Josip Juraj Strossmayer University of Osijek

University of Dubrovnik

Ruđer Bošković Institute

University Postgraduate Interdisciplinary Doctoral Study

Molecular Biosciences

Marin Volarić

Comprehensive analysis of satellite DNAs in the new genome assembly of the model organism Tribolium castaneum

PhD thesis

Zagreb, 2024

TEMELJNA DOKUMENTACIJSKA KARTICA

Sveučilište Josipa Jurja Strossmayera u Osijeku **Doktorski rad**
Sveučilište u Dubrovniku
Institut Ruđer Bošković
Doktorski studij Molekularne bioznanosti

Znanstveno područje: Interdisciplinarno područje znanosti
Znanstvena polja: Biologija

Sveobuhvatna analiza satelitnih DNA u novo posloženom genomu modelnog organizma *Tribolium castaneum*

Marin Volarić

Doktorski rad je izrađen u: Laboratoriju za nekodirajuće DNA Instituta Ruđer Bošković u Zagrebu

Mentor/i: dr. sc. Nevenka Meštrović

Kratki sažetak doktorskog rada:

Satelitne DNA su uzastopno ponavljajući genomski elementi ključni za organizaciju i regulaciju genoma, ali koji su još uvijek slabo istraženi. Korištenjem Nanopor sekvenciranja generiran je novi genomski sklop vrste *Tribolium castaneum*, koji je obogaćen satelitnim DNA. Sveobuhvatne analize genoma i transkriptoma otkrile su postojanje satelitskih DNA u regijama bogatim genima, koje pokazuju dinamičnu razmjenu između kromosoma, mehanizme samo-propagacije i diferencijalnu transkripciju tijekom embriogeneze i razvoja mozga.

Broj stranica: 119
Broj slika: 37
Broj tablica: 6
Broj literaturnih navoda: 163
Jezik izvornika: Engleski
Ključne riječi: *Tribolium castaneum*, genomski sklop, Nanopore sekvenciranje, satelitna DNA, male RNA

Datum javne obrane:

Povjerenstvo za javnu obranu:
1.
2.
3.
4. (zamjena)

Doktorski rad je pohranjen u: Nacionalnoj i sveučilišnoj knjižnici Zagreb, Ul. Hrvatske bratske zajednice 4, Zagreb;Gradskoj i sveučilišnoj knjižnici Osijek, Europska avenija 24, Osijek; Sveučilištu Josipa Jurja Strossmayera u Osijeku, Trg sv. Trojstva 3, Osijek

BASIC DOCUMENTATION CARD

Josip Juraj Strossmayer University of Osijek **PhD thesis**
University of Dubrovnik
Ruđer Bošković Institute
Doctoral Study of Molecular biosciences

Scientific Area: Interdisciplinary area of science
Scientific Fields: Biology

Comprehensive analysis of satellite DNAs in the new genome assembly of the model organism *Tribolium castaneum*

Marin Volarić

Thesis performed at: Laboratory for non-coding DNA at Ruđer Bošković Institute

Supervisor/s: Nevenka Meštrović, PhD

Short abstract:
Satellite DNAs are tandemly repeating genomic elements that are crucial for genome organization and regulation, but have been poorly understood. Nanopore sequencing was used to generate the new genome assembly of *Tribolium castaneum*, which is rich in satellite DNAs. The comprehensive genome and transcriptome analyzes revealed satellite DNAs in gene-rich regions, characterized by dynamic exchange between chromosomes, self-propagation mechanisms, and differential transcription during embryogenesis and brain development.

Number of pages: 119
Number of figures: 37
Number of tables: 6
Number of references: 163
Original in: English
Key words: Tribolium castanum, genome assembly, Nanopore sequencing, satellite DNA, small RNA

Date of the thesis defense:

Reviewers:
1.
2.
3.
4. (substitute)

Thesis deposited in: National and University Library in Zagreb, Ul. Hrvatske bratske zajednice 4, Zagreb; City and University Library of Osijek, Europska avenija 24, Osijek; Josip Juraj Strossmayer University of Osijek, Trg sv. Trojstva 3, Osijek

Contents

1. Introduction

1.1 Structure of eukaryotic genomes

1.1.1 Genome size paradox

The eukaryotic genome is organized in linear chromosomes, which are located in a membrane-bound organelle, the cell nucleus. Over the last 60 years, scientists have estimated the genome sizes of almost 20,000 eukaryotic genomes and have found an astonishing difference in genome size between different eukaryotic species. Apart from this unexpected difference in the genome size, it also became clear that there is no correlation between the genome size and organism complexity. In general, the size of eukaryotic genomes ranges from the modest 2.9 Mbp of the single-celled parasite *Encephalitozoon cuniculi* [1] to the massive 160 Gbp of the genome of the plant *Tmesipteris oblanceolata*, a span of over 61,000 [2]. This genome size paradox is especially prominent in plants, for example, model organism *Arabidopsis thaliana* has small genome with only ~135 Mb [3] while *Paris japonica* has of astonishing genome of 150 Gb in size [4]. The human genome (*Homo sapiens*) is more or less in the middle with 3Gb genome size[5]. However, most non-parasitic eukaryotic organisms have relatively consistent gene counts (ranging from 5,000 in *Saccharomyces pombe* [6] to 60,000 in *Trichomonas vaginalis* [7]), in multicellular organisms the discrepancy is even larger and the number of genes among the complex organisms changes only 2-3 times, ranging from 15000 to 35000 genes while the genome sizes can change 61000 fold. The axolotl (*Ambystoma mexicanum*), for example, has about the same number of genes as humans (~23,000), but its genome is 10 times larger [8] (32Gb, *H. sapiens* 3.2Gb). Therefore, it is clear that genome size does not correlate with the number of genes or with the biological complexity of an organism. This phenomenon, which is observed in all eukaryotic species, is known as the C-value enigma or genome size paradox [9]. The solution to this enigma/paradox lies in the structure of the eukaryotic genomes themselves.

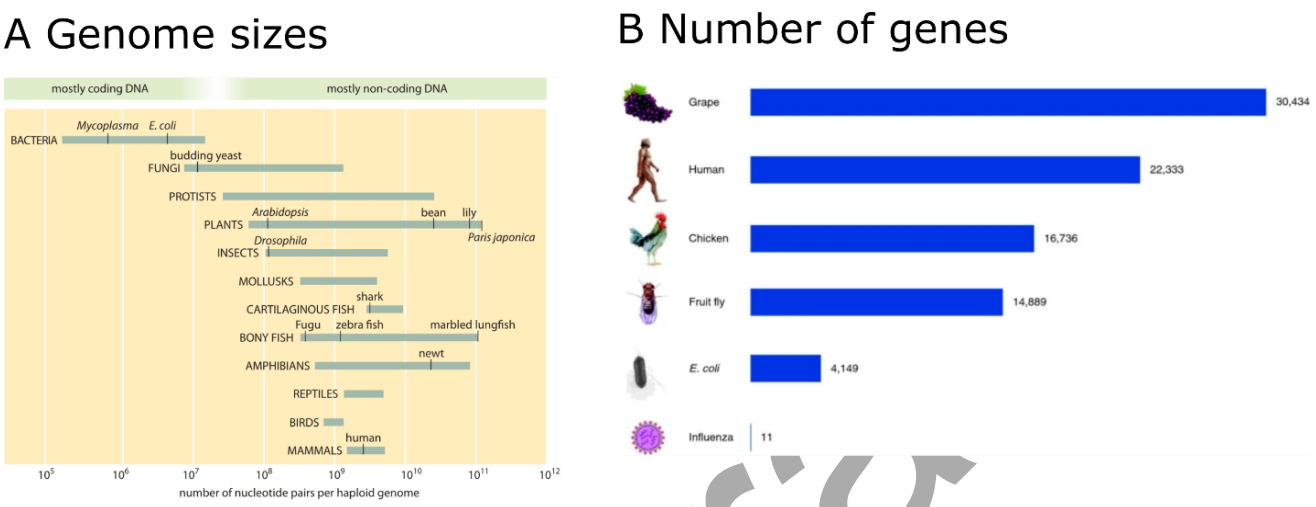


Figure 1.1 A Distribution of genome sizes across species (adapted from [10]) **B** Number of protein coding genes in selected species (adapted from [11])

1.1.2 Repetitive DNAs

Eukaryotic genomes can be broadly divided into two main components: coding and non-coding regions. The coding regions consist of genes that directly encode proteins, with the proportion of the genome occupied by these genes varying between species. In contrast, non-coding DNAs, with the exception of transposons that code for proteins for their own replication and transposition, do not code for proteins, and some of them are not even transcribed. For example, protein-coding genes make up 68% of the genome in *Saccharomyces cerevisiae*, but only about 2% in *H. sapiens*. However, the general trend in complex eukaryotic organisms is for the protein-coding regions to occupy only a small percentage of the genome, namely around 10% in animals and 8% in plants. The remaining genomic regions are non-coding and contain a variety of elements, including introns, non-coding RNAs, regulatory sequences, and repetitive DNA [12]. Repetitive DNA is further classified into two main categories: transposable elements (TEs) and tandem repeats.

Transposable elements are scattered, repetitive sequences that move through the genome by duplication and relocation. The two largest classes of TEs are autonomous and nonautonomous transposable elements. Autonomous TEs consist of DNA transposons and retrotransposons. DNA transposons encode a transposase enzyme, which is flanked by inverted terminal repeats. When expressed, the transposase recognizes these repeats, excises the transposon, and reinserts it at a new genomic location.

