at least equal to or less than the Shannon entropy [92]. Therefore, the threshold needs to be approximated to half of this value. Usually, visual inspection and optimization after the first run provide efficient results, but this process can also be automatic.

Another challenge is the characteristics of nonstationary content of higher-complexity regions, usually with the presence of peaks that have a mixed vertical direction (the same as in region R8 but of average higher complexity). This challenge requires an adaptive decrease of the smoothing parameter. However, this parameter solicits a trade-off. The lower the smoothing parameter, the lower the precision of the regions below a given threshold, while the higher the parameter, the higher the fragmentation and possibly the lower the accuracy in detecting the LCRs. Nevertheless, good precision and accuracy are achievable with a proper balance.

Accordingly, in this demonstration, a mixture of regions from synthetic and generated sequences is used, recurring in context models trained with a nonsynthetic viral sequence. For accessing the folder with the contents to perform this demonstration, the directory must be changed to "demoB." The "GenSeqB.sh" is a script that will create a FASTA file (sampleB.fasta) with the conjunction of several synthetic and nonsynthetic sequences. For generating this file, the "./GenSeqB.sh" command must run. This action will compute "AlcoR simulation" and generate the sequences containing 15 segments of synthetic DNA, some of which are exact, inverted, or mutated copies.

An illustration of this sequence is provided in Fig. 2B as the ground truth (Truth). The details of the composition of this

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

synthetic sequence are described in Supplementary Subsection 2.3.

For computing this demonstration, the "./DemoB.sh" script must run. This action will compute "AlcoR mapper," similar to Demonstration A but instead using the window size of 20 (the whole sample is larger). The output is the PDF file outB.pdf with the content provided in Fig. 2B.

The demonstration results include some examples with very high mutation rates, for example, a deletion mutation of a particular symbol with a probability of 0.15 and an additional mutation of a specific symbol with a probability of 0.1. Moreover, the subsequences contain substitution mutations, inversions, and origins from different sources. Despite the complexity of the challenge, all the LCRs have been identified. A small low-complexity region has been additionally identified in the R3 mainly because this sequence belongs to the HHV6B, which contains a low-complexity sequence. Despite the correct identification, the tuning of the filter window size can improve the results, for example, by enlarging the mapped LCRs.

Demonstration C

Demonstrations A and B showed the capability of the methodology to find LCRs with lower to higher mutation rates. However, a question remains: how are local LCRs split from distant LCRs? This demonstration answers this question by computing the method using different cache histories (p). Several subsequences with local and distant LCRs are simulated and modeled using natural sequences for this benchmark.

For accessing the folder with the contents to perform this demonstration, the directory must be changed to "demoC." The "GenSeqC.sh" will create a FASTA file (sampleC.fasta) with the sequences while assuming different low-complexity levels and distances. For generating this file, the "./GenSeqC.sh" command must run. This action will compute "AlcoR simulation" and generate a sequence with the mentioned characteristics.

An illustration of this sequence is provided in Fig. 2C as the ground truth (Truth). The details of the composition of this synthetic sequence are described in Supplementary Subsection 2.4.

For computing this demonstration, the "./DemoC.sh" script must run. This action will compute "AlcoR mapper," similar to Demonstrations A and B, but exclusively map the LCRs for different c values. The output is the PDF file outC.pdf with the content provided in Fig. 2C.

This demonstration shows the potential of this application for mapping and visualizing LCRs with low (500 bases), medium (5,000 bases), and high distances (all bases) between patterns. Region R4 is a copy of R2, and a high-redundancy content characterizes R2, thus enabling that the low memory is enough to map these regions. Region R8 is a copy of R6 with high statistical entropy and naturally is only mapped by medium and large c. Finally, R14 is also a copy of R8 and R6, enabling the detection for $c > 10k$, where $c =$ all providing this capability. As in Demonstrations A and B, this demonstration shows the automatic and accurate identification of the whole regions containing low complexity but, in this case, using different distances between low-complexity patterns. For example, while local patterns are associated with regulatory regions, distant patterns can account for similar genes or segmental duplications [93, 94].

As mentioned before, AlcoR can be applied to other types of sequences as long as it respects the FASTA format. To prove this, we provide the same demonstration for amino acid sequences, including the simulation, mapping, and visualization of the respective LCRs. This process can be replicated using the instruction in Supplementary Section 2.4.1. The results show the same output map despite the difference in the sequences used and the automatic calculation of the threshold using the process described in equation (5).

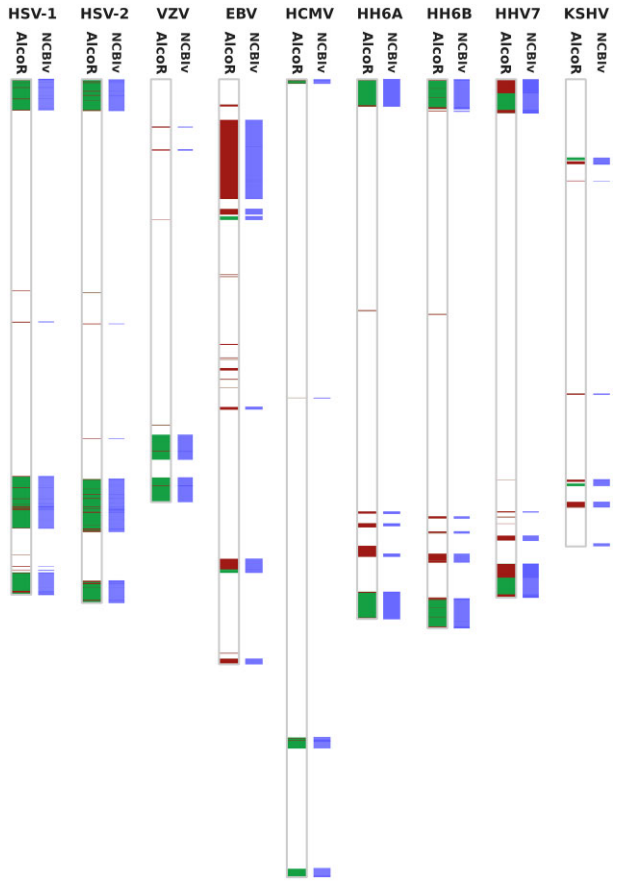

Figure 3: Low-complexity map of each herpesvirus sequence included in this sample with the segmented regions below the 1.2 threshold for different c values (red, 5,000 symbols; green, whole memory). Red regions overlap green regions. In each viral sequence, the panel at the right describes the map produced by localizing repeating regions using the NCBI sequence viewer (NCBIv). The NCBIv panel has been added to the image. The maps are proportionally in scale according to the length of the sequences.

Biological data

For testing AlcoR exclusively in biological data (no synthetic data), we use the reference genomes from 9 human herpesviruses (HHVs): herpes simplex virus 1 (HSV-1, NC_001806.2), herpes simplex virus 2 (HSV-2, NC_001798.2), varicella-zoster virus (VZV, NC_001348.1), Epstein–Barr virus (EBV, NC_009334.1), human cytomegalovirus (HCMV, NC_006273.2), human herpesvirus 6A (HHV6A, NC_001664.4), human herpesvirus 6B (HHV6B, NC_000898.1), human herpesvirus 7 (HHV7, NC_001716.2), and Kaposi sarcoma–associated herpesvirus (KSHV, NC_009333.1). Using these genome sequences, we compute the low-complexity maps and manually align them with the structural images retrieved from the NCBI sequence viewer tool (NCBIv) at the NCBI portal for comparison purposes [89, 95, 96]. The AlcoR computation map can be reproduced by running the script "RunHerpesvirus.sh" in the folder "herpesvirus."

