BISER: Fast Characterization of Segmental Duplication Structure in Multiple Genome Assemblies

Hamza Išerić
Department of Computer Science, University of Victoria, Canada

Can Alkan
Department of Computer Engineering, Bilkent University, Ankara, Turkey

Faraz Hach
Vancouver Prostate Centre, Canada

Ibrahim Numanagić ✉
Department of Computer Science, University of Victoria, Canada

Abstract

The increasing availability of high-quality genome assemblies raised interest in the characterization of genomic architecture. Major architectural parts, such as common repeats and segmental duplications (SDs), increase genome plasticity that stimulates further evolution by changing the genomic structure. However, optimal computation of SDs through standard local alignment algorithms is impractical due to the size of most genomes. A cross-genome evolutionary analysis of SDs is even harder, as one needs to characterize SDs in multiple genomes and find relations between those SDs and unique segments in other genomes. Thus there is a need for fast and accurate algorithms to characterize SD structure in multiple genome assemblies to better understand the evolutionary forces that shaped the genomes of today.

Here we introduce a new tool, BISER, to quickly detect SDs in multiple genomes and identify elementary SDs and core duplicons that drive the formation of such SDs. BISER improves earlier tools by (i) scaling the detection of SDs with low homology (75%) to multiple genomes while introducing further 8–24x speed-ups over the existing tools, and by (ii) characterizing elementary SDs and detecting core duplicons to help trace the evolutionary history of duplications to as far as 90 million years.

2012 ACM Subject Classification Applied computing → Bioinformatics

Keywords and phrases genome analysis, fast alignment, segmental duplications, core duplicons, sequence decomposition

Digital Object Identifier 10.4230/LIPIcs.WABI.2021.15

Supplementary Material BISER is implemented in Seq and C++ programming languages:
Software (Source Code): `https://github.com/0xTCG/biser`
archived at `swh:1:dir:0659879a5b71885efcca60618f9888ee6a5533c6`

Funding H.I. and I.N. were supported by the National Science and Engineering Council of Canada (NSERC) Discovery Grant (RGPIN-04973) and the Canada Research Chairs Program. F.H. was supported by NSERC Discovery Grant (RGPIN-05952) and Michael Smith Foundation for Health Research (MSFHR) Scholar Award (SCH-2020-0370).

Acknowledgements We thank Haris Smajlović for his invaluable comments and suggestions during the manuscript preparation.

© Hamza Išerić, Can Alkan, Faraz Hach, and Ibrahim Numanagić;
licensed under Creative Commons License CC-BY 4.0

21st International Workshop on Algorithms in Bioinformatics (WABI 2021).
Editors: Alessandra Carbone and Mohammed El-Kebir; Article No. 15; pp. 15:1–15:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Segmental duplications (SDs), also known as low-copy repeats, are genomic segments larger than 1 Kbp that are duplicated one or more times in a given genome with a high level of homology [8]. While nearly all eukaryotic genomes harbour SDs, it is the human genome that exhibits the largest diversity of SDs. At least 6% of the human genome is covered by SDs ranging from 1 Kbp to a few megabases [8]. The architecture of human SDs also differs from other mammalian species both in its complexity and frequency [27]. For example, while most species harbour tandem SDs, the human genome is repleted with interspersed (both intra- and inter-chromosomal) SD blocks [10]. Human SDs are also often duplicated multiple times within the genome, often immediately next to, or even within an already existing SD cluster. This complex duplication architecture points to a major role that SDs play in human evolution [9, 7, 31]. Human SDs also introduce a significant level of genomic instability that results in increased susceptibility to various diseases [5, 19]. This has led to evolutionary adaptation in the shape of genes and transcripts unique to the human genome that aim to offset the effects of such instability [15]. Finally, SDs display significant diversity across different human populations and can be used as one of the markers for population genetics studies [38].

In order to understand the architecture and evolution of eukaryotic SDs, the first step is detecting all SDs within a given genome. However, SD detection is a computationally costly problem. The theoretically optimal solution to this problem – a local alignment of the entire genome to itself – is unfeasible due to large sizes of eukaryotic genomes that render the classical quadratic time algorithms such as Smith-Waterman impractical. Furthermore, the homology levels between SD copies – as low as 75% – prevent the use of the available edit distance approximations with theoretical guarantees [4, 20]. This is likely to remain so due to the sub-quadratic inapproximability of edit distance metrics [6]. The vast majority of sequence search and whole-genome alignment tools that rely on heuristics to compute the local alignments, such as MUMmer [32] and BLAST [2], also assume high levels of identity between two sequences and therefore are not able to efficiently find evolutionarily older SD regions. Even specialized aligners for noisy long reads, such as Minimap2 [30] or MashMap [24], cannot handle 75% homology that is lower than the expected noise of long reads (up to 15%, although sequencing error rates are improved recently to 5%) [3]. Finally, even if we use higher homology thresholds (such as 90%) to define an SD, the presence of low-complexity repeats and the complex SD rearrangement architecture often prevents the off-the-shelf use of the existing search and alignment tools for detecting SDs.

For these reasons, only a few SD detection tools have been developed in the last two decades, and most of them employ various heuristics and workarounds – often without any theoretical guarantees – to quickly find a set of acceptable SDs. The gold standard for SD detection, Whole-Genome Assembly Comparison (WGAC), uses various techniques such as hard-masking and alignment chunking to find SDs [8]. While its output is used as the canonical set of SDs in the currently available genomes, and as such forms the basis of the vast majority of SD analysis studies, WGAC can only find recent or highly conserved SDs (i.e. SDs with $>$ 90% homology) within primate lineages. Furthermore, WGAC requires specialized hardware to run and takes several days to complete. Few other tools developed as a replacement for WGAC – namely SDdetector [13] and ASGART [14] – are also limited in their ability to find SDs with less than 90% homology. Currently, the only tools that are able to detect SDs with lower homology are SDquest [34] and SEDEF [33]. SEDEF exploits the unique biological properties of the SD evolutionary process and combines it with a Poisson error model and MinHash approximation scheme, previously used for long-read alignment [24], to quickly find SDs even with 75% homology – thus enabling the study of SDs as old as 90 million years [33], – while also providing basic theoretical guarantees about the sensitivity of the search process. SDquest, on the other hand, relies on k-mer counting to find seed SD regions that are later extended and aligned with LASTZ [21].

It is important to point out that an SD is often formed by copying parts of older, more ancient SDs to a different location. This, in turn, implies that each SD can be decomposed into a set of short building blocks, where each block either stems from an ancient SD or a newly copied genomic region. Such building blocks are called "elementary SDs" [27]. Elementary SDs are often shared across related species within the same evolutionary branch. It has been proposed that the whole SD formation process is evolutionarily driven by a small subset of elementary SDs, often dubbed *seeds* or *core duplicons*, and that every SD harbours at least one such core duplicon [27]. Based on their cores, SDs can be hierarchically clustered into distinct clades. For example, the human genome SDs can be divided into 435 duplicon blocks that are further classified into 24 clades, seeded by a set of core duplicons with a total span of 2 Mbps that is often gene-rich and transcriptionally active. The prime example of a mosaic-like recombination region that is seeded by an SD core is the *LCR16* locus of the human genome that is shared with many other primates [10].

The SD evolutionary history analysis and the detection of core duplicons require a joint analysis of SDs in many related species. However, while existing SD tools are able to find SDs in single genomes in a reasonable amount of time, none of them can scale – at least not efficiently – to multiple genome assemblies. Furthermore, no publicly available tool is able to provide a deeper understanding of SD evolutionary architecture or find core duplicons across different species, mostly due to the computational complexity of such analysis because of the large number of existing SDs within different species[1]. For these reasons, only a small subset of previously reported core duplicons was analyzed in-depth (e.g. *LCR16* cores), and often so by manually focusing on narrow genomic regions to make the analysis tractable [10]. This has prevented the emergence of a clearer picture of the SD evolution across different species, especially of those SDs that preclude the primate branch of the evolutionary tree.