Retrotransposons are divided into LTR (long terminal repeats) and non-LTR retrotransposons [13]. The most studied non-LTR transposons in mammals are L1 LINE (Long interspersed elements) elements, which encode reverse transcriptase and endonuclease and are present in a large number of mammalian genomes [14]. LTR retrotransposons originate from ancient retroviral infections and encode proteins such as gag (structural proteins), pol (reverse transcriptase and integrase), pro (protease), and in some cases env (envelope proteins) [15]. Nonautonomous elements require the presence of other TEs because they lack the genes that are needed for their transposition. The prominent examples of nonautonomous elements are Short Interspersed Nuclear Elements (SINEs) and Miniature Inverted-repeat Transposable Elements (MITEs) [16]. SINEs are typically short (100-500 bp) sequences that are dispersed throughout the genome relying on the machinery of LINEs to transpose. The most abundant SINEs in human genome are *Alu* elements [17]. MITEs are very small TEs ranging from 100 to 600 bp characterized by their terminal inverted repeats (TIRs) and similar as SINEs, MITEs rely on other transposable elements for their mobility. Genomic abundance of transposable elements can vary greatly depending on lineage and even between closely related species. In *H. sapiens* TEs account for approximately 45% of the genome. Retrotransposons dominate, accounting for about 20%, SINEs for about 13% and LTR retrotransposons for 8% while DNA transposons represent a smaller portion, around 3% of the genome, but most are inactive remnants of past activity [18]. In some plants like *Triticum aestivum* (wheat) with a huge genome (about 17 Gb), TEs can consist approximately 80% of the total genome content. Retrotransposons, particularly LTR elements, are the most abundant TEs, playing a key role in genome expansion and shaping genetic diversity [19]. Additionally, in maize (*Zea mays*), TEs make up an even larger portion of the genome, contributing about 85% of the maize's relatively large genome (2.3 Gb). The content is comprised of retrotransposons, particularly LTR retrotransposons, which have proliferated massively. DNA transposons also make up a significant fraction, contributing to the dynamic nature of the maize genome [20]. In insects, for example *Drosophila melanogaster* TEs account for about 15-20% of the genome. The most abundant TEs are non-LTR retrotransposons, particularly the LINE-like elements. DNA transposons are also present, though in lower abundance than in plants or mammals [21]. The axolotl (*A. mexicanum*) has one of the largest known vertebrate genomes (~32 Gb), with more than 60% of its genome composed of TEs. LINE elements are highly abundant, and the high content of repetitive sequences contributes to the massive genome size of these amphibians [8].

The other most abundant class of repetitive elements are tandemly repeated sequences such as ribosomal DNAs, telomeric repeats, microsatellites, minisatelites and satellite DNA which serve various biological roles. Ribosomal DNAs are components of ribosomes which are essential for translation of proteins in the cell. Telomeric repeats are short sequences (5-10bp in length) located at chromosome ends playing a key role in chromosome stability by preventing end-to-end fusions and eliminating recurrent DNA loss at chromosome ends after numerous replication cycles [22].

Microsatellites, also known as simple sequence repeats (SSR) or short tandem repeats (STR), are dispersed repetitive DNAs characterized by repeat units that range from 2 to 10 base pairs (bp) in length. They are highly polymorphic and associated with population variation, contributing to up to 3% of the human genome. Due to their variability, microsatellites are frequently used in population genetics, forensics, and evolutionary studies, as they serve as markers for tracking genetic differences among individuals or populations [23]. Minisatellites have larger repeat units, typically ranging from 10 to 100 bp. They are less abundant in the genome compared to microsatellites and are often found in euchromatic regions, forming arrays that can range in size from 0.5 to 30 kb. These regions are also highly variable between individuals, leading to their classification as variable number tandem repeats (VNTRs). Because of their hypervariability, minisatellites are also used in DNA fingerprinting and forensic applications to differentiate between individuals. Additionally, minisatellites act as hotspots for homologous genetic recombination events, which can lead to genomic rearrangements [24].

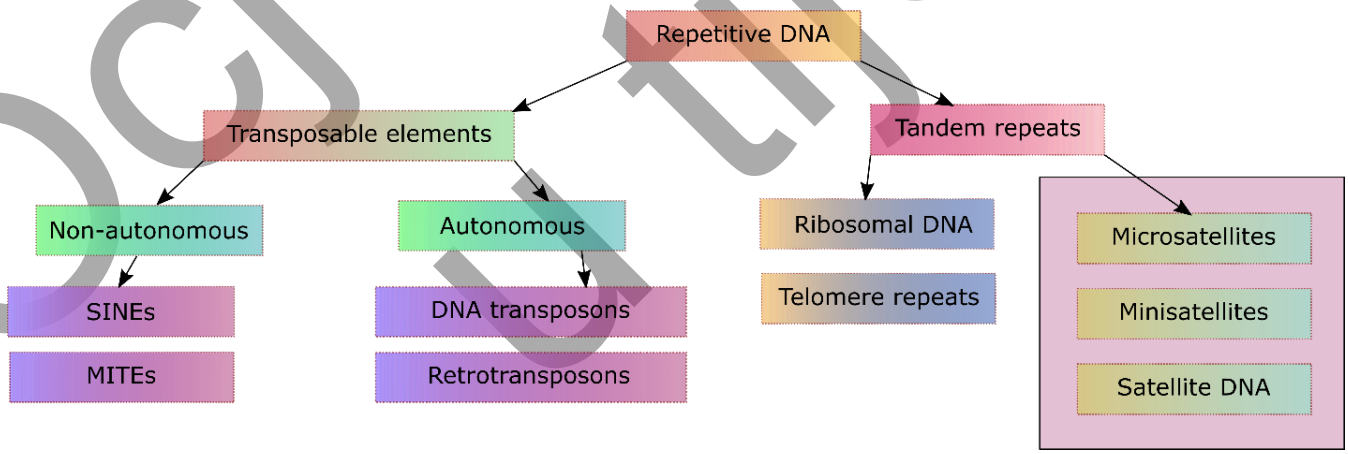


Figure 1.2 Hierarchical breakdown of different forms of repetitive DNA commonly found in genomes, with transposable elements having mobile capabilities and tandem repeats often forming structural components. Pink shade represents satellite DNA.

1.1.3 Satellite DNAs

Satellite DNA (satDNA) consists of much larger repeat units, typically starting at around 100 bp and extending up to several kilobases (kb) [25]. This basic unit of repetitive sequence is referred to as a monomer or repeat unit in the context of satDNA (Figure 1.3). Multiple monomers arranged in tandem form arrays that can range in length from a few hundred bases to megabases. Very long arrays are often found in the centromeric chromosome region. A satellitome refers to the complete set of satDNA families in an organism's genome, including all variations of these repetitive sequences that comprise different satDNA families and subfamilies. Satellitomes are species-specific, and their composition can vary greatly in terms of sequence length, abundance and organization. For example, some species may have a few dominant and highly abundant satDNA families, while other closely related species may have a greater diversity of satDNAs with moderate genome occupancy [26]. Although satDNAs have long been referred to as "junk" or selfish DNA [27] with no known biological function, decades of research have demonstrated that satDNAs are associated with many different cellular processes and structures, such as those in the peri-(centromere).

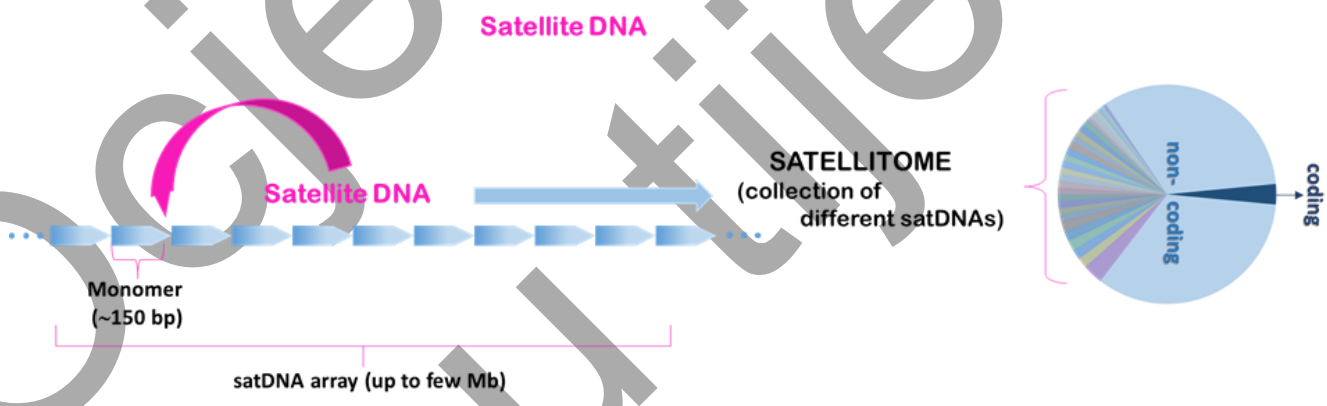


Figure 1.3 Satellite DNA organization. Monomer represent the repetitive unit of satDNAs and monomers are organized in satDNA arrays. Satellitome represents the collection of all satDNA families within a genome. Courtesy of Evelin Despot Slade.

1.1.4 (Peri)centromeric satellite DNA

Satellite DNA (satDNA) plays an important role in the organization of centromeres. In primates, centromeres are rich in α-satellite DNA, which can account for up to 10 % of the total repeat content in their genomes. In humans, α-Satellite DNA is characterized by a monomer length of ∼171 bp corresponding to the length of DNA wrapped around mono-nucleosome particles. Subsequently, subfamilies of α-satellites can form chromosome-specific α-satellite higher-order repeats (HORs) and differ by sequence variations, the arrangement of monomers and the overall size of the HOR [28]. This is best seen in the α-satellite regions of the human genome, where this single satDNA sequence accounts for almost half of all human satellite DNA[23]. Human α-satellite monomers exhibit up to 60% divergence and have been categorized into five suprachromosomal families (SFs), with SF1 and SF2 consisting of tandem arrangements of two alternating α-satellite units that can differ by up to 30%. These dimeric units have been identified as the most abundant functional α-satellites based on their binding to centromeric proteins in different populations [29] .