Figure 3 provides the low-complexity map of each herpesvirus sequence included in this sample with the segmented regions be-

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

low the 1.2 thresholds for local (red) and distant (green) regions and ignoring regions below 50 bases. Running the low-complexity maps with AlcoR, including the generation of the figure map for the whole herpesviruses, took approximately 10 seconds using the computer characteristics described above. All the regions that the NCBIv mapped were reported by AlcoR (except for the final region in KSHV). For this final region, we changed the threshold for a lower value but did not have these regions considered as low complexity. Also, we visually inspected this region, and besides a tendency for a higher content of GC, we did not find any relevant low-complexity pattern. Moreover, we extracted the last 1,000 symbols that contained this region highlighted by the NCBIv and measured the normalized compression (NC) [97] with GeCo3 [76] using custom optimized parameters (and ignoring the information header with the models), reaching a NC minimum of approximately 0.87 (repetitive values are close to zero, and self-dissimilar values are close to 1). This value indicates that there are no relevant low-complexity patterns in this subsequence. Potentially, this region may be underestimated, as an information quantity, by the NCBIv tool through the distribution of the nucleotide content.

The remaining reported regions are consistently localized in identical coordinates, as shown in Fig. 3 with the horizontal alignment between the maps. Additionally, AlcoR identified several more regions than the NCBIv tool. The main justification for these identified regions is given by the mixture of models and algorithmic methodologies that AlcoR contains, providing substantially higher sensitivity without compromising the efficiency of the computation regarding computational resources (memory and time) and, more important, without underestimation.

However, notice that the direct comparison of these tools is unfair. The parameters used in the NCBIv were the default (*k*-mer of 28 and threshold of 0.05), and by optimization, several of the regions that AlcoR mapped potentially will also be mapped by the NCBIv. However, the main reason for this unfair comparison is that these algorithms have different purposes, although they can be approximately compared for specific cases. The NCBIv tool can compute repetitive patterns through alignments using a single *k*-mer model; however, these are only a subgroup of LCRs. Additionally, the AlcoR tool can retrieve additional structural, distant, and algorithmic characteristics using multiple models with different capabilities. Nevertheless, we used the NCBIv tool specifically as a control.

Supplementary Table S1 provides information regarding the cumulative number of bases and respective sequence proportion considered by AlcoR as local and overall (local and distant) LCRs. These results are the automatic computation of AlcoR that can be reproducible using Supplementary Section 2.5.

Large-scale data

The computation and visualization of LCRs in large-scale data (>100 MB) are more challenging because of the diversity of low-complexity patterns, the size of the sequences, and the required higher computational resources (computational time and RAM) to cope with the memory of distant low-complexity patterns. For this purpose, the window size must be adapted to a higher value (usually to 5,000), the compression models need to be set with a high-order context model (at least 13), and if the number of segmented regions is too high for visualization purposes, ignoring some small regions (usually below 5,000) provides effective descriptive maps.

Taking into account these standard parameters, we run AlcoR unprecedentedly in 2 whole genomes—namely, the recent complete human genome [98] and the haplotype-resolved chromosome pairs of a heterozygous diploid African cassava cultivar (TME204) [3]. The low-complexity maps have been computed for both genomes using local low complexity (using a memory of at most 5,000 symbols) and the distant (or global) low complexity. Since the local low complexity (red color) is depicted on top of the complete low complexity in the visualization, it enables us to visualize the whole LCRs as exclusively distant (green color).

The computation of the whole maps took approximately 203 and 88 minutes for the human genome and the cassava diploid genomes, respectively, using the computational characteristics described above. These runs can be additionally parallelized, providing substantial computational time savings. The maximum peak RAM was approximately 1 GB for both genomes. To replicate the human run, the folder "human" contains the scripts "GetHuman.sh" and "RunHuman.sh" that must run in this order. To replicate the cassava run, the folder "cassava" contains the scripts "GunzipCassava.sh" and "RunCassava.sh" that also must run in this order.

The low-complexity map for the whole human chromosomes is depicted in Fig. 4. As depicted, most of the local LCRs overlap previous unsequenced regions characterized by high repetitive nature, for example, segmental duplications, GC regions, and telomeric regions, among others [93, 99]. Many of these regions are localized in the same coordinates addressed by segmental duplication analysis [99].

However, other regions that have been highlighted by AlcoR are characterized by other sources, namely, centromeric regions [100, 101]. Recently, a substantial structural diversity has been found in centromeric regions as human haplotype-specific evolution, while frequent SNPs remained conserved [102]. This finding shows the importance of having fast and sensitive tools to map these regions automatically for downstream analyses, especially in the new ongoing era of telomere-to-telomere (T2T) assembly [103].

Supplementary Table S2 provides the overall low-complexity quantities for each chromosome according to its length for both local and distant patterns. The chromosomes that show the highest ratio of LCRs are 1, 7, 9, 13, 14, 15, 16, 17, 19, 20, 21, 22, X, and Y. From these, the chromosomes with the highest levels of LCRs of local distant are 9, 13, 15, 16, 19, 21, 22, and Y. On the other hand, chromosomes 2 and 6 contain the lowest ratio of LCRs.

Additionally, this method offers the characterization possibility of each region through the automatic extraction of each sequence by the respective coordinates. This means that each region can be extracted and analyzed or compared against well-known sequences, such as GC regions, telomeric, centromeric, and TATA boxes, among others, and after, the index color can be changed according to a desired value. This flexibility allows the creation of low-complexity maps with specific colors standing for different sequence characteristics.

The low-complexity map for the diploid cassava chromosomes is depicted in Fig. 5. It is interesting to notice that in several chromosomes, for example, in 1, 2, 4, 14, 15, 16, and 18, the number of low-complexity mapped regions is higher in one part of the chromosome than the other. This finding is also congruent in both the haplotypes. The questions that emerge are if these LCRs are smaller (in length) in a chromosome harm or some chromosomal harms contain more repetitive parts than the others and if this distribution is related to the organization of the genomes, rearrangement probability, and speciation.

In each pair of chromosomes, there are an extensive number of LCRs that, by a horizontal visual alignment between the haplotypes, depict an extensive number of visible vertical changes. Specifically, if the sequence pairs were very similar, the low-

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

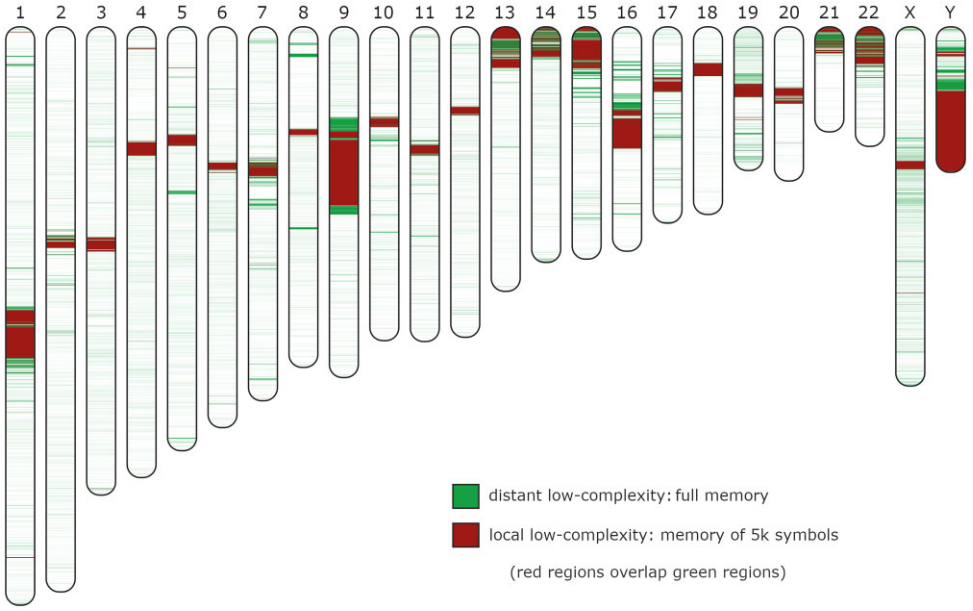

Figure 4: Human chromosome low-complexity (redundancy) map using different memories. The green color stands for the identification of LCRs using a setup of a model that uses the full memory (cache history of all symbols), while the red color uses a memory of $p = 5{,}000$ symbols. In the map, the regions with the red color overlap the green; therefore, regions marked with red are also green, and the green regions that are visible stand for low complexity related to distant patterns. Contiguous segmented regions below 5,000 symbols have been discarded from the visualization. All the chromosomes are in scale, with chromosome 1 (the largest) characterized by approximately 240 MB of length.