Here we introduce BISER (**B**risk **I**nference of **S**egmental duplication **E**volutionary st**R**ucture), a new framework implemented in Seq [36] and C++ that is specifically developed to quickly detect SDs even at low homology levels (75%) across multiple related genomes. BISER is also able to infer the elementary and core duplicons and thus allow an evolutionary analysis of all SDs in a given set of related genomes. The key conceptual advances of BISER consist of a novel linear-time algorithm that can quickly detect regions that harbour SDs in a given set of genomes, and a new method for fast SD decomposition into a set of elementary SDs based on the union-find algorithm. BISER can discover, decompose and cluster SDs in the human genome in 60 CPU minutes – an 8× speed-up over SEDEF and 25× speed-up over SDquest – and analyze all shared SDs in seven primate genomes in less than 16 CPU hours, translating to 2.5 hours on a standard 8-core laptop computer. The flexibility of BISER will make it a useful tool for SD characterizations that will open doors towards a better understanding of the complex evolutionary architecture of these functionally important genomic events.

[1] The source code that was used for older analyses [27] is not publicly available. SDquest, on the other hand, is able to detect elementary SDs but only at the single genome level.

2 Methods

2.1 Preliminaries

Consider a genomic sequence $G = g_1 g_2 g_3 \dots g_{|G|}$ of length $|G|$ and alphabet $\Sigma = \{A, C, G, T, N\}$. Let $G_i = g_i \dots g_{i+n-1}$ be a substring of G of length n that starts at position i in G. To simplify the notation, the length is assumed to be n. We will use an explicit notation $G_{i:i+n}$ for a substring of length n starting at position i when a need arises. Let $s_1 \circ s_2$ represent a string concatenation of strings s_1 and s_2.

Segmental duplications are long, low-copy repeats generated during genome evolution over millions of years. Following such an event, different copies of a repeat get subjected to different sets of mutations, causing them to diverge from each other over time. Thus, it is necessary to introduce a similarity metric between two strings in order to detect SDs in a given genome. To that end, we use the Levenshtein's [28] *edit distance* metric $\mathcal{E}$ between two strings s and s' that measures the minimum number of edit operations (i.e., substitutions, insertions, and deletions at the single nucleotide level) in the alignment of s and s'. Let ℓ be the length of such alignment; it is clear that $\max(|s|, |s'|) \leq \ell \leq |s| + |s'|$. We can also define an *edit error* $\operatorname{err}(\cdot, \cdot)$ between s and s' (or, in the context of this paper, an *error*) as the normalized edit distance: $\operatorname{err}(s, s') = \mathcal{E}(s, s')/\ell$. Intuitively, this corresponds to the sequence divergence of s and s'. Now we can formally define an SD as follows:

▶ **Definition 1.** *A segmental duplication (SD) within the error threshold ε is a tuple of paralog sequences (G_i, G_j) that satisfies the following criteria:*

1. $\operatorname{err}(G_i, G_j) \leq \varepsilon$*;*
2. $\ell \geq 1{,}000$ *where ℓ is the length of the optimal alignment between G_i and G_j; and*
3. *the overlap between G_i and G_j is at most $\varepsilon \cdot n$* [2]*.*

Given a set of genomes $G^1, \dots, G^\gamma$ and their mutual evolutionary relationships, our goal is to:

- find a set of valid SDs, $\mathcal{SD}^i$, within each G^i **(SD detection)**;
- find all copies of both s and s' for $(s, s') \in \mathcal{SD}^i$ in other genomes $G^j, j \neq i$, if such copies exist **(SD cross-species conservation detection)**; and
- decompose each SD from $\mathcal{SD} = \mathcal{SD}^1 \cup \dots \cup \mathcal{SD}^\gamma$ into a set of *elementary SDs* E, and determine the set of core duplicons that drive the formation of SDs in $\mathcal{SD}$ **(SD decomposition)**.

To that end, we present BISER, a computational framework that is able to efficiently perform these steps, and we describe the algorithms behind it in the following sections. For the sake of clarity, unless otherwise noted, we assume that we operate on a single genome G. Since SDs are by definition different from low-complexity repeats and transposons, we also assume that all genomes $G^1, \dots, G^\gamma$ are hard-masked and, as such, do not contain such elements.

2.1.1 SD Error Model

Different paralogs of an SD are mutated independently of each other. Therefore, the sequence similarity of paralogs is correlated with the age of the duplication event – more recent copies are nearly identical, while distant ancestral copies are dissimilar. It has been proposed that

[2] Ideally, the SD mates should not overlap; however, due to the presence of errors, we need to account for at most $\varepsilon \cdot n$ overlap.

the sequence similarity of older SDs (e.g., those shared by the mouse and human genomes) falls as low as 75% [33]. In other words, the error between different copies of an old SDs exceeds 25% (i.e., $\text{err}(s, s') \geq 0.25$ for SD paralogs s and s', according the definition above).

Detection of duplicated regions within such a large error threshold is a challenging problem, as nearly any edit distance approximation technique with or without theoretical guarantees breaks down at such high levels of dissimilarity [4, 24], provided that this error is truly random. However, that is not the case: we have previously shown that the SD mutation process is an amalgamation of two independent mutation processes, namely the background point mutations (also known as *paralogous sequence variants*, or PSVs) and the large-scale block edits. As such, the overall error rate ε can be expressed as a sum of two independent error rates, ε_P (PSV mutation rate) and ε_B (block edit rate), where only ε_P is driven by a truly random mutation process.

In the case when paralogs share the 75% sequence identity, it has been shown that the random point mutations (PSVs) contribute at most 15% ($\varepsilon_P \leq 0.15$) towards the total error ε [33] (this also holds for many other mammalian genomes, as their substitution rate is often lower than the human substitution rate [16]). The remaining 10% is assumed to correspond to the block edit rate ε_B. Note that these mutations are clustered *block* errors and as such are randomly distributed across SD regions. The probability of a large block event is roughly 0.005 based on the analysis of existing SD calls.

On the other hand, we assume that PSVs between two SD paralogs s and s' follow a Poisson error model [24, 17], and that those mutations occur independently from each other. It follows that any k-mer in s' has accumulated on average $k \cdot \varepsilon_P$ mutations compared to the originating k-mer in s, provided that such k-mer was part of the original copy event. By setting a Poisson parameter $\lambda = k \cdot \varepsilon_P$, we obtain the probability of a duplication event in which a k-mer is preserved in both SD paralogs (i.e., that a k-mer is error-free) to be $e^{-k\varepsilon_P}$.