In contrast to the chromosome specific organizational pattern in human, in mouse (*Mus musculus*) (peri)centromeric satDNA families are nearly identical across all chromosomes. This indicates a high level of sequence homogenization and the absence of chromosome-specific variants. The centromere is composed primarily of 120-bp minor satellites (MiSats), which include the CENP-B box and comprise about 1–2% of the mouse genome which are flanked with TeLoCentric satellite arrays on the telomeric side and Major Satellite (MaSat) on the chromosomal side [30]. Other centromeric satellites in mice include the 150-bp MS3 and 300-bp MS4 satellites, which co-localize with MiSats at centromeres, though their precise roles in centromeric function remain to be fully understood [31].

D. melanogaster, on the other hand, has a distinctly different centromeric satDNA structure, with repeat units that are much shorter, typically 5-10 bp, compared to the longer nucleosomal repeats found in most other complex eukaryotes [23]. Early research assigned the centromere function of *D. melanogaster* to a 420-kb locus on the X-derived minichromosome Dp1187, which contains AAGAG and AATAT satellites interspersed with complex sequence islands. The centromeric satellite repeats of *D. melanogaster* consists mostly of short tandem arrays of 5- to 12-bp sequences, often following the RRNRN pattern, where R represents a purine and N any nucleotide [32].

1.1.5 Euchromatic satellite DNAs

To date, studies have mainly focused on satDNAs in (peri)centromeric heterochromatin. Although there is clear evidence that satDNAs have been assigned some roles, primarily in centromere structure, there is an almost complete lack of understanding of their organization, evolutionary dynamics and the molecular mechanisms that drive their spread across the genome in euchromatic regions. While research into the structure, organization and function of satDNA is traditionally focused on heterochromatic regions, euchromatic satDNA, which resides in more transcriptionally active regions, has recently also been identified and described in multiple species. Although less abundant than its heterochromatic counterpart [26] euchromatic DNAs plays significant roles in various genomic functions. In *Drosophila*, euchromatic satDNA contributes to regulating X chromosome dosage compensation in males, highlighting its role in sex chromosome regulation [33]. In *Aedes*, piRNA derived from a satDNA located in euchromatic regions has been found to control embryonic development, indicating a critical function this euchromatic satDNA in early developmental processes [34]. Furthermore, euchromatic α-satellite DNA repeats show increased levels of H3K9me3 upon heat stress, which may influence the expression of nearby genes by altering chromatin states[35] .Additionally, human TopI topoisomerase has been found interacting with the human α-satellite DNA, which has also been found dispersed in smaller clusters across chromosomal arms, pointing to the possibility of a role in DNA relaxation during replication [36]. These findings suggest that euchromatic satDNA could act as "evolutionary tuning knobs," influencing gene regulation and chromatin dynamics [37], [38]. This regulatory capacity underlines the adaptive significance of euchromatic satDNA beyond its structural role.

1.1.6 Roles of satDNA

Even early studies focused on elucidating the role of satDNA found evidence that it is essential for the maintenance of chromosome stability and proper segregation during cell division. Ando et al. (2002) elucidated the role of CENP-A loading and kinetochore assembly at the centromere, with satDNA serving as essential scaffold for proper chromosome segregation in *H. sapiens* HeLa cells [39]. A unique mechanism of transposon "cleaning" from the centromere of *A. thaliana*, as described by Wlodzimierz et.

al. [40], suggests that satDNA may not just self-propagate but also engage in self-maintenance processes akin to essential genes. The enrichment of the non-B form in satDNAs at centromeres of primates, disclosed dyad symmetries as key factors in centromere formation and function, highlighting the structural complexity of these regions [41]. In addition, conserved regions of α-satDNA have been discovered in primates and hominids that are thought to be necessary for the correct binding of the CENP-B protein, which is essential for the assembly of specific centromere structures in interphase nuclei. This conserved region is 17bp long and is referred to as the CENP-B box [42] . Similar motifs have also been confirmed in holocentric *Meloidogyne spp*. where conservation of both the 19bp box of centromeric satDNAs and the centromeric H3 protein, αCENH3, has been observed. Interestingly, there are five different satDNA families that share the same conserved 19bp motif [43]. Thus, it can be assumed that conserved motifs in satDNAs, such as the CENP-B box and the 19bp motif, can potentially serve as a functional signal of the centromere in the form of a protein binding site.

Intriguingly, new studies have also provided evidences for role of satDNA transcripts in the process of malignant transformation, thus indicating their impact in cancer progression [44]. Although satDNA transcription has become a focus of interest in the recent years regarding its pathophysiologic contribution, our knowledge concerning significance of satDNAs transcripts in normal physiological conditions is still rather limited. Multiple studies have found that centromeric satDNA is transcribed in large quantities across various species into non-coding RNAs (ncRNA), including human *D. melanogaster* [45], *Gryllus* crickets [46], and *Felis catus* [47]. In humans, α-satellite transcripts are essential for cell cycle progression, as their depletion disrupts centromeric protein A (CENP-A) loading, leading to cell cycle arrest. α-satellite ncRNAs also regulate spindle attachment and chromatid separation through AURORA B proteins and associate with SUV39H1, suggesting a role in heterochromatin maintenance [48] . Pericentromeric satDNA transcripts have also been linked to chromatin formation and the accumulation of HP1 proteins, while human SATIII ncRNAs are involved in stress responses, particularly under heat shock, forming nuclear stress bodies (nSBs) that influence RNA splicing and protect against cell death [49]. Similar stress-related functions of satDNA transcripts are found in *Drosophila*, where SatDNA III, located on multiple chromosomes, plays roles in heterochromatin formation, centromeric function, and gene regulation [50]. In *F. catus*, FA-SAT, a major satDNA sequence is transcribed across different species, where it interacts with PKM2, regulating cell proliferation and apoptosis. The transcriptional activity of

FA-SAT and its absence is associated to cell death, with potential implications for cancer, as aberrant satDNA ncRNA expression has been associated with cancer progression, aneuploidy, and hypomethylation of satDNA regions [51] . Moreover, in *Meloidogyne* species the active and coordinated transcription of some satDNAs across related genomes is suggested to be under cell-specific and developmental control, suggesting a functional role for satDNA transcription, possibly related to genome regulation or chromosomal architecture [52].

Finally, Bosco et al. [53] demonstrated that satDNA under-replication is linked to changes in genome size in various *Drosophila* species, suggesting that the differential replication of these sequences may contribute to genomic plasticity and size. Evidence of the role of satDNA in species separation and evolution is found in the layers of pericentromeric satDNA clusters, which can vary significantly between species, acting as genomic signatures that reflect divergence and speciation events.

1.1.7 Evolution and propagation of satDNAs

The evolution of satDNA elements is shaped by complex processes involving the amplification and diversification of repetitive sequences. Two models have been proposed to explain the evolution of satDNA: concerted evolution and the library hypothesis. Concerted evolution refers to the process by which repetitive DNA sequences within a species evolve as if in coordination, resulting in greater sequence similarity among individuals of the same species than between species. Rather than accumulating mutations in a single monomer as would be the case under canonical evolutionary models, mutations either spread across the repetitive units of satellite DNA or are eliminated. This evolutionary pattern is driven by a two-level process known as molecular drive, which involves both the homogenization and fixation of mutations, as well as the rate at which these mutations either spread or are removed. In reproductively isolated organisms, this process leads to rapid homogenization of satDNA within the genome of a species, causing the repeats to become more similar within a population than between two reproductively separated groups [54].

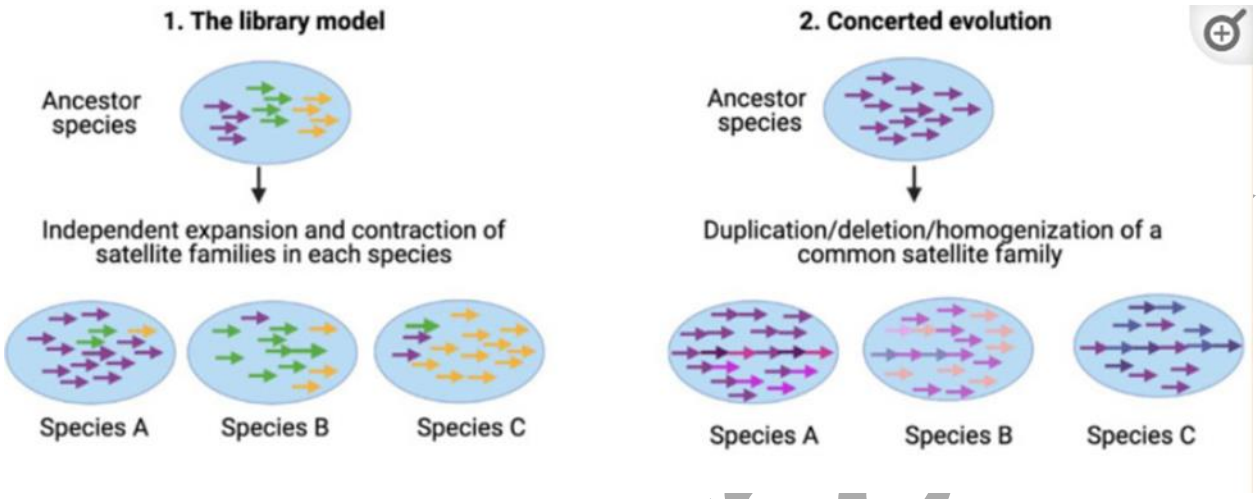


Figure 1.4 Two models of satDNA evolution, the library model as hypothesized [55] and concerted evolution of satDNA monomers (adapted from [23])

In *Drosophila* species, Kuhn et al. [56] explored this process at chromosomal and array levels using the 1.688 tandem repeats of *D. melanogaster*. They found that large arrays are present in the heterochromatin of chromosomes 2, 3, and X, with shorter arrays in euchromatin, demonstrating differential homogenization of 1.688 repeats in different genomic regions. Broad population studies of *D. melanogaster* have revealed that even selected populations can rapidly develop population-specific satDNA variants, evidenced by distinct k-mer spectra, supporting the idea that satDNA represents some of the fastest-evolving parts of the genome [57]. According to the library hypothesis of satellite DNA evolution, closely related species inherit a set of conserved satellite DNA families from a common ancestor, with each species differentially amplifying these families over time. When a specific satDNA family is amplified in one species, it remains as a low-copy variant in sister species. Differences in dominant satDNA sequences among closely related species are often attributed to rapid, gradual evolution within separate lineages. This process involves not only sequence changes but also constant alterations in the copy number of satellite DNA through expansions and contractions of satDNA arrays. According to the library concept of satellite DNA evolution, species-specific satellite profiles emerge from differential amplifications or contractions of a shared pool of sequences across related genomes. This "library" acts as a persistent source of sequences, allowing each species to independently expand certain sequences into dominant, high-copy satellites. As multiple satellite DNAs typically exist within a genome, fluctuations in their copy numbers can swiftly and significantly alter the overall genomic satellite profile.