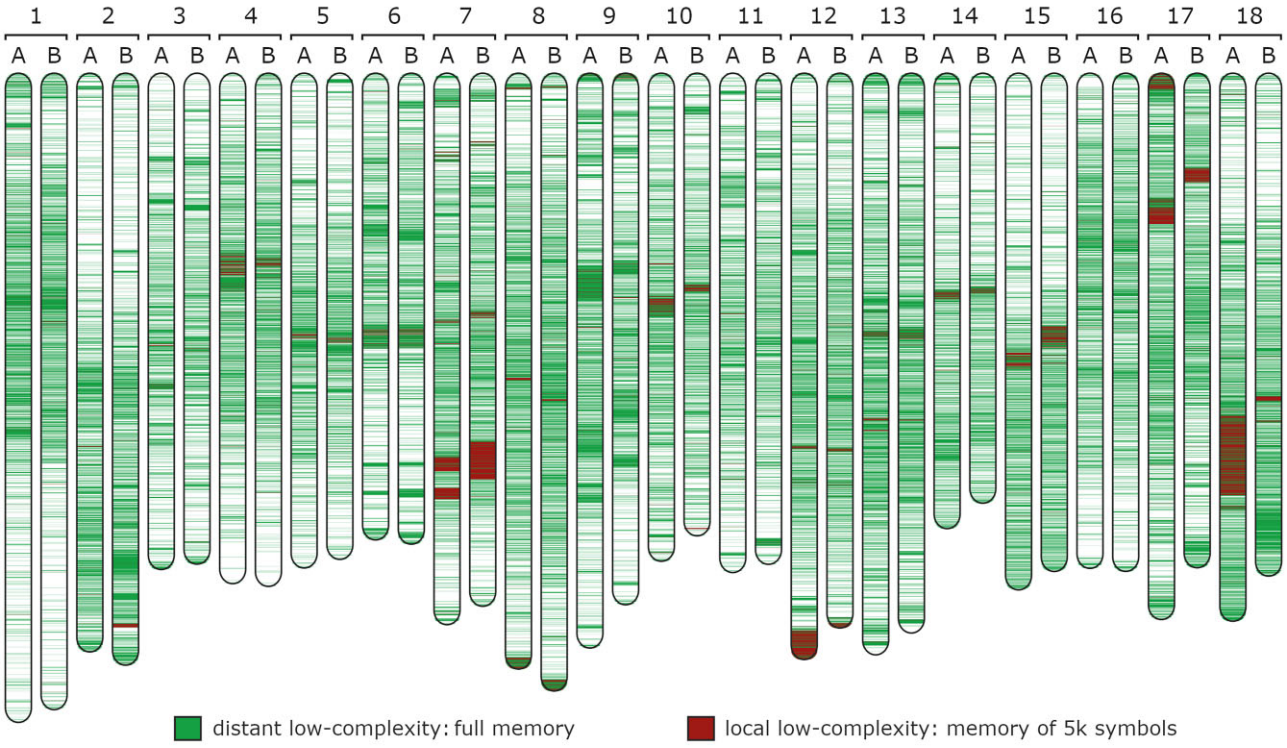

Figure 5: Diploid cassava (TME204) chromosome low-complexity (redundancy) map using different memories. Each chromosome number (1 to 18) contains the respective haplotype (A or B). We used 0.6 as the threshold because of the higher global low complexity of this plant genome. The green color stands for the identification of LCRs using a setup of a model that uses the full memory (cache history of all symbols), while the red color uses a memory of $p = 5{,}000$ symbols. In the map, the regions with the red color overlap the green; therefore, regions marked with red are also green, and the green regions that are visible stand for low complexity related to distant patterns. Contiguous segmented regions below 5,000 symbols have been discarded from the visualization. All the chromosomes are in scale with chromosome 1–haplotype A (the largest), which is approximately 44 MB of length. Although the labels have been manually added, the full map has been generated automatically using AlcoR.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

complexity map would be highly similar; this case shows many differences in all chromosome pairs. These changes are increasingly visible in the regions identified as local low complexity (red regions). These findings are in accordance with the recent analysis of the extensive number of rearrangements found in this genome, namely through the comparison with previous references and between haplotypes [3].

Moreover, cassava, like some other plant species, has "atypical" centromeres [104]. As in [3], we observed that in some chromosomes, the number of LCRs was higher in 1 chromosome arm than the other, such as in extreme cases of chromosomes 2 and 18, where the pericentromere and centromere region extends along 1 chromosome arm.

Supplementary Table S3 provides the overall low-complexity quantities for each pair of chromosomes according to their length for both local and distant patterns. We are able to notice extensive redundancy differences between the haplotypes. Specifically, the most obvious difference is in the local low-complexity percentage between haplotypes A and B in chromosome 18. Supplementary Fig. S1 enhances this difference that can only be marginally approximated in chromosomes 12 and 17.

We also notice 2 large local LCRs in the tips of haplotype A in chromosomes 12 (end tip) and 17 (beginning tip). For example, the end tip in chromosome 12 and haplotype A has been shown to be dissimilar relative to reference AM560 v8.0 assembly and the haplotype B of TME204. The AlcoR method shows that this large segment is characterized by an approximate continuous local repetitive nature (below 5,000 symbols). Using AlcoR extract, we extracted this segment and noticed that it contains a large number of approximate copies of the sequence "TTTTGGGATGGAAGCTGTTAGTCCGAAATCGGGCACCGGATGTAACAATGATGATAGACTTGGCAG." A BLASTn search (on 4 January 2023) of this sequence did not report any significant similarity, including the optimization with the increase of sensitivity, mismatch to 4/5, and the unfiltering of LCRs. This repetitive region is approximately 3% of this haplotype chromosome. An additional analysis (the comparative analysis that can be reproducible using Supplementary Subsection 2.6.2) on similar sequences in the remaining cassava sequences showed similarity with some regions of chromosome 17 (haplotype A) in nearly 1% of the chromosome (interestingly, not found in haplotype B), followed by chromosome 12 (haplotype B) with nearly 0.2%, and also in chromosome 8 (haplotypes A and B) with nearly 0.15% and 0.13%, respectively.

For benchmarking purposes, we now compare AlcoR with RepeatModeler and RepeatMasker [58, 59] in the task of sequence masking. Notice that this comparison is only indicative because the methods contain different specifications and, thus, the outcome can be substantially different.