Let us return to the main problem of determining whether two strings s and s' are "similar enough" to be classified as SDs. As mentioned before, classical edit distance calculation algorithms would be too slow for this purpose. Instead, we use an indirect approach that measures the similarity of strings s and s' by counting the number of shared k-mers in their respective k-mer sets $\mathbf{K}(s)$ and $\mathbf{K}(s')$. It has been shown that Jaccard index ($\mathcal{J}(\mathbf{K}(s), \mathbf{K}(s')) = \frac{\mathbf{K}(s) \cap \mathbf{K}(s')}{\mathbf{K}(s) \cup \mathbf{K}(s')}$) is a good proxy for $\mathcal{E}(s, s')$ under the Poisson error model [24]. Thus we can combine the Poisson error model with the SD error model, and obtain the expected value of Jaccard index τ between any two strings s and s', whose edit error $\text{err}(s, s')$ follows the SD error model and is lower than $\varepsilon = \varepsilon_P + \varepsilon_B$ to be [33]:

$$\tau = \mathbb{E}[\mathcal{J}(\mathbf{K}(s), \mathbf{K}(s'))] \geq \frac{1 - \varepsilon_B}{1 + \varepsilon_B} \cdot \frac{1}{2e^{k\varepsilon_P} - 1}. \tag{1}$$

As we cannot use local alignment to efficiently enumerate all SDs in a given genome due to both time and space complexity, we utilize a heuristic approach to enumerate all pairs of regions in G that are likely to harbour one or more segmental duplications. We call these pairs *putative SDs*. These pairs are not guaranteed to contain a "true" SD and must be later aligned to each other to ascertain the presence of true SDs. Nevertheless, such an approach will *filter out* the regions that do not harbour SDs, and thus significantly reduce the amount of work needed for detecting "true" SDs. The overall performance of our method, both in terms of performance and sensitivity, will depend on how well the putative SDs are chosen.

The problem of putative SD detection can be, thanks to the SD error model, easily expressed as an instance of a filtering problem: find all pairs of indices i, j in G such that $\mathcal{J}(\mathbf{K}(G_i), \mathbf{K}(G_j)) \geq \tau$, where τ is the lower bound from the Equation 1. Here we assume that the size of G_i and G_j exceeds the SD length threshold (1,000 bp), and no k-mer occurs twice in either G_i or G_j.

2.2 SD Detection

Algorithm 1 Algorithm for finding putative SDs.

```
        Input  : Genomic sequence $G$, its $k$-mer index $I_G$, threshold $\Delta$.
        Output : A list $\mathcal{SD}$ of putative SDs.
        $L \leftarrow []$; $\mathcal{SD} \leftarrow []$;
        for $x \leftarrow 1$ to $|G|$ do
            $K \leftarrow I_G[G_{x:x+k}]$; $(i_K, i_L) \leftarrow (1, 1)$;
            append $(x, K_{i_K}, k)$ to $L$ if $L$ is empty;
            while $i_K \leq |K|$ and $i_L \leq |L|$ do
                $y \leftarrow K_{i_K}$; $(\ell_x, \ell_y, l) \leftarrow L_{i_L}$;
(1)             if $y < \ell_y$ then
                    insert $(x, y, k)$ to $L$ before $i_L$;
                    advance $i_K$ and $i_L$;
(2)             else if $y \geq \ell_y + l$ and $\max(x - \ell_x, y - \ell_y) - l \leq \Delta$ then
                    extend $L_{i_L}$ to cover $G_{x:x+k}$ and $G_{y:y+k}$;
                    increase counts for $\bigcup(L_{i_L})$ and $\bigcap(L_{i_L})$;
                    advance $i_K$; $i_L \leftarrow$CheckJaccard $(L_{i_L})$
(1')            else if $y \leq start(L_{i_L+1})$ then
                    insert $(x, y, k)$ to $L$ before $i_L$;
                    advance $i_K$ and $i_L$;
(3)             else
                    increase count for $\bigcup(L_{i_L})$;
                    $i_L \leftarrow$CheckJaccard $(L_{i_L})$
                end
            end
        end
```

(a)

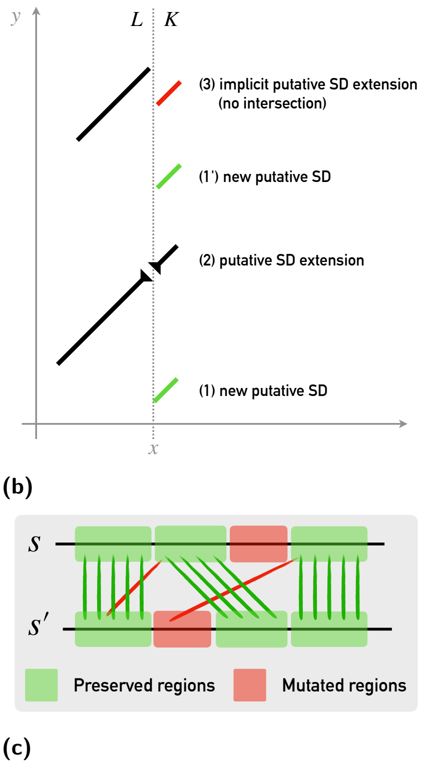

Figure 1 **(a)** A plane-sweep algorithm for finding putative SDs. **(b)** Visual guide for the algorithm. The algorithm sweeps a vertical dashed line through the set of winnowed k-mers in a genome G (represented by the x axis). At each k-mer starting at the location x, it queries the index I_G to obtain a sorted list K of k's occurrences in G (shown on the right side of the sweep line). The algorithm then scans K and the list L of putative SDs found thus far at the same time. At each step, it examines i_L-th element of L and i_K-th element of K, and decides whether to start a new putative SD (cases (1) and (1'), green k-mers on the right), extend the current putative SD with the current k-mer (case (2), black k-mer on the right), or subsume the current k-mer within the current putative SD (case (3), red k-mer). In all of these cases, the algorithm updates the counts of k-mer union $\bigcup_L$ and intersection $\bigcap_L$ for each processed putative SD (note that it only updates the counts and does not maintain the sets themselves). The `CheckJaccard` procedure marks each putative SD in L that satisfies the SD criteria as "good" and removes it from L when the extension is complete. **(c)** A visual representation of a valid k-mer matching in a valid alignment (shown by green lines). Any addition of a red matching to the set of green matchings would render the alignment invalid as red matchings are not co-linear with the green matchings (in other words, they "cross" the existing matchings).

As we cannot use local alignment to efficiently enumerate all SDs in a given genome due to both time and space complexity, we utilize a heuristic approach to enumerate all pairs of regions in G that are likely to harbour one or more segmental duplications. We call these pairs *putative SDs*. These pairs are not guaranteed to contain a "true" SD and must be later aligned to each other to ascertain the presence of true SDs. Nevertheless, such an approach will *filter out* the regions that do not harbour SDs, and thus significantly reduce the amount of work needed for detecting "true" SDs. The overall performance of our method, both in terms of performance and sensitivity, will depend on how well the putative SDs are chosen.

The problem of putative SD detection can be, thanks to the SD error model, easily expressed as an instance of a filtering problem: find all pairs of indices i, j in G such that that $\mathcal{J}(\mathbf{K}(G_i), \mathbf{K}(G_j)) \geq \tau$, where τ is the lower bound from the Equation 1. Here we assume that the size of G_i and G_j exceeds the SD length threshold (1,000 bp) and no k-mer occurs twice in either G_i or G_j.

The filtering approach has already been successfully used in other software packages and forms the backbone of both SEDEF (SD detection tool) and MashMap (Nanopore read aligner).

However, both methods maintain the k-mer sets $\mathbf{K}(s)$ and $\mathbf{K}(s')$ to calculate the Jaccard index between the sequences s and s'.

As these methods dynamically grow s and s' (as the length n is not known in advance), the corresponding sets $\mathbf{K}(s)$ and $\mathbf{K}(s')$ are constantly being updated, necessitating a costly recalculation of $\mathbf{K}(s) \cap \mathbf{K}(s')$ on each update. A common trick is to use the MinHash technique to reduce the sizes of $\mathbf{K}(s)$ and $\mathbf{K}(s')$, and thus the frequency of such updates. However, the frequent recalculation of the Jaccard index still remains a major bottleneck even in the MinHash-based approaches.