This satDNA interspecies conservation without species-specific mutations was first observed in four *Palorus* congeneric species, which have been separated by up to 60 million years [55]. Each species contains a single AT-rich pericentromeric satDNA on all chromosomes, comprising 20-40% of their genomes, with the sequences showing high conservation in terms of sequence, repeat length, and organization. In each *Palorus* species, one of the four satellite families is amplified while the others are present as low-copy-number repeats, making up about 0.05% of the genome. These low-copy satellites are interspersed within the large arrays of the major satellite throughout the heterochromatic blocks. The library model has been also confirmed in plants and nematodes [52], [58].

Given the fact that satDNA exhibits very complex evolution at the genomic level, such as the diversity of satDNA profiles, dynamic processes of tandem duplications, contractions and sequence homogenization, high-quality and highly continuous telomere-to-telomere genome assemblies are essential for the proper understanding of the underlying evolutionary models of satDNAs.

1.2 Genome assembly

1.2.1 History of sequencing

Since the discovery of DNA and its role in inheritance, numerous methods have been developed to extract and convert genomic sequences into digital data. The first-generation DNA sequencing, based on Sanger method, works by conducting four separate polymerization reactions using tritium-labeled primers and chain-terminating 2,3-dideoxynucleoside triphosphates (ddNTPs), which terminate DNA strand elongation at specific points, generating DNA fragments of varying lengths. In 1977, this method led to the sequencing of the first genome, the 5,368 bp phage ϕX174 genome [59]. Significant improvements to Sanger sequencing, such as capillary sequencing and automated gel reading, allowed for rapid growth in the number of sequenced genomes. By the late 1980s, the NCBI database had over 40 million sequenced bases [60]. The Human Genome Project was launched in 1990 with the aim of sequencing the human genome by 2005, but was completed ahead of schedule in 2003 at a cost of around 2.7 billion dollars. This project spurred innovation in sequencing technologies and assembly algorithms, particularly the whole-genome shotgun strategy, which uses restriction enzymes to break genomes into millions of smaller

pieces, avoiding the time-consuming cloning step commonly used and allowing parallel sequencing of multiple short fragments at once [61]. This innovation in turn enabled the development of NGS (Next-Generation Sequencing), which revolutionized genomics through massively parallel sequencing and enabled the simultaneous sequencing of millions of DNA fragments. This advance represented a significant step forward as sequencing became faster and more affordable, which in turn transformed large-scale genomic studies. TGS (Third-Generation Sequencing) took this further, offering longer read lengths and real-time sequencing, enabling even more accurate assembly of challenging, repetitive regions. These technologies drastically reduced the time and cost of sequencing, accelerating the study of complex genomes and making high-quality genome assemblies more accessible.

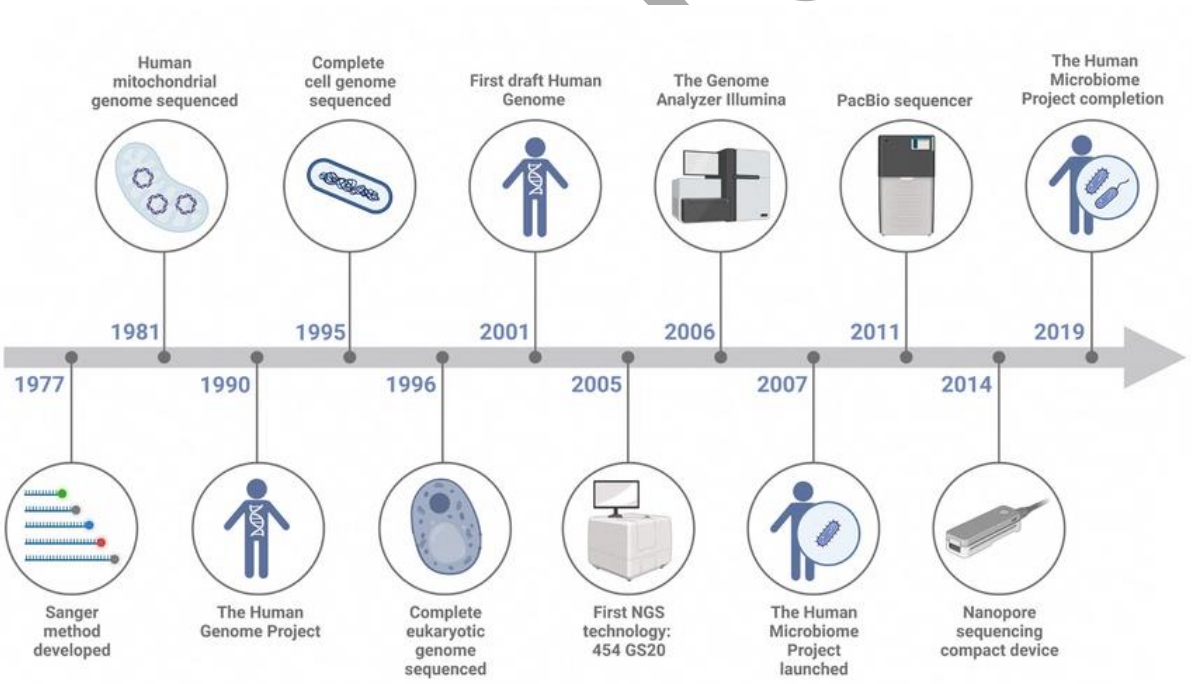


Figure 1.5 History of technology development and genome sequencing milestones. Advent of TGS technologies starts right at the beginning of 2010s. Adapted from [62]

1.2.2 Genome assembly approaches

Genome assembly is the computational process of decoding the sequence composition of the DNA in the cell of an organism, using numerous sequences, sequenced from different parts of the target DNA as input. The key output of sequencing experiments is "reads," or short DNA fragments. Assembly algorithms overlap these reads to form longer sequences called "contigs." These contigs are then arranged into scaffolds using additional data, such as linkage maps and jumping libraries, which are further organized to construct complete chromosomes (Figure 1.6).

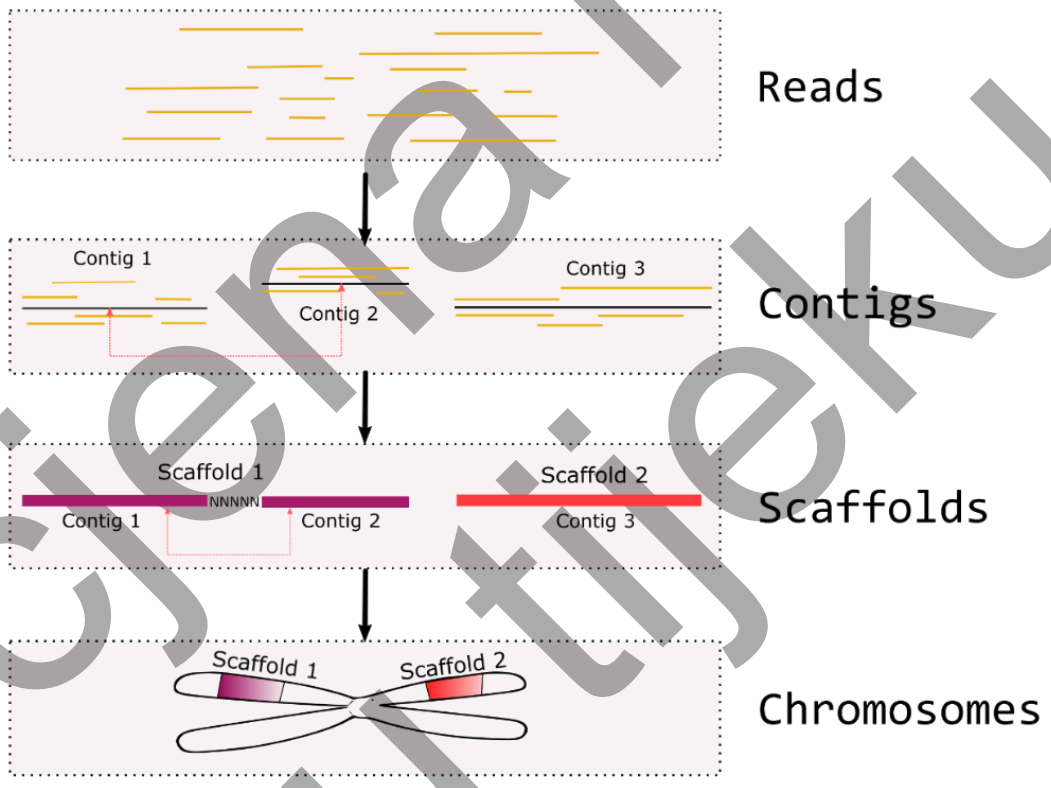


Figure 1.6 Schematic representation of the assembly process and key products of the assembly. Reads are assembled into contigs, which are linked into scaffolds spanned by sequence gaps (Ns). Finally, scaffolds are linked into full chromosomes.