From the AlcoR LCRs mapped regions, 68% and 70% completely (100%) overlapped with those masked by RepeatModeler+RepeatMasker in A (haplotype 1) and B (haplotype 2), respectively. In both haplotypes, 6% of AlcoR masked regions were not masked by RepeatModeler+RepeatMasker. The rest were partially overlapped.

The 6% AlcoR specific regions were likely caused by the parameter setting of RepeatMasker. These sequences were mostly very short, with low GC (Supplementary Fig. S2). The longer sequences were simple tandem repeats. Since RepeatMasker was set with "-nolow -norna," it is expected that simple tandem repeats and low-complexity (polypurine, AT-rich) regions were not masked.

Among the regions masked by RepeatModeler+RepeatMasker, 3% in both haplotypes were not masked by AlcoR. These regions were masked by similarity to 1,260/1,256 repeat families predicted by RepeatModeler, of which only 2 repeat families (rnd-1_family-558 and rnd-1_family-598) were totally missed by AlcoR. The reason is that AlcoR only mapped at a chromosome level for visualization purposes. Although this type of analysis can be performed by smash++ [36], AlcoR can also perform the whole-genome analysis without affecting the processing time if the chromosomes are in a single FASTA file (e.g., concatenated).

All in all, AlcoR is valuable in quickly masking LCRs in newly assembled genome sequences and visualizing the LCRs. Specifically, AlcoR is highly sensitive and can be used in comparative genomics for fast sequence complexity visualization. Moreover, AlcoR does not depend on software dependencies, unlike RepeatModeler. AlcoR is also time and resource-light, unlike the other tools. Moreover, it also contains additional tools for simulation, extraction, and computation of statistical information.

Discussion

Alignment-based methods usually offer an intuitive and enhanced local resolution that prevails in the comparative analysis of specific features. However, with the substantial increase in the development and availability of alignment-free methods [105–109], these alignment-free methods have evolved from unrolling an alternative way to provide clear advantages that would otherwise be complex to achieve using feasible computational resources, mainly in large-scale data. There are many examples, for instance, the SpaM [110] and Prot-SpaM [111], that provide the calculation of a higher data volume with similar or improved accuracy. Another example is the fast metagenomic composition characterization that can be achieved by these methodologies with high accuracy for extant [112] and ancient [113] samples.

Moreover, alignment-free methods based on lossless data compression offer a clear advantage: the absence of underestimation through the approximation of the Kolmogorov complexity. This property is essential to avoid false positives when the sensitivity is higher, especially when dealing with the identification of LCRs.

Therefore, the automatic identification of these LCRs without underestimation provides a sensitive and fast complementary approach to minimize the set of regions that have a higher probability of containing local errors and structural misassemblies in initial draft assemblies [114].

The recent development of T2T sequencing and assembly methodologies provides the reconstruction of the full sequences, including the LCRs that previously were challenging to assemble because of the limited capability and assembly ambiguity that short reads provide [115]. This technology is extremely valuable to provide completeness of large genomes as well as the ability to solve haplotype characterization. Ironically, this technology is solving the high-complexity task beyond the sequencing and assembly of LCRs, which are the main gaps that large sequence genomes, such as the human genome, previously contained.

In this article, we have proposed AlcoR, offering a convenient, fast, and easy way to automatically produce multiple low-complexity maps (and coordinates retrieval) of small to large sequence species that are being reconstructed with this technology without recurring to any additional tool (than AlcoR). The AlcoR methodology is made available at an ideal time since the quality of these complete sequences is now satisfactory across the entire sequence, including in repetitive or redundant subsequences. From the cassava analysis in this article, we can verify a high variability of LCRs between chromosomal haplotypes that before were thought to be in a smaller quantity.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

Another area where LCRs are currently being seen as fundamental is viral integration. This area addresses the potential capability of viruses to integrate into the host genome, where some are particularly associated with tumorigenesis [116]. Therefore, viral integration is a probabilistic genetic marker for discovering virus-caused cancers and inferring cancer cell development [117].

There are several computational tools to predict viral integration [6, 118–120]. However, only recently, a methodology concerning LCRs (namely repeats) has been developed [6]. Specifically, inaccuracy is caused by the inconsistent alignment of reads in repetitive regions. Since integration sites vary according to the virus, host (usually in telomeric regions), and unknown factors, the availability of AlcoR to map and localize LCRs in both virus and host is significant; this allows the selection, validation, and knowledge building on this highly important thematic, especially with the surprisingly high number of viral genomes residing in human tissues [121].

The AlcoR package also contains a new FASTA simulation tool. The AlcoR mapper could not be adequately tested without the simulation tool because of its mapping singularity capability. The AlcoR flexible simulation tool allowed us to successfully validate the mapper by generating the data according to expected outcome. Notice that validating AlcoR against other tools is unfair because the mapper is designed to dismiss false positives and enables finding LCRs containing different characteristics and specific potential algorithmic expressions. Nevertheless, both the AlcoR simulation and mapping tools have been validated in a complementary way. The AlcoR simulation tool would not be properly validated without a tool to detect LCRs. At the same time, the AlcoR mapper tool would also not be properly validated without the simulation tool. Therefore, this duality represented in this article was essential.

The final remark of low-complexity mapping is related to optimizing reference-free data compression tools. Identifying regions with similarity within a sequence enables clustering regions with the same characteristics, especially for distant LCRs. This means that AlcoR can be used to map distant LCRs followed by grouping and using a data compressor with lower RAM to increase the lossless compression of genomic or proteomic sequences. Naturally, this requires using side information to describe the regions and ensure a complete and lossless decompression. However, notice that the decompression side does not require the re-detection of these regions; hence, it provides a faster decompression process. Therefore, the development of tools such as AlcoR contributes to developing higher-ratio data compression tools while maintaining the same or smaller computational resources in the lossless decompression process.

Conclusion

This article proposed an alignment-free method for the simulation, mapping, and visualization of LCRs in FASTA data. The implementation of the method has been provided in an open-source tool (AlcoR) without any dependencies on external software. The AlcoR method uses parallel bidirectional complexity profiles. It is fully automatic, where the input consists of a FASTA file, and the output provided contains coordinates and an SVG image map and file(s) with the respective coordinates. AlcoR can successfully mask sequences with high sensitivity and without underestimation.

Included in this tool is the additional capability to simulate FASTA sequences with different characteristics, namely using reference-based and reference-free approaches as well as the application of custom degrees of mutations. The AlcoR mapper can distinguish between local and distant LCRs without underestimation and is optimized and efficient to be applied in small- to large-scale sequences.

The tool has been successfully benchmarked using synthetic, nearly synthetic, and natural sequences, providing whole LCR identification using efficient computational resources. We applied AlcoR in large-scale data, unprecedentedly providing a whole-genome low-complexity map of a recent complete human genome and haplotype-resolved chromosome pairs of a heterozygous diploid African cassava cultivar. Additionally, we extended the analysis to identify some insights into repetitive cassava-specific sequences.

With the current T2T sequencing and assembly methodologies, AlcoR offers a convenient, fast, and easy way to automatically produce multiple low-complexity maps of species that are being sequenced and assembled with this technology without underestimation and recurring in any additional tool.