Here we note that the Jaccard index calculation can be significantly simplified by not having to maintain the complete k-mer sets $\mathbf{K}(s)$ and $\mathbf{K}(s')$. The need for keeping such sets arises from the fact that calculation of $\mathbf{K}(s) \cap \mathbf{K}(s')$ allows any k-mer in $\mathbf{K}(s')$ to match any k-mer in $\mathbf{K}(s)$. However, such a loose intersection requirement is not only redundant for approximation of edit distance under the SD error model but is even undesirable as such intersections can introduce a cross-over k-mer matches that are not possible in the edit distance metric space (see Figure 1c for an example of valid and invalid matchings).

By disallowing such cross-over cases, we can significantly optimize the calculation of the Jaccard index. Let us show how to do that without sacrificing sensitivity. Let us first introduce $s \circledast s'$ as an alternative way of measuring the k-mer similarity between strings s and s'.

For that purpose, let us introduce a notion of a *co-linear k-mer matching* between s and s' as a set of index pairs (i, j) $(1 \leq i \leq |s|, 1 \leq j \leq |s'|)$ such that the k-mers that start at i and j in s and s' respectively are equal, and such that all pairs (i, j) in a matching are co-linear (i.e. for each (i, j) and (i', j'), either $i < i'$ and $j < j'$, or $i > i'$ and $j > j'$).

A $\circledast$ operation describes the size of a maximum co-linear matching of k-mers between s and s'. In other words, we want to select a maximal set of matching k-mers in $\mathbf{K}(s)$ and $\mathbf{K}(s')$ such that no two k-mer matchings cross-over each other (see Figure 1c for an example of cross-over, or non-co-linear, matchings). We can replace $\mathbf{K}(s) \cap \mathbf{K}(s')$ with $s \circledast s'$ and introduce an *ordered Jaccard index* $\hat{\mathcal{J}}(s, s')$, formally defined as:

$$\hat{\mathcal{J}}(s, s') = \frac{s \circledast s'}{\mathbf{K}(s) \cup \mathbf{K}(s')}.$$

The following lemma allows us to use an ordered Jaccard index $\hat{\mathcal{J}}$ in lieu of classical Jaccard index $\mathcal{J}$:

▶ **Lemma 2.** *The ordered Jaccard index $\hat{\mathcal{J}}(s, s')$ of two strings s and s' is equal to the Jaccard index $\mathcal{J}(\mathbf{K}(s), \mathbf{K}(s'))$ (under the assumptions of SD error model, namely the separation of ϵ_B and ϵ_P), assuming that s and s' only share k-mers that have not been modified by PSVs following the originating copy event.*

Proof. It is sufficient to prove that the size of $|\mathbf{K}(s) \cap \mathbf{K}(s')|$ always corresponds to the size of maximal co-linear matching between s and s'.

To show that $s \circledast s' \leq |\mathbf{K}(s) \cap \mathbf{K}(s')|$, it is enough to note that matched k-mers in any matching are by definition identical, and thus belong to $\mathbf{K}(s) \cap \mathbf{K}(s')$.

We will prove that $s \circledast s' \geq |\mathbf{K}(s) \cap \mathbf{K}(s')|$ by contradiction. First, note that the string s is equal to s' immediately after the duplication event (i.e. before the occurrence of PSVs) and that all k-mers are co-linear in their maximal matching because s contains no repeated k-mers (an assumption made by the SD error model). Now, suppose that there is a cross-over in $\mathbf{K}(s) \cap \mathbf{K}(s')$. That implies either a cross-over between s and s' before PSVs occurred – contradicting the previous observation – or a cross-over after it, contradicting the assumption that any matched k-mer pair was matched before the occurrence of PSVs. Hence $\mathbf{K}(s) \cap \mathbf{K}(s')$ cannot contain any cross-overs, and $s \circledast s' = |\mathbf{K}(s) \cap \mathbf{K}(s')|$. ◀

If the conditions of Lemma 2 are satisfied, we can calculate $s \circledast s'$ in linear time by a simple scan through s and s' at the same time. A linear calculation of $s \circledast s'$, together with the fact that the lower bound τ in Equation 1 equally holds for $\hat{\mathcal{J}}$ as well (a consequence of Lemma 2), allows us to use a plane sweep technique to select all pairs of substrings (s, s') in G whose ordered Jaccard distance $\hat{\mathcal{J}}(s, s')$ exceeds τ, and as a result, select all putative SDs in G (see Figure 1 for details).

We begin by creating a k-mer index I_G that connects each k-mer in G to an ordered list of its respective locations in G. Then we sweep a vertical line in G from left to right while maintaining a sorted list L of putative SDs found thus far. For each location x in G encountered by a sweep line, we query I_G to obtain a sorted list K containing loci of $G_{x:x+k}$'s copies in G. Then, for any y in K, we check if it either (1) begins a new potential putative SD that maps x to y, (2) extends an existing putative SD, or (3) is covered by existing putative SD in L. If a putative SD in L is too distant from y, it is promoted to the final list of putative SD regions if it satisfies the ordered Jaccard index threshold τ and the other SD criteria from the Definition 1. Note that we do not allow a k-mer to extend a putative SD if the distance between it and the SD exceeds the user-defined threshold Δ (set to 250 by default). It takes $|L| + |K|$ steps to process each k-mer in G because both L and K are sorted. However, because the size of $|L|$ is kept low by the distance criteria, and because $|K|$ is low enough in practice, [3], the time complexity of Algorithm 1 is $O(|G|)$ for constructing the index I_G, and linear in terms of the genome size for plane sweeping.

The key assumption in Lemma 2 – that two paralogs only share the k-mers that have not been mutated since the copy event – does not always hold in practice on real data. As such, the Algorithm 1 might occasionally under-estimate the value of $\hat{\mathcal{J}}$, leading potentially to some false negatives. We control that by using Δ – the same parameter that controls the growth of putative SDs by limiting the maximum distance of neighbouring k-mers in $s \circledast s'$ (Figure 1) – to limit the growth of under-estimated SDs and thus start the growth of potentially more successful SDs earlier. This heuristic might cause a large SD to be reported as a set of smaller disjoint SD regions. For that reason, we post-process the set of putative SDs upon the completion of Algorithm 1 and merge together any two SDs that are close to each other if their union satisfies the ordered Jaccard index criteria. We also extend each putative SD by 5 Kbp both upstream and downstream to account for the small SD regions that might have been filtered out during the search process. This parameter is user-defined and might be adjusted for different genome assemblies.

[3] The average size of L in our experiments was 370, and the average size of K is 30.

The performance of the plane sweep technique can be further improved by winnowing the set of k-mers used for the construction of I_G [24]. Instead of indexing all k-mers in G, we only consider k-mers in a *winnowing fingerprint* $W(G)$ of G. $W(G)$ is calculated by sliding a window of size w through G and by taking in each window a lexicographically smallest k-mer (the rightmost k-mer is selected in case of a tie). The expected size of $W(G)$ for a random sequence G is $2|G|/(w+1)$ [35]. The main benefit of winnowing is that it can significantly speed up the search step (up to an order of magnitude) without sacrificing sensitivity. The winnow $W(G)$ can be computed in a streaming fashion in linear time using $O(w)$ space with the appropriate data structures [11].

Following the discovery of putative SDs, we locally align paralogs from each putative SD and only keep those regions whose size satisfies the SD criteria mentioned above. BISER uses a two-tiered local chaining algorithm described previously in SEDEF that uses a seed-and-extend approach and efficient $O(n \log n)$ chaining method following by a SIMD-parallelized sparse dynamic programming algorithm to calculate the boundaries of the final SD regions and their alignments [1, 30, 39].