Two main algorithmic approaches for assembling reads into contigs developed in the early phases of genome sequencing and are still widely used today are Overlap-Layout-Consensus (OLC) and De Bruijn

Graph (DBG). Both algorithms aim to reconstruct the genome from short sequence reads, but they do so in distinct ways. The OLC algorithm begins by identifying overlaps between all pairs of reads in an "all-vs-all" manner, often using dynamic programming techniques like the Needleman-Wunsch algorithm to find the best possible alignments between reads. This phase produces an overlap graph, where nodes represent the reads and edges represent the overlaps. Since the initial overlap graph can contain many redundant or conflicting overlaps, the next step, called the layout phase, simplifies the graph by removing unnecessary information, reducing it to the smallest and most accurate possible form. In the final consensus phase, the assembler breaks any unresolvable parts of the graph—regions where no read can bridge a gap—into separate contigs. The consensus sequence is generated from all reads mapped to each contig, producing an optimal representation of the genome segments [63].

The DBG assembly takes a different approach. Instead of directly comparing entire reads for overlaps, it breaks each read into shorter subsequences called k-mers, where "k" is a fixed length. These k-mers are used to build a directed graph, where nodes represent k-mers and edges indicate their adjacencies in the reads. The graph is traversed using a Eulerian walk, which ensures that each edge (or connection between k-mers) is visited exactly once. Unlike the OLC approach, DBG focuses on the relationships between these smaller subsequences rather than whole reads. However, like OLC, DBG assemblies are also challenged by genomic repeats, which can lead to breaks in the graph and the formation of separate contigs when a repeat region cannot be resolved [64].

Both methods have strengths and weaknesses. OLC is more suited to longer reads, as it relies on finding overlaps between entire reads, making it computationally intensive for large datasets. On the other hand, DBG is faster and more efficient for assembling a larger number of shorter reads, as it works on fixed-length k-mers rather than aligning full reads. However, DBG assemblies can struggle with high repeat content which “tangle” the graphs and may require more sophisticated methods to handle genome complexity. Both approaches are constantly evolving, and many modern assemblers use hybrid methods and various optimizations to take advantage of both algorithms.

1.2.3 2nd generation sequencing based assembly approaches

In the mid-2000s, the advent of next-generation sequencing (NGS) technologies, led by Illumina, revolutionized genomics by introducing a novel method called Sequencing by Synthesis (SBS). Unlike traditional sequencing methods, SBS enabled much faster and more cost-effective sequencing through several key innovations. The process begins with DNA fragmentation, followed by size selection of the fragments, favoring fragments close to the output capabilities of the machine (generally 50-300bp). Special adapters are then ligated to the ends of these fragments, a step that prepares them for sequencing. Before sequencing, the fragments are typically amplified through polymerase chain reaction (PCR) to increase the library size, ensuring that even small amounts of starting material can produce sufficient sequencing data. Once prepared, the amplified library is loaded onto a flow cell—a surface containing embedded sequences complementary to the adapters. The sequencing itself occurs in a massive parallel sequencing reaction, where thousands of DNA fragments are sequenced simultaneously. The flow cell's sequencing process involves stepwise, polymerase-driven incorporation of fluorescently labeled nucleotides. Each nucleotide added emits a unique fluorescent signal, which is detected by an optical reader, allowing the sequence to be determined in real-time as bases are incorporated. The development of SBS and other NGS technologies led to a dramatic reduction in sequencing costs (Figure 1.7a). Companies competed to offer the most efficient sequencing services, which further accelerated the spread of this technology. Together with the increase in computing power, this led to the democratization of genome sequencing and enabled the assembly of numerous draft genomes of eukaryotic and prokaryotic organisms. This explosive growth in sequencing capacity is perhaps best illustrated by the rapid increase in the number of publicly available genome sequences and the simultaneous decline in the cost of sequencing (Figure 1.7b). For example, the cost of sequencing a human genome has dropped from millions of dollars in the early 2000s to less than $1,000 in the 2010s [65].

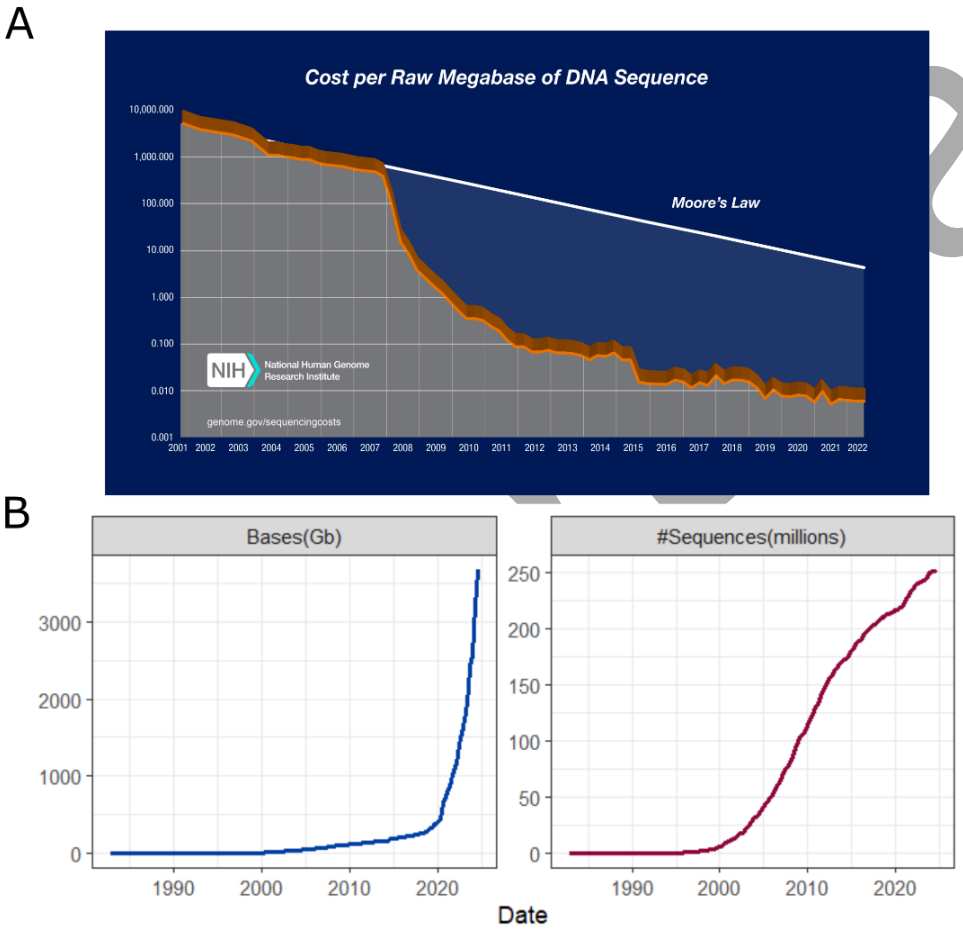


Figure 1.7 A Exponential reduction in cost of sequencing per Megabase of sequencing as calculated by NCBI from [66] **B** Rapid increase in the number of bases stored in public databases (left) and number of sequences (right), data adapted from [60]

By 2015, these advanced in sequencing technologies and subsequent assembly allowed for complete genome assemblies of multiple model organisms such as the first genome assembly of *D. melanogaster* published in 2000, with a genome size of approximately 180 million base pairs (Mb) [67]. The *Caenorhabditis elegans* genome, sized at about 100 Mbp, was sequenced in 1998 [68]. *A. thaliana* had its genome, which is around 125 Mb, published in 2000 [69]. For *M. musculus* the 2.7 Gbp sized was published in 200 2[70]. Lastly, the first genome assembly for *Z. mays*, with a size of approximately 2.3 Gbp, was published in 2009 [71].

Limitations of 2nd generation based assembly approaches

Despite the numerous advancements in NGS sequencing technologies, there remained significant limitations in their ability to resolve and assemble complete complex genomes. One of the main challenges is accurately assembling the non-coding parts of the genome, particularly repetitive regions. These repetitive sequences are difficult and more often straight impossible to resolve with short-read sequencing methods like those employed by Illumina [72]. The short read lengths, typically ranging from 100 to 300 base pairs, make it challenging to span long repetitive regions fully resulting in mis-assembly, collapse, or complete omittance. This can lead to gaps in the assembly or incorrect representations of the genome structure, which is vital for studies focusing on biological or evolutionary role of these sequences. In addition to these technical challenges, the computational resources required to assemble genomes with a high degree of repetitive content using NGS short read data can be substantial to impossible. The algorithms used to assemble short reads depend on the construction of complete overlap graphs and a single viable path through the graph. As the repetitiveness of the genome increases, often much higher coverage is needed to properly resolve the graph with exponential growth in time-complexity for solving such complex graphs.