Availability of Source Code and Requirements

- Project name: AlcoR
- Project homepage: https://github.com/cobilab/alcor
- Video and tutorials: https://cobilab.github.io/alcor
- RRID: SCR_023728
- Biotools: alcor
- Operating system(s): Linux
- Programming language: C (main program) and Bash (scripts)
- Other optional requirements: Conda v4.3.27
- License: GPLv3 License

Abbreviations

B19V: human parvovirus; EBV: Epstein–Barr virus; FCM: finite-context model; FLOAT: a positive real number with 16 bits of precision; GC: guanine and cytosine; HCMV: human cytomegalovirus; HHV: human herpesvirus; HHV6A: human herpesvirus 6A; HHV6B: human herpesvirus 6B; HHV7: human herpesvirus 7; HSV-1: herpes simplex virus 1; HSV-2: herpes simplex virus 2; KSHV: Kaposi sarcoma–associated herpesvirus; LCG: linear congruential generator; LCR: low-complexity region; MB: megabases; NC: normalized compression; NCBI: National Center for Biotechnology Information; NCBIv: NCBI Sequence Viewer; SVG: Scalable Vector Graphics; T2T: telomere-to-telomere; VZV: varicella-zoster virus.

Additional Files

Supplementary Table S1. Viral genome low-complexity quantities. Local stands for a $c = 5{,}000$.

Supplementary Table S2. Human chromosomal low-complexity quantities. Local stands for a $c = 5{,}000$.

Supplementary Table S3. Cassava chromosomal low-complexity percentages for both haplotypes (A and B). "L+D" stands for local and distant low complexity while "L" stands for a local low complexity with $c = 5{,}000$.

Supplementary Fig. S1. Cassava low-complexity proportion in percentages for both haplotypes (A and B). "L+D" stands for local and distant low complexity while "L" stands for a local low complexity with $c = 5{,}000$.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

Supplementary Fig. S2. Cassava low-complexity regions according to GC and length for A and B.

Authors' Contributions

A.P. and D.P. designed the experiment. J.S. and D.P. coded the tools. W.Q. and D.P. executed the data analysis. W.Q. and D.P. discussed the results. J.S. created the video tutorial. D.P. created the website. D.P. wrote the manuscript. All the authors revised the manuscript.

Funding

National funds partially funded this work through the Foundation for Science and Technology (FCT), in the context of the project UIDB/00127/2020. Furthermore, this work has received funding from the EC under grant agreement 101081813, Genomic Data Infrastructure. J.M.S. acknowledges the FCT grant SFRH/BD/141851/2018. National funds funded D.P. through FCT–Fundação para a Ciência e a Tecnologia, I.P., under the Scientific Employment Stimulus–Institutional Call (reference CEECINST/00026/2018).

Data Availability

Supplementary material, available in the *GigaScience* repository, GigaDB [122], includes the information to download, install, and use the toolkit as well as the case studies used in the experiments and results of this article.

The datasets supporting the results of this article are publicly available in the NCBI repository. The identifiers for the herpesviruses are NC_009334.1 (EBV), NC_006273.2 (HCMV), NC_001664.4 (HHV6A), NC_000898.1 (HHV6B), NC_001716.2 (HHV7), NC_001806.2 (HSV-1), NC_001798.2 (HSV-2), NC_009333.1 (KSHV), and NC_001348.1 (VZV). The identifiers for the human genome T2T are contained from NC_060925.1 to NC_060948.1, while those for both cassava haplotype genomes are contained from CM037096.1 to CM037131.1.

Competing Interests

The authors declare no competing interests.

Acknowledgments

The authors thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

1. Miga KH, Koren S, Rhie A, et al. Telomere-to-telomere assembly of a complete human X chromosome. Nature 2020;585(7823):79–84. https://doi.org/10.1038/s41586-020-2547-7.
2. Wang T, Antonacci-Fulton L, Howe K, et al. The Human Pangenome Project: a global resource to map genomic diversity. Nature 2022;604(7906):437–46. https://doi.org/10.1038/s41586-022-04601-8.
3. Qi W, Lim YW, Patrignani A, et al. The haplotype-resolved chromosome pairs of a heterozygous diploid African cassava cultivar reveal novel pan-genome and allele-specific transcriptome features. GigaScience 2022;11:giac028. https://doi.org/10.1093/gigascience/giac028.
4. Hubbard T, Barker D, Birney E, et al. The Ensembl genome database project. Nucleic Acids Res 2002;30(1):38–41. https://doi.org/10.1093/nar/30.1.38.
5. Reis AL, Deveson IW, Madala BS, et al. Using synthetic chromosome controls to evaluate the sequencing of difficult regions within the human genome. Genome Biol 2022;23(1):1–24. https://doi.org/10.1186/s13059-021-02579-6.
6. Rajaby R, Zhou Y, Meng Y, et al. SurVirus: a repeat-aware virus integration caller. Nucleic Acids Res 2021;49(6):e33. https://doi.org/10.1093/nar/gkaa1237.
7. Kao JH, Chen DS. Global control of hepatitis B virus infection. Lancet Infect Dis 2002;2(7):395–403. https://doi.org/10.1016/S1473-3099(02)00315-8.
8. Schiffman M, Castle PE, Jeronimo J, et al. Human papillomavirus and cervical cancer. Lancet 2007;370(9590):890–907. https://doi.org/10.1016/S0140-6736(07)61416-0.
9. Parkin DM. The global health burden of infection-associated cancers in the year 2002. Int J Cancer 2006;118(12):3030–44. https://doi.org/10.1002/ijc.21731.
10. Xu M, Zhang WL, Zhu Q, et al. Genome-wide profiling of Epstein-Barr virus integration by targeted sequencing in Epstein-Barr virus associated malignancies. Theranostics 2019;9(4):1115. https://doi.org/10.7150/thno.29622.
11. Golan A. Foundations of Info-metrics: Modeling, Inference, and Imperfect Information. Oxford University Press: Oxford; 2018.
12. Li M, Vitányi P. An Introduction to Kolmogorov Complexity and Its Applications. Vol. 3. Springer: New York; 2008.
13. Kolmogorov AN. Three approaches to the quantitative definition of information. Prob Inf Trans 1965;1(1):1–7.
14. Pinho AJ, Garcia SP, Pratas D, et al. DNA sequences at a glance. PLoS One 2013;8(11):e79922. https://doi.org/10.1371/journal.pone.0079922.
15. Ferragina P, Giancarlo R, Greco V, et al. Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment. BMC Bioinformatics 2007;8(1):1–20. https://doi.org/10.1186/1471-2105-8-252.
16. Orlov YL, Potapov VN. Complexity: an internet resource for analysis of DNA sequence complexity. Nucleic Acids Res 2004;32(suppl 2):W628–33. https://doi.org/10.1093/nar/gkh466.
17. Vinga S, Almeida JS. Local Renyi entropic profiles of DNA sequences. BMC Bioinformatics 2007;8(1):1–19. https://doi.org/10.1186/1471-2105-8-393.
18. Kempa D, Prezza N. At the roots of dictionary compression: string attractors. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018). 2018; Association for Computing Machinery: New York, USA; 827–40. https://doi.org/10.1145/3188745.3188814.
19. Resende J, Martins R, Antunes L. A survey on using Kolmogorov complexity in cybersecurity. Entropy 2019;21(12):1196. https://doi.org/10.3390/e21121196.
20. Menéndez HD, Llorente JL. Mimicking anti-viruses with machine learning and entropy profiles. Entropy 2019;21(5):513. https://doi.org/10.3390/e21050513.
21. Alshahwan N, Barr ET, Clark D, et al. Detecting malware with information complexity. Entropy 2020;22(5):575. https://doi.org/10.3390/e22050575.
22. Donahue M, Horner DJ, Cavagnolo KW, et al. Entropy profiles in the cores of cooling flow clusters of galaxies. Astrophys J 2006;643(2):730. https://doi.org/10.1086/503270.
23. Osswald PJ, del Rosario M, Garche J, et al. Fast and accurate measurement of entropy profiles of commercial lithium-ion