2.3 SD Decomposition

Once the set of final SDs $\mathcal{SD} = \{(s_1, s_1'), \dots\}$ is discovered and the precise global alignment of each paralog pair $(s, s') \in \mathcal{SD}$ is calculated, we proceed by decomposing the set $\mathcal{SD}$ into a set of evolutionary building blocks called *elementary SDs*. More formally, we aim to find a minimal set of elementary SDs $E = \{e_1, \dots, e_{|E|}\}$, such that each SD paralog s is a concatenation of $\hat{e}_1^s \circ \cdots \circ \hat{e}_{n_s}^s$. Each $\hat{e}_i$ either belongs to E or there is some $e_j \in E$ such that $\text{err}(\hat{e}_i, e_j) \leq \varepsilon$. An example of such a decomposition is given in Figure 2.

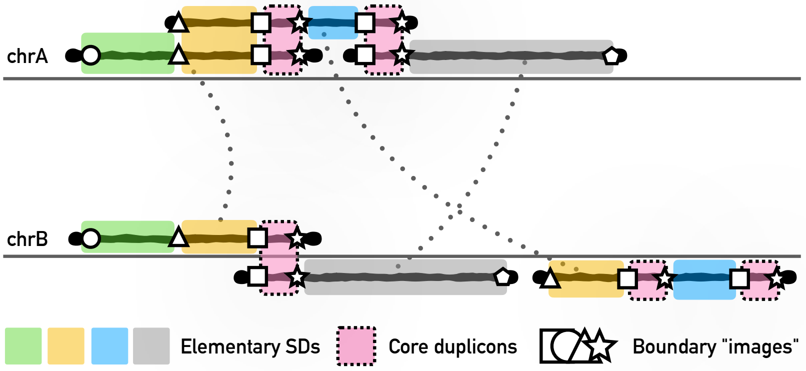

Figure 2 A decomposition of three partially overlapping SDs into a set of elementary SDs. Each paralog pair is indicated by a pair of two thick black lines linked by a dashed line. Each elementary SD is represented as a coloured box. The boxes of core duplicons – elementary SDs shared by all SDs – are depicted with a dashed border. Note that a boundary of each elementary SD is induced by a boundary of an existing SD. Different boundaries are represented by different shapes, and their images (paralog copies) also share the same shape. For the sake of simplicity, we only show the identifiers (shapes) for locations that define elementary SD boundaries.

Note that each locus covered by an SD paralog is either copied to another locus during the formation of that SD (in other words, it is "mirrored" by its paralog) or belongs to an alignment gap. As SD events can copy over the regions that already form an existing SD, a single locus might "mirror" a large number of existing locations. In order to find all locations that a locus i mirrors, we use a modification of Tarjan's union-find disjoint set algorithm [41] to link together all mirrored locations. This algorithm begins by giving each locus in a genome covered by an existing SD a distinct identifier (represented by a distinct shape in Figure 2). It then iterates over the set of SDs, and for each pair of SD paralogs (s, s') uses the global alignment between s and s' to link the identifiers of any two loci that are mirrored by the SD event. Linking merges the two identifiers into a single identifier. Upon the completion of the algorithm, all copies ("mirrors") of each locus will share the same identifier.

Note that the boundaries of SD paralogs, or their images, correspond to the boundaries of elementary SDs, as each paralog by definition starts and ends with an elementary SD (Figure 2). However, as the set of elementary SDs should be minimal, it is not only necessary but sufficient to focus only on the identifiers that belong to a boundary of an existing SD paralog in order to construct the set of elementary SDs E. These identifiers describe a set of locations in G that form the boundaries of elementary SDs. We can obtain the final set E by iterating over G and checking if a locus is identified with a boundary-covering identifier.

In practice, SD boundaries and SD alignments are highly uneven, and SDs themselves exhibit a complex mosaic structure that often introduces "mirror loops" [34] that can collapse multiple unrelated loci into a single identifier. BISER handles these cases by discarding any mirrored loci that lie within a close distance of an already existing elementary SD boundary.

After decomposing SDs into the set of elementary SDs E, we select some of them as *core duplicons*. We define these duplicons as the minimal set of elementary SDs that cover all existing SDs (an SD is covered by an elementary if either paralog is composed of that elementary SD). We use a classical set-cover approximation algorithm [12] to determine a set of core duplicons from E.

2.4 Multiple Genomes

The above method can be efficiently scaled to γ distinct genomes $G^1, \dots, G^\gamma$ by constructing a composite k-mer index $I_{G^1 \cup \dots \cup G^\gamma}$ and by running the search procedure on each G^i in parallel. We only need to ensure that the SD overlap criteria are enforced only if a k-mer belongs to the same genome that is currently being searched. Note that the size of the genome index grows sub-linearly with the number of genomes, as most genomes – especially mammalian ones – share the large number of common k-mers. Also, note that this method can be trivially extended to search for reverse-complemented SD copies by adding an additional iteration over $\bar{G}$, where $\bar{G}$ is a reverse complement of G.

The detection of cross-species conserved SD regions happens automatically when using a composite index $I_{G^1 \cup \dots \cup G^\gamma}$. However, as such procedure does not distinguish between conserved SDs and other conserved regions, BISER uses additional checks to ensure that every reported conservation is also a valid SD in at least one of the provided genomes.

3 Results

We have evaluated all stages of BISER for speed and accuracy on both simulated and real-data datasets. All results were obtained on a multi-core Intel Xeon 8260 CPU (2.40GHz) machine with 1 TB of RAM. The run times are rounded to the nearest minute and are reported for both single-core as well as multi-core (8 CPU cores) modes when ran in parallel via GNU Parallel [40]. All real-data genomes were hard-masked, and all basepair coverage statistics are provided with respect to the hard-masked genomes.

In our experiments, we used $k = 14$ when searching for putative SDs and $k = 10$ during the alignment step (note that both parameters are user-adjustable). The size of the winnowing window was set to 16. We found that the lower values of k significantly impact the running time without providing any visible improvement to the detection sensitivity, while higher values of k significantly lower the detection sensitivity.

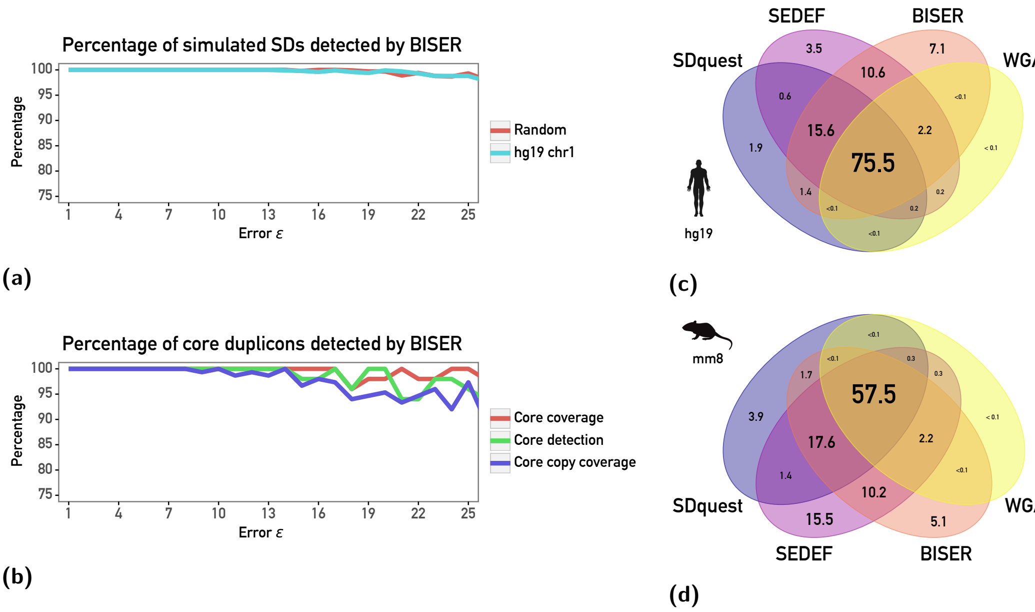

Figure 3 **(a)** Performance of BISER's algorithm on simulated SDs. x-axis represents the simulated SD error rate ε, while y axis represents the percentage of correctly detected SDs. Note that the plot area is cropped as BISER detects more than 98% of simulated SDs for any $\varepsilon \leq 0.25$. **(b)** Performance of BISER's core duplicon detection on simulated cores. The red line shows the percentage of ancestral core locations covered by a detected SD; the green line shows the percentage of cores identified as such; while the blue line shows the coverage of later core copies covered by a detected SD. Note that the plot area is also cropped. **(c)** Venn diagram depicts the SD coverage of the BISER, WGAC, SEDEF and SDquest (in Mbp) on the hard-masked human genome (hg19). **(d)** Venn diagram depicts the SD coverage of the BISER, WGAC, SEDEF and SDquest (in Mbp) on the hard-masked mouse genome (mm8). Note that nearly all bases out of ≈17 Mbp bases that are shown to be unique to SEDEF (and not covered by BISER) map to gaps and low-copy repeats and should be therefore treated as noise (not true SDs).