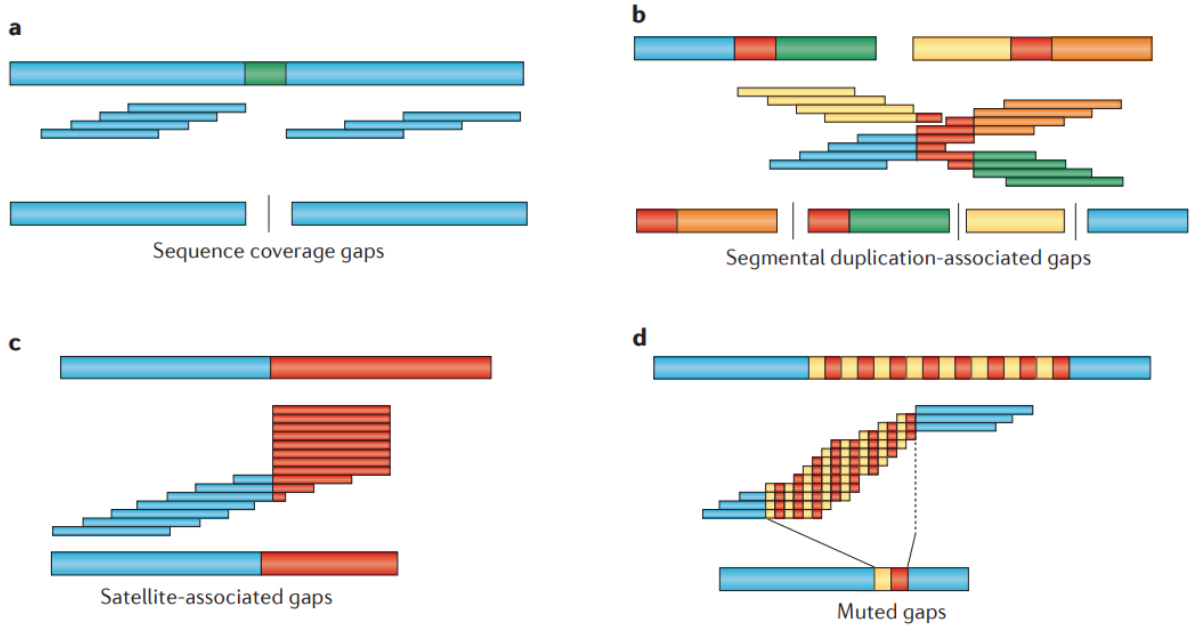


Figure 1.8 Limitations in efficient genome assemblies. All apart of A are problems stemming from various repetitive regions in the genome, preventing complete telomere to telomere assembly when using reads of insufficient length, while A can be solved by higher coverage and better sequencing. Adapted from [73]

Due to the added complexity of resolving repetitive regions of the genome, complete assembly of even the most important model organisms has long been challenging. For example, until recently [5], the human reference genome assembly, known as GRCh38.p14, contained 999 gaps [74]. The importance of continuous, gap-free genome assemblies, often referred to as telomere-to-telomere (T2T) assemblies, cannot be overstated, as these previously missing repetitive elements have been shown to contribute to the evolution of genomes by facilitating chromosomal rearrangements, gene duplications and the regulation of gene expression [47]. However, their proper inclusion in genome assemblies depended on the development of novel methods for genome sequencing that enabled much longer read lengths, and this is the main driver for the development of third-generation sequencing technologies.

1.2.4 3rd generation sequencing based assembly approaches

Several paradigm shifts in genome sequencing technologies have enabled the leap to third-generation sequencing (TGS). The first is the limitation of sequencing space from the DNA adapter plate to the single molecule level combined with much higher polymerase accuracy and more powerful optical devices that can capture the light signal at the molecular level. The second major change has been the move away from SBS-based methods and the development of different protein chemistries such as nanopores that enable the sequencing of native, minimally processed DNA.

The first commercial single-molecule sequencing technology was introduced by Helicos Biosciences in 2008. This breakthrough technology enabled the direct sequencing of single DNA molecules without amplification by attaching DNA molecules to coated glass surfaces and then applying conventional SBS techniques, skipping the preparative amplification step. However, this approach was characterized by short read lengths (30-35bp), high costs and the need for a lot of starting material, which limited its accessibility and widespread use. The first true single-molecule sequencing technology, known as Single-Molecule Real-Time (SMRT) sequencing, was introduced by Pacific Biosciences (PacBio) in 2011. This

technology represented a significant advance in genome sequencing as it allowed direct observation of DNA synthesis in real time at much longer lengths [75].

The concept of using nanopores for sequencing DNA or RNA molecules dates back to the late 1980s. Researchers envisioned using nanopores embedded in a membrane to read single-stranded (ss) nucleic acids as they pass through the pore. Despite the promising concept, technical problems delayed its practical implementation. It was not until 2012 that the first successful sequencing results using nanopore technology were reported. This breakthrough proved the feasibility of nanopore sequencing and paved the way for the development of commercial nanopore sequencing platforms, such as those from Oxford Nanopore Technologies, which have since revolutionized the field with their ability to sequence long reads and process a wide variety of sample types [76].

PacBio sequencing

One of the most significant advancements in sequencing-by-synthesis (SBS) technologies was the development of zero-mode waveguides (ZMWs) by PacBio. These nanometer-scale wells, combined with advanced material engineering, enable real-time observation of nucleotide incorporation by a single DNA polymerase molecule in an individual well. This innovation represents a major leap forward from traditional SBS methods, which relied on adapter-based plating and were limited by the inherent inaccuracies associated with the adapter plates, mainly the quality drop-off after a certain read length limit. Initially, PacBio SMRT technology faced challenges with short read lengths (around 1.5 kb) and high error rates (about 11%). Over time, PacBio made several improvements, such as the introduction of hairpin adapters, better polymerases, and labeling the 5’ phosphate of dNTPs, which is subsequently released instead of incorporating the nucleotide base into the growing nucleotide chain as used in NGS. These advances have enabled the development of two main types of PacBio sequencing: High Fidelity (HiFi) and Continuous Long Reads (CLR) [75]. HiFi sequencing provides highly accurate reads (approximately 99.9%) by generating multiple runs of the same DNA molecule, resulting in long reads of typically 10 to 20 kb. In contrast, CLR sequencing focuses on generating extremely long reads with a higher error rate (5-15%), which is useful for applications such as de novo genome assembly and structural variant detection [77]. These sequencing technologies have several advantages, such as very high read

length compared to NGS and, in the case of HiFi, extremely high accuracy, high yield in single sequencing experiments and, because they work with native DNA, they can be used to find indels and other structural variations when used in studies with multiple genomes. On the other hand, the main disadvantages of PacBio sequencing are the relatively high prices and start-up costs for sequencing machines and flow cells. In addition, they are limited to DNA input, and modified base detection is also limited due to the complexity of modified base detection based on polymerase sequencing alone, and although efforts have been made in recent years, error rates in modified base detection are still high [78] .

Oxford nanopore sequencing

Development of nanopore-based sequencing technologies, led by Oxford Nanopore Technologies (ONT) marked a significant departure from traditional sequencing methods. Oxford Nanopore sequencing directly reads the change in current on a membrane caused by passing of DNA or RNA molecules through nanoscale protein pores. In this method, the changes in ionic current are monitored as the nucleic acid moves through the nanopore, with each nucleotide causing a distinct disruption in the current. A single flow cell can have up to 48000 nanopores in 24000 wells (PromethION) allowing for real-time, single-molecule sequencing [79] . The biggest obstacle in obtaining usable sequencing data from flow cells is the post processing or the basecalling step. Unlike other sequencing technologies that depend only on the sensitivity of the light array, and are therefore computationally simple, obtaining accurate nucleotide data from slight current changes in nanopore sequencing is much more computationally intensive. For these purposes, specialized machine learning algorithms, called neural networks, must be trained and developed on known sequences before any sequencing data is generated [80]. With these advanced models, the basecalling error, which was 5-10% 5 years ago, has dropped to <1%, making nanopore sequencing a promising method for de novo genome assemblies at the telomere-to-telomere level. The key feature of ONT sequencing that enables such high-level assemblies is its ability to produce exceptionally long reads, often exceeding 100 kilobases, with the longest ever reported being 4.2Mb [81], allowing for complete coverage of centromeric regions with one or more reads, as was the case with the recent T2T assembly of *H. sapiens* chromosome X [82]. In addition, ONT sequencing offers

flexibility/portability and high throughput, with devices such as the compact MinION or the high-throughput PromethION. Since only native DNA passes through the pore and this DNA can be modified in many different organisms and acts as an epigenetic regulator of gene activity, researchers have developed various algorithms to recognize these modified bases. Currently 5mC, 5hmC, 6mA [83] are officially supported by the ONT and many other modifications are currently being developed by both Oxford Nanopore and independent researchers.

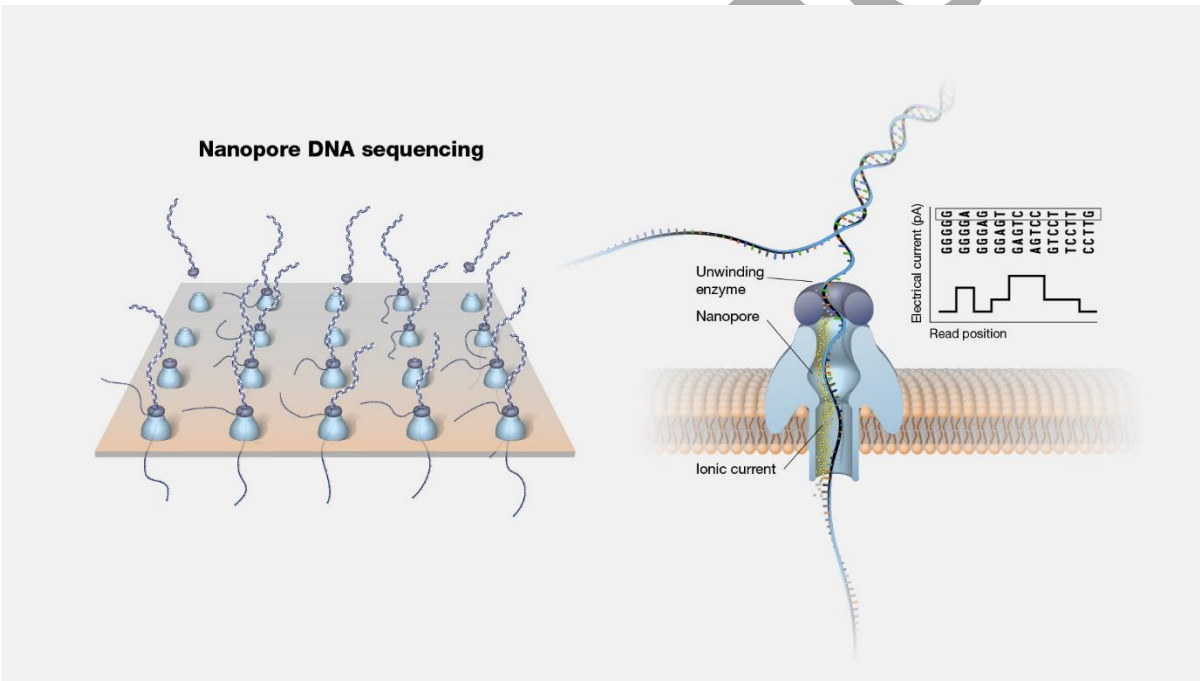


Figure 1.9 Schematic representation of Oxford Nanopore sequencing and subsequent basecalling, adapted from [84].