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

cells. Electrochim Acta 2015;177:270–6. https://doi.org/10.1016/j.electacta.2015.01.191.
24. Allison L, Edgoose T, Dix TI. Compression of strings with approximate repeats. In: ISMB. In Proceedings of the Sixth Int. Conf. on Intelligent Systems for Molecular Biology, Association for the Advancement of Artificial Intelligence (AAAI). 1998; 8–16.
25. Rivals E, Delgrange O, Delahaye JP, et al. Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. Bioinformatics 1997;13(2):131–36. https://doi.org/10.1093/bioinformatics/13.2.131.
26. Crochemore M, Vérin R. Zones of low entropy in genomic sequences. Comput Chem 1999;23(3–4):275–82. https://doi.org/10.1016/S0097-8485(99)00009-1.
27. Allison L, Stern L, Edgoose T, et al. Sequence complexity for biological sequence analysis. Comput Chem 2000;24(1):43–55. https://doi.org/10.1016/S0097-8485(00)80006-6.
28. Dix TI, Powell DR, Allison L, et al. Exploring long DNA sequences by information content. In: Probabilistic Modeling and Machine Learning in Structural and Systems Biology, Workshop Proceedings.RousuJ.,T.I.. Dix,KaskiS., et al. (eds.), Helsinki, Finland; 2006:97–102.
29. Wu C, Chen J, Liu Y, et al. Improved prediction of regulatory element using hybrid abelian complexity features with DNA sequences. Int J Mol Sci 2019;20(7):1704. https://doi.org/10.3390/ijms20071704.
30. Pinho AJ, Pratas D, Ferreira PJ, et al. Symbolic to numerical conversion of DNA sequences using finite-context models. In: 2011 19th European Signal Processing Conference. IEEE: Barcelona, Spain; 2011:2024–8.
31. Hosseini M, Pratas D, Pinho AJ. On the role of inverted repeats in DNA sequence similarity. In: 11th International Conference on Practical Applications of Computational Biology and Bioinformatics.Fdez-RiverolaF.,MohamadM.,RochaM. , et al., (eds.), Advances in Intelligent Systems and Computing. Springer: Cham; 2017:616: 228–36. https://doi.org/10.1007/978-3-319-60816-7_28.
32. Pinho AJ, Neves AJ, Afreixo V, et al. A three-state model for DNA protein-coding regions. IEEE Trans Biomed Eng 2006;53(11):2148–55. https://doi.org/10.1109/TBME.2006.879477.
33. Chao A, Jost L. Estimating diversity and entropy profiles via discovery rates of new species. Methods Ecol Evol 2015;6(8):873–82. https://doi.org/10.1111/2041-210X.12349.
34. Lai WJC, Kayedkhordeh M, Cornell EV, et al. mRNAs and lncRNAs intrinsically form secondary structures with short end-to-end distances. Nat Commun 2018;9(1):1–11. https://doi.org/10.1038/s41467-017-02088-w.
35. Ermolenko DN, Mathews DH. Making ends meet: new functions of mRNA secondary structure. Wiley Interdiscip Rev RNA 2021;12(2):e1611. https://doi.org/10.1002/wrna.1611.
36. Hosseini M, Pratas D, Morgenstern B, et al. Smash++: an alignment-free and memory-efficient tool to find genomic rearrangements. Gigascience 2020;9(5):giaa048. https://doi.org/10.1093/gigascience/giaa048.
37. Jiang Y, Adjeroh DA, Jiang BH, et al. J_2^*: A new method for alignment-free sequence similarity measurement. In: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE: Seoul, Korea (South); 2020:2209–14. https://doi.org/10.1109/BIBM49941.2020.9313338.
38. Pratas D, Toppinen M, Pyöriä L, et al. A hybrid pipeline for reconstruction and analysis of viral genomes at multi-organ level. Gigascience 2020;9(8):giaa086. https://doi.org/10.1093/gigascience/giaa086.
39. Troyanskaya OG, Arbell O, Koren Y, et al. Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity. Bioinformatics 2002;18(5):679–88. https://doi.org/10.1093/bioinformatics/18.5.679.
40. Dickson ZW, Golding GB. Low complexity regions in mammalian proteins are associated with low protein abundance and high transcript abundance. Mol Biol Evol 2022;39(5):msac087. https://doi.org/10.1093/molbev/msac087.
41. Enright JM, Dickson ZW, Golding GB. Low complexity regions in proteins and DNA are poorly correlated. Mol Biol Evol 2023;40(4):msad084. https://doi.org/10.1093/molbev/msad084.
42. Dickson ZW, Hackenberger D, Kuch M, et al. Probe design for simultaneous, targeted capture of diverse metagenomic targets. Cell Rep Methods 2021;1(6):100069. https://doi.org/10.1016/j.crmeth.2021.100069.
43. Escalona M, Rocha S, Posada D. A comparison of tools for the simulation of genomic next-generation sequencing data. Nat Rev Genet 2016;17(8):459–69. https://doi.org/10.1038/nrg.2016.57.
44. Posada D. CellCoal: coalescent simulation of single-cell sequencing samples. Mol Biol Evol 2020;37(5):1535–42. https://doi.org/10.1093/molbev/msaa025.
45. Huang PS, Boyken SE, Baker D. The coming of age of de novo protein design. Nature 2016;537(7620):320–7. https://doi.org/10.1038/nature19946.
46. Yang KK, Wu Z, Arnold FH. Machine-learning-guided directed evolution for protein engineering. Nat Methods 2019;16(8):687–94. https://doi.org/10.1038/s41592-019-0496-6.
47. Wu Z, Johnston KE, Arnold FH, et al. Protein sequence design with deep generative models. Curr Opi Chem Biol 2021;65:18–27. https://doi.org/10.1016/j.cbpa.2021.04.004.
48. Feldkamp U, Saghafi S, Banzhaf W, et al. DNASequenceGenerator: a program for the construction of DNA sequences. In: International Workshop on DNA-Based Computers.JonoskaN.,SeemanN.C., (eds.) DNA Computing. DNA 2001. Lecture Notes in Computer Science Springer: Berlin, Heidelberg. 2001:23–32. https://doi.org/10.1007/3-540-48017-X_3.
49. Ponty Y, Termier M, Denise A. GenRGenS: software for generating random genomic sequences and structures. Bioinformatics 2006;22(12):1534–5. https://doi.org/10.1093/bioinformatics/btl113.
50. Rouchka EC, Hardin CT. rMotifGen: random motif generator for DNA and protein sequences. BMC Bioinformatics 2007;8(1):1–8. https://doi.org/10.1186/1471-2105-8-292.
51. Chen GK, Marjoram P, Wall JD. Fast and flexible simulation of DNA sequence data. Genome Res 2009;19(1):136–42. https://doi.org/10.1101/gr.083634.108.
52. Pratas D, Bastos CA, Pinho AJ, et al. DNA synthetic sequences generation using multiple competing Markov models. In: 2011 IEEE Statistical Signal Processing Workshop (SSP). IEEE Statistical Signal Processing Workshop (SSP): Nice, France. 2011:133–136. https://doi.org/10.1109/SSP.2011.5967639.
53. Angermueller C, Dohan D, Belanger D, et al. Model-based reinforcement learning for biological sequence design. In: International Conference on Learning Representations. 2020. https://openreview.net/forum?id=HklxbgBKvr.
54. Almeida JR, Pinho AJ, Oliveira JL, et al. GTO: a toolkit to unify pipelines in genomic and proteomic research. SoftwareX 2020;12:100535. https://doi.org/10.1016/j.softx.2020.100535.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