3.1 Simulations

3.1.1 SD detection

The accuracy of using the strong Jaccard index together with the SD error model as a function of error parameter ε, as well as the overall sensitivity of BISER's SD detection pipeline, was evaluated on a set of 1,000 simulated segmental duplications ranging from 1 to 100Kbp. All sequences and mutations were randomly generated with uniform distribution according to the SD error model with $\varepsilon \in \{0.01, 0.02, \ldots, 0.25\}$ (i.e., we allowed the overall error rate to reach 25%). We consider a simulated SD as being "covered" if BISER found an SD that covers more than 90% of the original SD's basepairs. As shown in Figure 3a, the overall sensitivity is around 99% even for $\varepsilon = 0.25$.

We performed the same experiment on human (hg19) chromosome 1 (Figure 3a), where we selected uniformly at random 10,000 sequences of various lengths and duplicated them within the chromosome. Each duplication was followed by introducing random PSVs according to the SD error model while varying the values of ε as described above. Even in this case, BISER's performance stays the same, and only a handful of very small SDs (of size ≈1,000) were missed.

3.1.2 Core duplicon detection

We also devised a simulation experiment to measure the power of BISER's core duplicon detection module. We began with a random DNA sequence of size 10 Mbp. Then we simulated an evolutionary process by introducing a set of novel SDs for 50 generations. In each generation, we introduced a novel core duplicon that does not overlap with an existing core, and we introduce a random number of SD events according to the SD error model that contains at least one core duplicon introduced thus far. In each iteration, we made sure that the difference between paralogs does not exceed $\varepsilon \in \{0.01, 0.02, \ldots, 0.25\}$ regardless of their age. We finally used BISER to analyze the final sequence to predict SDs and their elementary decomposition, as well as the core duplicons. BISER was able to successfully cover all ancestral cores, and properly decompose them into the elementary SDs and finally identify them as core duplicons (Figure 3b). Furthermore, BISER covered 95% or more of the more recent copies of the ancestral cores. We note that the combination of large block indels leads BISER to occasionally report a single ancestral core as a set of 2 or more core duplicons or to report an elementary SD that covers less than 90% of a core. These cases are rare and happen less than 8% of the time at the highest levels of ε. The sensitivity of core duplicon detection is further described in Figure 3b.

Table 1 Running time performance of BISER (single-core and 8-core mode) on Intel Xeon 8260 CPU at 2.40 GHz for single genomes (hg19 and mm8).

Single genome (hg19)

	Total	Putative SDs	Alignment	Decomposition
1 core	**1h 8m**	44m	24m	<1m
8 cores	**10m**	6m	4m	<1m

Single genome (mm8)

	Total	Putative SDs	Alignment	Decomposition
1 core	**1h 26m**	40m	46m	<1m
8 cores	**13m**	5m	8m	<1m

Table 2 Running time performance of BISER (single-core and 8-core mode) on Intel Xeon 8260 CPU at 2.40 GHz for seven genomes.

Seven genomes (see below)

	Total	Putative SDs	Alignment	Decomposition
1 core	**30h 23m**	11h 58m	18h 19m	6m
8 cores	**4h 30m**	1h 54m	2h 29m	6m

3.2 Single-genome results

We have run BISER on the *H.sapiens* hg19 genome and *M.musculus* mm8 genome, and compared it to the published WGAC [8][4], SEDEF [33] and SDquest [34] SD calls. We also compared the runtime performance of BISER to that of SEDEF and SDquest. Note that we were not able to run WGAC due to the lack of hardware necessary for its execution. We did not compare BISER to other SD detection tools – namely SDdetector [13], MashMap2 [25] and ASGART [14] – as it has been previously shown that these tools underperform when compared to SEDEF or SDquest, and require an order of magnitude more resources than either SEDEF or SDquest do (see [33] for the detailed comparisons with these tools). For the same reason, we did not compare BISER to whole-genome aligners such as Minimap2 [30] and MUMmer/nucmer [32], as well as DupMasker [26], as none of these tools were designed to detect *de novo* SDs in a genome.

BISER was able to find and align all SD regions in hg19 in 10 minutes on 8 cores (or ≈one hour on a single core) (Table 1). To put this into perspective, BISER is nearly 8× faster than SEDEF, 24× faster than SDquest, and an order of magnitude faster than WGAC that takes days to find human SDs (personal communication; we were not able to run WGAC pipeline ourselves due to legacy hardware requirements). As a side note, BISER has the same memory requirements as SEDEF or SDquest and needs around 5 GB of RAM per core. Since SEDEF by default operates on a genome that is not hard-masked, we also ran SEDEF on a hard-masked genome to measure its theoretical speed (note that SEDEF was not designed for hard-masked genomes; thus, the basepair analysis is omitted). SEDEF took 21 minutes on 8 CPU cores to process a hard-masked hg19, leaving it still >2× slower than BISER. Similar performance gains were observed on the mm8 genome as well.

In terms of sensitivity, BISER discovers about 1 GB of putative SD regions. After the alignment step, BISER reports 112 Mb of final SD regions within the 75% edit distance in hg19. That is 34 Mbp more than WGAC and 17 Mbp more than SDquest. The total coverage of SEDEF and BISER are similar to each other, differing by 3 Mbp uniquely detected by SEDEF and 7 Mbp uniquely covered by BISER (Table 3). BISER misses a few Mbp of SD regions unique to SDquest and a negligible amount unique to WGAC (Figure 3). On the mm8 genome, we can observe similar trends. However, we also observed that SEDEF covers roughly 17 Mbp not covered by BISER (Figure 3). When SEDEF is run on a hard-masked genome, it does not cover these bases; further analysis showed that nearly all bases (≥16.3 Mbp) originally reported as unique to SEDEF actually map either to alignment gaps, soft-masked repeat elements, or small islands (<200bp) between the low-copy repeats and as such do not constitute "true" SDs.

BISER found roughly ≈67,000 elementary SDs that describe hg19 SD calls. Of those, 2,759 were identified as core duplicons. BISER's core duplicons cover all 100 of the core duplicons reported in the earlier work [27], including the cores from the *LCR16* cluster. Note that many previously identified core duplicons in the hg17 version of the human genome turned out to be short tandem repeats in the hg19 version. The whole decomposition process took less than a minute on the final set of ≈58,000 SDs.