1.2.5 Modern assemblers

Traditional genome assembly approaches using short read sequences such as those generated by Illumina have long been the foundation for genome assembly. Assemblers such as ALLPATHS-LG, Velvet, and SOAPdenovo were developed specifically for Illumina short, relatively error-free reads [85]. However, with the development of third generation sequencing technologies that generate longer and but up to 100x more error prone sequencing reads, conventional assemblers and their strategies are no longer effective. This is mainly due to the fact that conventional assemblers rely on short read lengths to perform efficient

data processing steps such as indexing and hashing. Furthermore, as read length increases, the number of potential overlaps grows exponentially, making it difficult to identify the best overlaps in long, noisy reads [72].

New genome assemblers like Canu, HiCanu, *hifiasm* and Flye [86]–[89] have been developed to overcome the limitations of traditional assembly methods when dealing with long-read sequencing data. There are several main approaches these assemblers use to address the problems associated with integrating TGS data. First, assemblers like *hifiasm* and Canu implement sophisticated error correction algorithms prior to assembly to reduce the noise and the number of redundant overlaps. Next, OLC based assemblers such as Canu, Raven and Flye implement some variation of hierarchical overlapping, by iteratively filtering alignments which are less probable, thus reducing the size of the final overlap graph and managing computational demands more efficiently. Additionally, they employ memory-efficient data structures and algorithms, such as minimizers (Canu), FM-index (Flye), or in case of Redbean [90], utilizing a "fuzzy Bruijn graph" of larger k-mers (up to 256), drastically reducing the number of potential overlaps from the reads, making it less memory-intensive and better suited for long-read data. *Hifiasm* [86] contains dynamic data structures such as Bloom filters for faster retrieval of subsequences from generated graphs and iterative simplifications of the graph that adapt to the sequence data and prune the graph at each step. This enables efficient processing of PacBio HiFi sequences by saving and processing only the most important parts of the graph. These assemblers also feature modular and parallelizable pipelines that enable effective scaling across large data sets and high-performance computing resources. Furthermore, some of these assemblers such as *hifiasm*, are flexible enough to operate in hybrid modes, i.e., combining long and short reads to leverage the strengths of both, enabling the accurate assembly of complex genomes despite the challenges posed by newer sequencing technologies. However, generating a high-quality T2T genome assembly from only one sequencing technology is not feasible, thus, successfully producing a high-quality genome assembly requires more complex approaches based on multiple sequencing technologies.

1.2.6 Hybrid assembly approaches

These approaches, also called hybrid assembly approaches or hybrid assembly pipelines are based on integrating a whole plethora of sequencing technologies and algorithms in order to produce the best assembly possible. Thus, recent versions of *hifiasm* now incorporate ultra-long nanopore reads, which enhance the ability to construct more complete and contiguous genomes by using the longest reads to span unresolvable gaps. Additionally, scaffolding techniques, such as Hi-C contact maps and BioNano optical mapping, play a crucial role in refining and organizing high-quality contigs by providing structural information that improves the overall assembly and several algorithms such as YaHS and AllHiC [91], [92] leverage this in order to create chromosome level scaffolds from contigs. However, the best example of such holistic integration of multiple sequencing approaches is verkko [93], which represents a standardized pipeline used in human T2T assembly by combining ONT, PacBio HiFi and Hi-C data from the beginning of the assembly process iteratively building the final assembly.

As an example of the integration of these technologies, several large sequencing projects have been launched in recent years. These include several notable initiatives. The Earth BioGenome Project (EBP) is a large project to sequence the genomes of all eukaryotic species on Earth using TGS technologies to capture the complex and repetitive regions that are often missed by short-read sequencing [94]. Another important project is the Genome 10K Project, which aims to sequence 10,000 vertebrate genomes [95]. This project uses TGS methods to increase the resolution of genome assemblies and improve the scientific understanding of vertebrate evolution. The Darwin Tree of Life project [96], which focuses on sequencing the genomes of all eukaryotic species in the UK and Ireland, is also using TGS technologies to produce high quality, contiguous genome assemblies.

In addition, hybrid assembly approaches often use existing Illumina-based chromosome assemblies as a scaffold on which new, high-quality contigs are assembled and aligned using third-generation sequencing data and advanced algorithms such as RagTag, TGS-GapCloser and Liftoff [97]–[99]

1.2.7 Bioinformatical analyses of satDNAs

The study of satellite DNA (satDNA) evolution and organization in genomes has advanced significantly through the development of specialized algorithms and sequencing methodologies. The basics of these algorithms involve *de novo* detection and analysis of repetitive sequences within the genome, allowing

researchers to identify, classify, and study satDNA elements with greater precision. Commonly used algorithms for analyses of satDNAs sequenced data are; Tandem Repeats Finder (TRF) [100], which detects and analyzes tandem repeats based on nucleotide sequence alignment; ULTRA [101], which extends detection capabilities to more complex repeat structures with higher sensitivity; and TRASH [102], which combines alignment-free approaches with machine learning to classify repetitive DNA sequences from large genomic datasets. One of the most important algorithms for *de novo* satDNA detection is TAREAN [103]. This algorithm employs a graph-based sequence clustering approach using raw Next-Generation Sequencing (NGS) reads. TAREAN works by iteratively clustering the reads and constructing directed graphs, followed by de Bruijn graph construction from all k-mers present in potential satDNA candidates. Finally, it generates a consensus sequence representing the identified satDNA. In recent years, the combination of TAREAN and low-cost NGS methods has led to the discovery of numerous complete satellitomes across a wide range of eukaryotic species[104]–[107].

To gauge the evolutionary background of satDNA, researchers employ a range of bioinformatics and comparative genomics methods. One widely used approach is the construction of evolutionary trees or phylogenetic analyses (as seen in [108]), which map the relationships between satDNA sequences across different species. These analyses help to identify conserved satDNA families, track their amplification or contraction, and uncover the evolutionary pressures that shape their distribution. However, these methods also have limitations, particularly when it comes to resolving recent evolutionary events or dealing with the rapid diversification and turnover of satDNA sequences, which can lead to a loss of phylogenetic relationship. Additionally, the repetitive and highly mutable nature of satDNA makes it challenging to accurately reconstruct their evolutionary history, as homologous relationships can be obscured by sequence divergence and structural rearrangements.

Advances in sequencing and genome assembly have significantly pushed the boundaries for high-quality telomere-to-telomere assembly and comprehensive satDNA detection. Notable achievements include the first complete assembly of the human X chromosome in 2020 [82], followed by the complete assembly of the human genome in 2022 [5]. These milestones were complemented by the first complete decoding of *A. thaliana* in 2021 [3] and of *Z. mays* in 2023 [109]. Even more complex challenges, such as the attempt to resolve holocentromeric root-knot nematodes (*Meloidogyne spp.*) in 2024, come close to chromosome resolution of very complex assemblies with many different and abundant satDNAs [110]. These successes

have been made possible by the use of advanced sequencing technologies such as long-read sequencing, optical mapping and improved bioinformatics tools, which have enabled researchers to assemble genomes with remarkable accuracy and continuity.

The advances provided by these assemblies have allowed new insight into centromeric satDNAs, previously thought to be too complex due to repetitiveness to analyze completely. In the largest study of the human centromeres to date, [111], it was discovered that satellite repeats make up 6.2% of the T2T-CHM13 human genome assembly, with α-satellite repeats representing the largest component, constituting 2.8% of the genome. By investigating the sequence relationships of α-satellite repeats across individual centromeres in newly sequenced genomes, it was found genome-wide evidence that human centromeric satDNAs evolve through a process known as "layered expansions." In this mechanism, distinct repetitive variants arise within centromeric regions and expand through successive tandem duplications, while older, flanking sequences shrink and diverge over time.

Similarly, studies of centromeres in *Arabidopsis* species using new TGS assemblies have revealed remarkable inter- and intra-species diversity and mechanisms of sequence diversification [40]. Research involving 68 populations across *A. thaliana* and *A. lyrata* demonstrated that *Arabidopsis* centromere repeat arrays are embedded in linkage blocks, despite ongoing internal satellite turnover. This finding is consistent with the idea that unidirectional gene conversion or unequal crossover between sister chromatids contributes to satDNA sequence diversification in *Arabidopsis* centromere.