55. Rong R, Jiang S, Xu L, et al. MB-GAN: Microbiome Simulation via Generative Adversarial Network. Gigascience 2021;10(2):giab005. https://doi.org/10.1093/gigascience/giab005.
56. Huang W, Li L, Myers JR, et al. ART: a next-generation sequencing read simulator. Bioinformatics 2012;28(4):593–4. https://doi.org/10.1093/bioinformatics/btr708.
57. Escalona M, Rocha S, Posada D. NGSphy: phylogenomic simulation of next-generation sequencing data. Bioinformatics 2018;34(14):2506–7. https://doi.org/10.1093/bioinformatics/bty146.
58. Flynn JM, Hubley R, Goubert C, et al. RepeatModeler2 for automated genomic discovery of transposable element families. Proc Natl Acad Sci 2020;117(17):9451–7. https://doi.org/10.1073/pnas.1921046117.
59. Smit AFA, Hubley R, Green P. RepeatMasker Open-4.0. http://www.repeatmasker.org . Accessed 2 January 2023.
60. Grumbach S, Tahi F. Compression of DNA sequences. In: [Proceedings] DCC93: Data Compression Conference. IEEE: Snowbird, UT, USA; 1993:340–50. https://doi.org/10.1109/DCC.1993.253115.
61. Grumbach S, Tahi F. A new challenge for compression algorithms: genetic sequences. Inf Proc Manage 1994;30(6):875–86. https://doi.org/10.1016/0306-4573(94)90014-0.
62. Manzini G, Rastero M. A simple and fast DNA compressor. Software Pract Experience 2004;34(14):1397–411. https://doi.org/10.1002/spe.619.
63. Cherniavsky N, Ladner R. Grammar-based compression of DNA sequences. DIMACS Working Group Burrows-Wheeler Transform 2004;21:2004.
64. Korodi G, Tabus I. An efficient normalized maximum likelihood algorithm for DNA sequence compression. ACM Trans Inform Syst 2005;23(1):3–34. https://doi.org/10.1145/1055709.1055711.
65. Cao MD, Dix TI, Allison L, et al. A simple statistical algorithm for biological sequence compression. In: 2007 Data Compression Conference (DCC'07). IEEE: Snowbird, UT, USA; 2007:43–52. https://doi.org/10.1109/DCC.2007.7.
66. Mishra KN, Aaggarwal A, Abdelhadi E, et al. An efficient horizontal and vertical method for online DNA sequence compression. Int J Comput Appl 2010;3(1):39–46. https://doi.org/10.5120/757-954.
67. Rajeswari R, Apparao A. GENBIT Compress-Algorithm for repetitive and non repetitive DNA sequences. Int J Comput Sci Inf Technol 2010;2(3):25–29.
68. Gupta A, Agarwal S. A novel approach for compressing DNA sequences using semi-statistical compressor. Int J Comput Appl 2011;33(3):245–51. https://doi.org/10.2316/Journal.202.2011.3.202-3114.
69. Pinho AJ, Pratas D, Ferreira PJ. Bacteria DNA sequence compression using a mixture of finite-context models. In: 2011 IEEE Statistical Signal Processing Workshop (SSP). IEEE: Nice, France. 2011:125–8. https://doi.org/10.1109/SSP.2011.5967637.
70. Pinho AJ, Ferreira PJ, Neves AJ, et al. On the representability of complete genomes by multiple competing finite-context (Markov) models. PLoS One 2011;6(6):e21588. https://doi.org/10.1371/journal.pone.0021588.
71. Pratas D, Pinho AJ, Ferreira PJSG. Efficient compression of genomic sequences. In: 2016 Data Compression Conference (DCC). IEEE: Snowbird, UT, USA. 2016:231–40. https://doi.org/10.1109/DCC.2016.60.
72. Kryukov K, Ueda MT, Nakagawa S, et al. Nucleotide Archival Format (NAF) enables efficient lossless reference-free compression of DNA sequences. Bioinformatics 2019;35(19):3826–8. https://doi.org/10.1093/bioinformatics/btz144.
73. Liu Y, Wong L, Li J. Allowing mutations in maximal matches boosts genome compression performance. Bioinformatics 2020;36(18):4675–81. https://doi.org/10.1093/bioinformatics/btaa572.
74. Grabowski S, Kowalski TM. MBGC: Multiple Bacteria Genome Compressor. Gigascience 2022;11:giab099. https://doi.org/10.1093/gigascience/giab099.
75. Deorowicz S, Danek A, Li H. AGC: compact representation of assembled genomes with fast queries and updates. Bioinformatics 2023;39(3):btad097. https://doi.org/10.1093/bioinformatics/btad097.
76. Silva M, Pratas D, Pinho AJ. Efficient DNA sequence compression with neural networks. Gigascience 2020;9(11):giaa119. https://doi.org/10.1093/gigascience/giaa119.
77. Silva M, Pratas D, Pinho AJ. AC2: an efficient protein sequence compression tool using artificial neural networks and cache-hash models. Entropy 2021;23(5):530. https://doi.org/10.3390/e23050530.
78. Kryukov K, Ueda MT, Nakagawa S, et al. Sequence Compression Benchmark (SCB) database—a comprehensive evaluation of reference-free compressors for FASTA-formatted sequences. Gigascience 2020;9(7):giaa072. https://doi.org/10.1093/gigascience/giaa072.
79. Pratas D, Hosseini M, Pinho AJ. GeCo2: an optimized tool for lossless compression and analysis of DNA sequences. In: Practical Applications of Computational Biology and Bioinformatics, 13th International Conference. PACBB 2019. Advances in Intelligent Systems and Computing. Springer: Cham.FdezRiverolaF.,RochaM.,MohamadM. , et al. 2020; 1005: 137–45. https://doi.org/10.1007/978-3-030-23873-5_17.
80. Pinho AJ, Neves AJ, Martins DA, et al. Finite-context models for DNA coding. Signal Process IntechOpen: Rijeka. MironS 2010; 117–30. https://doi.org/10.5772/8539.
81. Carvalho JM, Brás S, Pratas D, et al. Extended-alphabet finite-context models. Pattern Recognit Lett 2018;112:49–55. https://doi.org/10.1016/j.patrec.2018.05.026.
82. Pratas D, Hosseini M, Pinho AJ. Substitutional tolerant Markov models for relative compression of DNA sequences. In: 11th International Conference on Practical Applications of Computational Biology & Bioinformatics. Advances in Intelligent Systems and Computing PACBB.Springer: Cham.FdezRiverolaF.,MohamadM.,RochaM. , et al.(eds.), 2017:265–272. https://doi.org/10.1007/978-3-319-60816-7_32.
83. Silva JM, Qi W, Pinho AJ, et al. AlcoR Code Repository. 2023. https://github.com/cobilab/alcor. Accessed 20 November 2023.
84. Silva JM, Qi W, Pinho AJ, et al.. AlcoR website. 2023. https://cobilab.github.io/alcor. Accessed 20 November 2023.
85. Williams HE. Effective query filtering for fast homology searching. In: Pac Symp Biocomput 1999:214–25. PMID: 10380199.
86. Morgulis A, Gertz EM, Schäffer AA, et al. A fast and symmetric DUST implementation to mask low-complexity DNA sequences. J Comput Biol 2006;13(5):1028–40. https://doi.org/10.1089/cmb.2006.13.1028.
87. Wootton JC, Federhen S. Statistics of local complexity in amino acid sequences and sequence databases. Comput Chem 1993;17(2):149–63. https://doi.org/10.1016/0097-8485(93)85006-X.
88. Shin SW, Kim SM. A new algorithm for detecting low-complexity regions in protein sequences. Bioinformatics