[4] `http://humanparalogy.gs.washington.edu`

Table 3 SD coverage of the human and mouse genomes (hg19 and mm8) and the runtime performance of BISER, SEDEF and SDquest. "Missed" and "Extra" columns are calculated with respect to the WGAC SD calls. All running times are reported on 8 CPU cores. We could not run WGAC as we do not have access to the legacy hardware needed for its execution; the reported runtime is from [33].

hg19				
Tool	**Covered** (MBp)	**Missed** (MBp)	**Extra** (MBp)	**Time**
WGAC (gold standard)	78.2			days
BISER	112.4	0.4	34.6	10m
SEDEF	108.4	0.1	30.3	1h 15m
SDquest	95.2	2.5	19.5	3h 56m
mm8				
Tool	**Covered** (MBp)	**Missed** (MBp)	**Extra** (MBp)	**Time**
WGAC (gold standard)	60.6			days
BISER	94.4	0.8	34.6	13m
SEDEF	105.2	0.1	44.7	1h 24m
SDquest	82.5	2.7	24.5	6h 06m

3.3 Multi-genome results

In addition to running BISER on a single genome, we also ran BISER on the following seven related genomes:

- *C.jacchus* (marmoset, version calJac3),
- *M.mulatta* (macaque, version rheMac10),
- *G.gorilla* (gorilla, version gorGor6),
- *P.abelii* (orangutan, version ponAbe3),
- *P.troglodytes* (chimpanzee, version panTro6),
- *H.sapiens* (human, version hg19), and
- *M.musculus* (mouse, version mm8).

These genomes were analyzed in the previous work [10] with the sole exception of *M.musculus* that is novel to this analysis.

BISER took around four and a half hours to complete the run on 8 cores. Of that, it took around 2 hours to find putative SDs within the same species (42 minutes for in-species detection and 71 minutes for detecting SDs conserved across different species). The remaining time (2h 29m) was spent calculating the final alignments for all reported SDs (Table 2). The vast majority of alignment time (1h 37m minutes out of 2h 29m) was spent only on aligning putative SDs from calJac3 genome. We presume that this is due to the high presence of unmasked low-complexity regions in this particular assembly.

The SD decomposition and core duplicon detection took slightly less than 6 minutes to complete on a set of nearly 2,407,000 SDs. BISER found ≈116,000 elementary SD sets seeded by ≈13,000 cores. All cores from [27] and [10] were covered by BISER's cores as well.

4 Conclusion

More than a decade ago, the Genome 10K Project Consortium proposed to build genome assemblies for 10,000 species [18]. Due to the lack of high-quality long-read sequencing data, this aim was not immediately realized. However, the Genome 10K Project spearheaded the development of other large-scale many-genome sequencing projects such as the Earth BioGenome Project [29] and Vertebrate Genomes Project[5]. Recent developments in generating more accurate long-read sequencing data, coupled with better algorithms to assemble genomes now promise to make the aforementioned and similar projects feasible.

Analyzing the recently and soon-to-be generated genome assemblies to understand evolution requires the development of various algorithms for different purposes, from gene annotation [37] to orthology analysis [22] and the selection and recombination analysis [23]. Although a handful of tools such as SEDEF and SDquest are now available to characterize segmental duplications in genome assemblies, they cannot perform multi-species SD analysis, and they suffer from computational requirements. We developed BISER as a new segmental duplication characterization algorithm to be added to the arsenal of evolution analysis tools.

We demonstrate that (1) BISER is substantially faster than earlier tools; (2) it can characterize SDs in multiple genomes to delineate the evolutionary history of duplications; and (3) it can identify elementary SDs and core duplicons to help understand the mechanisms that give rise to SDs. We believe that BISER will be a powerful and common tool and will contribute to our understanding of SD evolution when thousands of genome assemblies become available in the next few years. The next step in this line of research would consist of interpreting BISER's results, biological analysis of the reported core duplicons, and applying BISER to a larger set of available mammalian genomes to infer the evolutionary history of ancient duplications.

References

1 Mohamed Ibrahim Abouelhoda and Enno Ohlebusch. Multiple genome alignment: Chaining algorithms revisited. In Ricardo Baeza-Yates, Edgar Chávez, and Maxime Crochemore, editors, *Combinatorial Pattern Matching*, pages 1–16. Springer Berlin Heidelberg, 2003.

2 S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, October 1990. doi:10.1016/S0022-2836(05)80360-2.

3 Shanika L. Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E. Ritchie, and Quentin Gouil. Opportunities and challenges in long-read sequencing data analysis. *Genome Biology*, 21:30, 2020. doi:10.1186/s13059-020-1935-5.

4 A. Andoni, R. Krauthgamer, and K. Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *Proc. IEEE 51st Annual Symp. Foundations of Computer Science*, pages 377–386, October 2010. doi:10.1109/FOCS.2010.43.

5 Francesca Antonacci, Jeffrey M Kidd, Tomas Marques-Bonet, Brian Teague, Mario Ventura, Santhosh Girirajan, Can Alkan, Catarina D Campbell, Laura Vives, Maika Malig, Jill A Rosenfeld, Blake C Ballif, Lisa G Shaffer, Tina A Graves, Richard K Wilson, David C Schwartz, and Evan E Eichler. A large and complex structural polymorphism at 16p12.1 underlies microdeletion disease risk. *Nat Genet*, 42(9):745–750, September 2010. doi:10.1038/ng.643.

6 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 51–58, New York, NY, USA, 2015. ACM. doi:10.1145/2746539.2746612.

7 J. A. Bailey, J. M. Kidd, and E. E. Eichler. Human copy number polymorphic genes. *Cytogenet Genome Res*, 123(1-4):234–243, 2008. doi:10.1159/000184713.

8 J. A. Bailey, A. M. Yavor, H. F. Massa, B. J. Trask, and E. E. Eichler. Segmental duplications: organization and impact within the current human genome project assembly. *Genome Res*, 11(6):1005–1017, June 2001. doi:10.1101/gr.187101.

[5] https://vertebrategenomesproject.org/

9 Jeffrey A Bailey and Evan E Eichler. Primate segmental duplications: crucibles of evolution, diversity and disease. *Nat Rev Genet*, 7(7):552–564, July 2006. doi:10.1038/nrg1895.

10 Stuart Cantsilieris, Susan M. Sunkin, Matthew E. Johnson, Fabio Anaclerio, John Huddleston, Carl Baker, Max L. Dougherty, Jason G. Underwood, Arvis Sulovari, PingHsun Hsieh, Yafei Mao, Claudia Rita Catacchio, Maika Malig, AnneMarie E. Welch, Melanie Sorensen, Katherine M. Munson, Weihong Jiang, Santhosh Girirajan, Mario Ventura, Bruce T. Lamb, Ronald A. Conlon, and Evan E. Eichler. An evolutionary driver of interspersed segmental duplications in primates. *Genome biology*, 21:202, 2020. doi:10.1186/s13059-020-02074-4.

11 Keegan Carruthers-Smith. Sliding window minimum implementations, 2013. last accessed 28 January 2021. URL: SlidingWindowMinimumImplementations.

12 Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

13 Jean-Félix Dallery, Nicolas Lapalu, Antonios Zampounis, Sandrine Pigné, Isabelle Luyten, Joëlle Amselem, Alexander H. J. Wittenberg, Shiguo Zhou, Marisa V. de Queiroz, Guillaume P. Robin, Annie Auger, Matthieu Hainaut, Bernard Henrissat, Ki-Tae Kim, Yong-Hwan Lee, Olivier Lespinet, David C. Schwartz, Michael R. Thon, and Richard J. O'Connell. Gapless genome assembly of colletotrichum higginsianum reveals chromosome structure and association of transposable elements with secondary metabolite gene clusters. *BMC genomics*, 18:667, 2017. doi:10.1186/s12864-017-4083-x.

14 Franklin Delehelle, Sylvain Cussat-Blanc, Jean-Marc Alliot, Hervé Luga, and Patricia Balaresque. ASGART: fast and parallel genome scale segmental duplications mapping. *Bioinformatics*, 34:2708–2714, 2018. doi:10.1093/bioinformatics/bty172.