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

2005;21(2):160–70. https://doi.org/10.1093/bioinformatics/bth497.
89. Rangwala SH, Kuznetsov A, Ananiev V, et al. Accessing NCBI data using the NCBI sequence viewer and genome data viewer (GDV). Genome Res 2021;31(1):159–69. https://doi.org/10.1101/gr.266932.120.
90. Morgulis A, Gertz EM, Schäffer AA, et al. WindowMasker: window-based masker for sequenced genomes. Bioinformatics 2006;22(2):134–41. https://doi.org/10.1093/bioinformatics/bti774.
91. Kroese DP, Taimre T, Botev ZI. Handbook of Monte Carlo Methods, Wiley Series in Probability and Statistics. John Wiley and Sons: New York. 2011.
92. Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27(3):379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
93. Išerić H, Alkan C, Hach F, et al. Fast characterization of segmental duplication structure in multiple genome assemblies. Algorithm Mol Biol 2022;17(1):1–15. https://doi.org/10.1186/s13015-022-00205-z.
94. Mouakkad-Montoya L, Murata MM, Sulovari A, et al. Quantitative assessment reveals the dominance of duplicated sequences in germline-derived extrachromosomal circular DNA. Proc Natl Acad Sci USA 2021;118(47): e2102842118. https://doi.org/10.1073/pnas.2102842118.
95. Camacho C, Coulouris G, Avagyan V, et al. BLAST+: architecture and applications. BMC Bioinformatics 2009;10(1):1–9. https://doi.org/10.1186/1471-2105-10-421.
96. Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool. J Mol Biol 1990;215(3):403–10. https://doi.org/10.1016/S0022-2836(05)80360-2.
97. Pratas D, Pinho AJ. On the approximation of the Kolmogorov complexity for DNA sequences. In: Iberian Conference on Pattern Recognition and Image Analysis. Springer: Cham.AlexandreL.,Salvador SánchezJ.,RodriguesJ.(eds.), 2017:10255:259–66. https://doi.org/10.1007/978-3-319-58838-4_29.
98. Nurk S, Koren S, Rhie A, et al. The complete sequence of a human genome. Science 2022;376(6588):44–53. https://doi.org/10.1126/science.abj6987.
99. Vollger MR, Guitart X, Dishuck PC, et al. Segmental duplications and their variation in a complete human genome. Science 2022;376(6588):eabj6965. https://doi.org/10.1126/science.abj6965.
100. Miga KH. Centromere studies in the era of 'telomere-to-telomere' genomics. Exp Cell Res 2020;394(2):112127. https://doi.org/10.1016/j.yexcr.2020.112127.
101. Altemose N, Logsdon GA, Bzikadze AV, et al. Complete genomic and epigenetic maps of human centromeres. Science 2022;376(6588):eabl4178. https://doi.org/10.1126/science.abl4178.
102. Suzuki Y, Myers EW, Morishita S. Rapid and ongoing evolution of repetitive sequence structures in human centromeres. Sci Adv 2020;6(50):eabd9230. https://doi.org/10.1126/sciadv.abd9230.
103. Aganezov S, Yan SM, Soto DC, et al. A complete reference genome improves analysis of human genetic variation. Science 2022;376(6588):eabl3533. https://doi.org/10.1126/science.abl3533.
104. Cuacos M, H. Franklin FC, Heckmann S. Atypical centromeres in plants—what they can tell us. Front Plant Sci 2015;6:913. https://doi.org/10.3389/fpls.2015.00913.
105. Vinga S, Almeida J. Alignment-free sequence comparison—a review. Bioinformatics 2003;19(4):513–23. https://doi.org/10.1093/bioinformatics/btg005.
106. Reinert G, Chew D, Sun F, et al. Alignment-free sequence comparison (I): statistics and power. J Comput Biol 2009;16(12):1615–34. https://doi.org/10.1089/cmb.2009.0198.
107. Wan L, Reinert G, Sun F, et al. Alignment-free sequence comparison (II): theoretical power of comparison statistics. J Comput Biol 2010;17(11):1467–90. https://doi.org/10.1089/cmb.2010.0056.
108. Zielezinski A, Vinga S, Almeida J, et al. Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol 2017;18(1):1–17. https://doi.org/10.1186/s13059-017-1319-7.
109. Zielezinski A, Girgis HZ, Bernard G, et al. Benchmarking of alignment-free sequence comparison methods. Genome Biol 2019;20(1):1–18. https://doi.org/10.1186/s13059-019-1755-7.
110. Morgenstern B. Sequence comparison without alignment: the SpaM approaches. In: Multiple Sequence Alignment. Springer: Humana, New York, NYKatohK.(ed.), Methods in Molecular Biology, 2021:121–34. https://doi.org/10.1007/978-1-0716-1036-7_8.
111. Leimeister CA, Schellhorn J, Dörrer S, et al. Prot-SpaM: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. Gigascience 2019;8(3):giy148. https://doi.org/10.1093/gigascience/giy148.
112. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biol 2019;20(1):1–13. https://doi.org/10.1186/s13059-019-1891-0.
113. Pratas D, Hosseini M, Grilo G, et al. Metagenomic composition analysis of an ancient sequenced polar bear jawbone from Svalbard. Genes 2018;9(9):445. https://doi.org/10.3390/genes9090445.
114. Mc Cartney AM, Shafin K, Alonge M, et al. Chasing perfection: validation and polishing strategies for telomere-to-telomere genome assemblies. Nat Methods. 2022; 19:687–95. https://doi.org/10.1038/s41592-022-01440-3.
115. Treangen TJ, Salzberg SL. Repetitive DNA and next-generation sequencing: computational challenges and solutions. Nat Rev Genet 2012;13(1):36–46. https://doi.org/10.1038/nrg3117.
116. Bodelon C, Untereiner ME, Machiela MJ, et al. Genomic characterization of viral integration sites in HPV-related cancers. Int J Cancer 2016;139(9):2001–11. https://doi.org/10.1002/ijc.30243.
117. Cantalupo PG, Katz JP, Pipas JM. Viral sequences in human cancer. Virology 2018;513:208–16. https://doi.org/10.1016/j.virol.2017.10.017.
118. Pischedda E, Crava C, Carlassara M, et al. ViR: a tool to solve intrasample variability in the prediction of viral integration sites using whole genome sequencing data. BMC Bioinformatics 2021;22(1):1–15. https://doi.org/10.1186/s12859-021-03980-5.
119. Cameron DL, Jacobs N, Roepman P, et al. VIRUSBreakend: viral integration recognition using single breakends. Bioinformatics 2021;37(19):3115–9. https://doi.org/10.1093/bioinformatics/btab343.
120. Chen X, Kost J, Li D. Comprehensive comparative analysis of methods and software for identifying viral integrations. Brief Bioinform 2019;20(6):2088–97. https://doi.org/10.1093/bib/bby070.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

121. Pyöriä L, Pratas D, Toppinen M, et al. Unmasking the tissue-resident eukaryotic DNA virome in humans. Nucleic Acids Res 2023; 51(7):3223–39. https://doi.org/10.1093/nar/gkad199.

122. Silva JM, Qi W, Pinho AJ, et al. Supporting data for "AlcoR: Alignment-Free Simulation, Mapping, and Visualization of Low-Complexity Regions in Biological Data." GigaScience Database. 2023. http://dx.doi.org/10.5524/102465.

Downloaded from https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giad101/7471725 by ETH Zürich user on 20 February 2024

Received: June 28, 2023. **Revised:** September 29, 2023. **Accepted:** November 7, 2023
© The Author(s) 2023. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.