15 Max L. Dougherty, Jason G. Underwood, Bradley J. Nelson, Elizabeth Tseng, Katherine M. Munson, Osnat Penn, Tomasz J. Nowakowski, Alex A. Pollen, and Evan E. Eichler. Transcriptional fates of human-specific segmental duplications in brain. *Genome research*, 28:1566–1576, 2018. doi:10.1101/gr.237610.118.

16 John W. Drake, Brian Charlesworth, Deborah Charlesworth, and James F. Crow. Rates of spontaneous mutation. *Genetics*, 148(4):1667–1686, 1998. arXiv:https://www.genetics.org/content/148/4/1667.full.pdf.

17 Huan Fan, Anthony R Ives, Yann Surget-Groba, and Charles H Cannon. An assembly and alignment-free method of phylogeny reconstruction from next-generation sequencing data. *BMC genomics*, 16:522, July 2015. doi:10.1186/s12864-015-1647-5.

18 Genome 10K Community of Scientists. Genome 10K: a proposal to obtain whole-genome sequence for 10,000 vertebrate species. *J Hered*, 100(6):659–674, 2009. doi:10.1093/jhered/esp086.

19 Santhosh Girirajan, Megan Y. Dennis, Carl Baker, Maika Malig, Bradley P. Coe, Catarina D. Campbell, Kenneth Mark, Tiffany H. Vu, Can Alkan, Ze Cheng, Leslie G. Biesecker, Raphael Bernier, and Evan E. Eichler. Refinement and discovery of new hotspots of copy-number variation associated with autism spectrum disorder. *Am J Hum Genet*, 92(2):221–237, February 2013. doi:10.1016/j.ajhg.2012.12.016.

20 Hiroyuki Hanada, Mineichi Kudo, and Atsuyoshi Nakamura. On practical accuracy of edit distance approximation algorithms. *arXiv preprint arXiv:1701.06134*, 2017. arXiv:1701.06134v1.

21 Robert S. Harris. *Improved Pairwise Alignment of Genomic Dna*. PhD thesis, Pennsylvania State University, University Park, PA, USA, 2007. AAI3299002.

22 Xiao Hu and Iddo Friedberg. SwiftOrtho: A fast, memory-efficient, multiple genome orthology classifier. *GigaScience*, 8, October 2019. doi:10.1093/gigascience/giz118.

23 Martin Hölzer and Manja Marz. PoSeiDon: a Nextflow pipeline for the detection of evolutionary recombination events and positive selection. *Bioinformatics*, July 2020. doi:10.1093/bioinformatics/btaa695.

24 Chirag Jain, Alexander Dilthey, Sergey Koren, Srinivas Aluru, and Adam M. Phillippy. A fast approximate algorithm for mapping long reads to large reference databases. In S. Cenk Sahinalp, editor, *Proceedings of 21st Annual International Conference on Research in Computational Molecular Biology (RECOMB 2017)*, volume 10229, pages 66–81, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-56970-3_5.

25 Chirag Jain, Sergey Koren, Alexander Dilthey, Adam M Phillippy, and Srinivas Aluru. A fast adaptive algorithm for computing whole-genome homology maps. *Bioinformatics*, 34(17):i748–i756, 2018.

26 Zhaoshi Jiang, Robert Hubley, Arian Smit, and Evan E. Eichler. Dupmasker: a tool for annotating primate segmental duplications. *Genome research*, 18:1362–1368, August 2008. doi:10.1101/gr.078477.108.

27 Zhaoshi Jiang, Haixu Tang, Mario Ventura, Maria Francesca Cardone, Tomas Marques-Bonet, Xinwei She, Pavel A Pevzner, and Evan E Eichler. Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nature genetics*, 39:1361–1368, November 2007. doi:10.1038/ng.2007.9.

28 Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

29 Harris A. Lewin, Gene E. Robinson, W. John Kress, William J. Baker, Jonathan Coddington, Keith A. Crandall, Richard Durbin, Scott V. Edwards, Félix Forest, M. Thomas P. Gilbert, Melissa M. Goldstein, Igor V. Grigoriev, Kevin J. Hackett, David Haussler, Erich D. Jarvis, Warren E. Johnson, Aristides Patrinos, Stephen Richards, Juan Carlos Castilla-Rubio, Marie-Anne van Sluys, Pamela S. Soltis, Xun Xu, Huanming Yang, and Guojie Zhang. Earth BioGenome Project: Sequencing life for the future of life. *Proceedings of the National Academy of Sciences of the United States of America*, 115:4325–4333, April 2018. doi:10.1073/pnas.1720115115.

30 Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics (Oxford, England)*, 34:3094–3100, September 2018. doi:10.1093/bioinformatics/bty191.

31 Tomas Marques-Bonet, Jeffrey M Kidd, Mario Ventura, Tina A Graves, Ze Cheng, LaDeana W Hillier, Zhaoshi Jiang, Carl Baker, Ray Malfavon-Borja, Lucinda A Fulton, Can Alkan, Gozde Aksay, Santhosh Girirajan, Priscillia Siswara, Lin Chen, Maria Francesca Cardone, Arcadi Navarro, Elaine R Mardis, Richard K Wilson, and Evan E Eichler. A burst of segmental duplications in the genome of the African great ape ancestor. *Nature*, 457(7231):877–881, February 2009. doi:10.1038/nature07744.

32 Guillaume Marçais, Arthur L. Delcher, Adam M. Phillippy, Rachel Coston, Steven L. Salzberg, and Aleksey Zimin. MUMmer4: A fast and versatile genome alignment system. *PLoS computational biology*, 14:e1005944, January 2018. doi:10.1371/journal.pcbi.1005944.

33 Ibrahim Numanagić, Alim S Gökkaya, Lillian Zhang, Bonnie Berger, Can Alkan, and Faraz Hach. Fast characterization of segmental duplications in genome assemblies. *Bioinformatics*, 34:i706–i714, September 2018. doi:10.1093/bioinformatics/bty586.

34 Lianrong Pu, Yu Lin, and Pavel A Pevzner. Detection and analysis of ancient segmental duplications in mammalian genomes. *Genome research*, 28:901–909, June 2018. doi:10.1101/gr.228718.117.

35 Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85. ACM, 2003.

36 Ariya Shajii, Ibrahim Numanagić, Riyadh Baghdadi, Bonnie Berger, and Saman Amarasinghe. Seq: A high-performance language for bioinformatics. *Proc. ACM Program. Lang.*, 3, October 2019. doi:10.1145/3360551.

37 Alaina Shumate and Steven L. Salzberg. Liftoff: accurate mapping of gene annotations. *Bioinformatics*, December 2020. doi:10.1093/bioinformatics/btaa1016.

38 Peter H Sudmant, Jacob O Kitzman, Francesca Antonacci, Can Alkan, Maika Malig, Anya Tsalenko, Nick Sampas, Laurakay Bruhn, Jay Shendure, 1000 Genomes Project, and Evan E Eichler. Diversity of human copy number variation and multicopy genes. *Science*, 330(6004):641–646, October 2010. doi:10.1126/science.1197005.

39 Hajime Suzuki and Masahiro Kasahara. Introducing difference recurrence relations for faster semi-global alignment of long sequences. *BMC bioinformatics*, 19(1):33–47, 2018.

40 O. Tange. GNU Parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47, February 2011. doi:10.5281/zenodo.16303.

41 Robert Endre Tarjan. A class of algorithms which require nonlinear time to maintain disjoint sets. *J. Comput. Syst. Sci.*, 18(2):110–127, 1979. doi:10.1016/0022-0000(79)90042-4.