1.3 *Tribolium* beetles

1.3.1 *Tribolium castaneum* as a model organism

Tribolium castaneum, the red flour beetle, has established itself as one of the most important model organisms in genetic and developmental research due to its advantageous features, such as its well-characterized RNA interference (RNAi) system and its comparative genetic insights, which often offer a more nuanced representation of gene function and evolution than other insect models like the most widely experimented species *D. melanogaster* [112]. Additionally, it was the among the first sequenced and assembled insect species, second only to *D. melanogaster*, with the first complete genome sequence

published in 2008 [113] and subsequently updated several times, with the most recent Tcas5.2 genome assembly published in 2020 [114]. For assembly of Tcas5.2 authors used large-distance jumping libraries and BioNano Genomics optical mapping to resolve problems with the previous versions and RNA-seq reads from different life stages producing the most complete gene set of *T. castaneum* to date, OGS3. Despite the exhaustive efforts, the complete genome sequence of *T. castaneum* is still missing almost 25% of its experimentally confirmed genome size [115], thus challenges remain, particularly in accurately representing repetitive regions, including satDNAs, which were estimated to comprise up to 42% of the genome [116].

1.3.2 SatDNAs of *T. castaneum*

The TCAST satellite DNA is a major satDNA in the genome of *T. castaneum*, making up 17% of its total genetic content. This satellite DNA consists of a monomer 360 base pairs (bp) long, characterized by a high A+T content (73%) and lacking significant internal substructures, which suggests a relatively simple repetitive sequence. Through fluorescent in situ hybridization (FISH), TCAST was shown to be distributed uniformly in the (peri)centromeric heterochromatin regions of all 10 chromosomes of *T. castaneum* [117]. In addition, chromatin immunoprecipitation experiments and immunofluorescence (IF)-FISH have shown that TCAST associates with cCENH3, a variant of the histone H3 protein that is specific to centromeric chromatin, suggesting an important role for TCAST may play in centromere function [118]. The structural organization of TCAST main satellite was found to organized in HOR organization, similar to the α satellite DNA in *H. sapiens* [119].

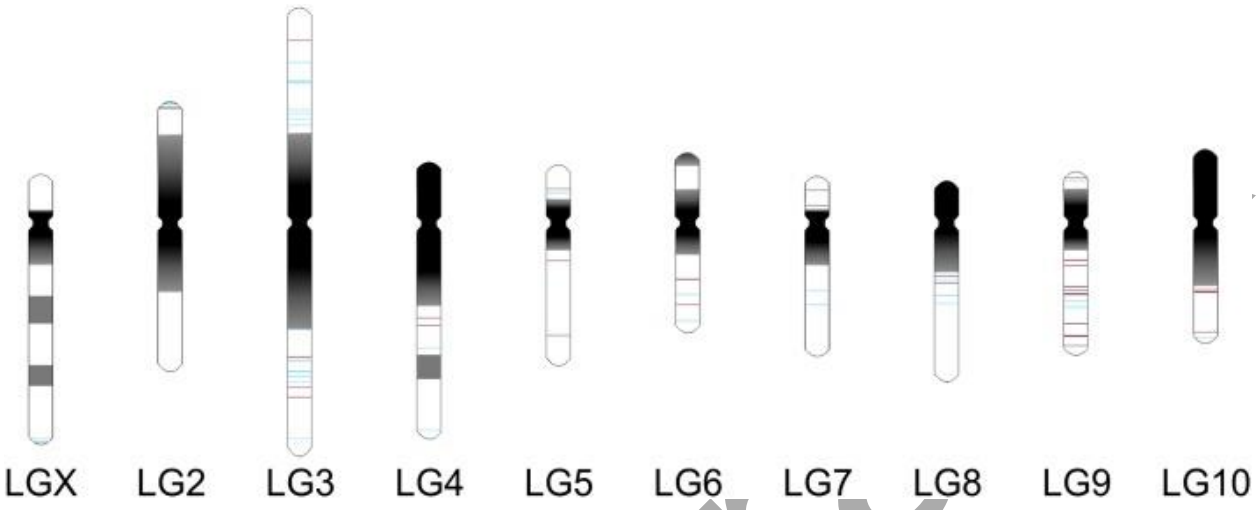


Figure 1.10 Karyogram of *T. castaneum* chromosomes together with the distribution of TCAST main satDNA with TCAST transposon-like elements (blue) and TCAST satellite-like elements (red). Visualization from [37]

Apart from TCAST, about 42% of the *T. castaneum* genome is composed of repetitive elements, which include transposable elements and other satellite DNAs [116]. Among this, approximately 4% of the genome consists of euchromatic satDNAs, distributed among nine distinct families (Cast1-Cast9). These satDNAs are significantly underrepresented with 0,4% abundance in the reference Tcas5.2 genome assembly. FISH experiments have indicated that these nine satDNA families localize almost exclusively to non-centromeric regions of the chromosomes [120]. Regarding the structure of these satDNAs, there is a notable correlation between the monomer length and the number of monomers in arrays, with a predominance of ~170 bp monomers in longer arrays. Analyses have also revealed a periodic distribution of A or T tracts (4–10 nucleotides) within these satDNAs, suggesting that unequal crossing over, a process predicted by computer simulations, plays a role in the homogenization of longer arrays. In addition to the 9 abundant satDNAs, recent research has identified 46 novel satDNAs that together comprise 1.2% of the genome[121]. These newly discovered sequences are predominantly 140–180 bp or 300–340 bp in length and, like all *T. castaneum* satDNAs, are also highly enriched in A+T content, ranging from 59.2% to 80.1%. Many of these satDNAs are organized into short arrays, often not exceeding five consecutive repeats, raising questions about their role in the genome and whether are these sequences merely "seeds" for future tandem expansions, or are they already established throughout unassembled genomic regions. This remains an open question, and further assembly and analysis will be necessary to fully elucidate the genomic landscape of *T. castaneum*. Despite significant insights into the composition and structure of the

euchromatic satDNAs, the need for a new continuous assembly is crucial to properly understand the evolutionary dynamics and mechanisms of propagation. This is due to the fact that many satDNAs are missing or incomplete in the current genome assembly, limiting the ability to analyze their full structure and genomic context. Therefore, considering genome gaps and the potential of nanopore sequencing new *T. castaneum* assembly based on ONT long-read sequencing, enriched in the repetitive regions, could be an excellent platform for global and in-depth analyses of the dominant satDNA fraction in euchromatin.

1.4 Isolation of high molecular weight DNA

The most critical factor for successful Nanopore long-read sequencing approach is extraction of high molecular weight (HMW) DNA of sufficient purity and quantity. Unfortunately, this step, which is a prerequisite for the successful sequencing of long fragments with a nanopore, is very difficult and often requires optimization for a specific organism. High molecular weight (HMW) DNA refers to isolated DNA fragments that are significantly longer and of higher quality compared to typical genomic DNA extractions such as those used by NGS sequencing technologies. Main characteristic of HMW DNA is its length, that often exceeds 50 kilobases (kb) and can sometimes span into the megabase (Mb) range [122]. The second feature is the quality of HMW DNA, which affects the efficiency and accuracy of downstream applications and allows for minimal degradation during various experiments. There are various methods to gauge isolated HMW DNA length and quality, such as pulsed-field gel electrophoresis (PFGE) for size estimation and using fluorometric assays or spectrophotometry for purity and concentration measurements using the A240/A260 and A220/A240 absorbance ratios. There are also specialized instruments for performing integrated length and quality checks the Agilent TapeStation or Bioanalyzer [123], which provide detailed fragment size distribution profiles, ensuring that the isolated DNA meets the stringent requirements needed for advanced genomic analyses.

The isolation of HMW DNA has evolved significantly since the first protocols described in 1973, which laid the foundation for its use in various genomic applications [122]. One of the earliest uses of HMW DNA was in genotyping, where the length and integrity of the DNA allowed for more precise identification of genetic variants across large genomic regions [124]. Furthermore, HMW DNA is crucial in structural variation studies, particularly using technologies like optical mapping, which rely on large, intact DNA molecules to visualize and characterize genomic rearrangements, insertions, deletions, and other large-

scale variations [125]. The importance of isolating HMW DNA has grown exponentially with the advent of third-generation sequencing technologies where high-quality HMW DNA is essential for these technologies to function optimally, as fragmented or degraded DNA would result in shorter reads, increased error rates, and reduced coverage of critical genomic regions [126].

Current commercial protocols for the isolation of HMW DNA, as offered by companies such as Qiagen, Thermo Fisher and Circulomics, typically rely on techniques such as salting out, phenol-chloroform extraction, separation by magnetic beads and column-based purification. Circulomics Nanobinding kit, for example, uses nanomagnetic disk technology that enables gentle binding and release of DNA, minimizes shear and ensures high purity and yield. The E.Z.N.A.® HMW DNA kit uses a combination of optimized salting-out and column-based protocols in which DNA is selectively bound to a silica membrane within the column in the presence of chaotropic salts. The Qiagen Genomic-tip, on the other hand, uses a technology based on anion exchange resins that gently binds DNA through ionic interactions. [127]

Although existing HMW extraction protocols attempt to address the unique requirements of different species and cell types, a variety of problems remain, particularly when dealing with hard tissue, whole organisms and samples with high levels of interfering substances. Common reasons for HMW protocols failing to deliver sufficiently long DNA molecules include shear forces during extraction due to forces during pipetting, mixing or centrifugation, incomplete cell lysis and nuclease contamination, and the presence of contaminants such as polysaccharides, lipids and secondary metabolites [128]. Consequently, there are numerous modifications of HMW extraction protocols, often focusing on one group of organisms, tissue types or even cell lines [126], and the transferability of these protocols to other species or cell types without major changes is often not so straightforward and requires careful optimization of lysis conditions, buffer compositions and mechanical digestion parameters to accommodate the specific biological and chemical properties of the new samples.

2. Aims and hypothesis

Satellite DNAs (satDNAs) are one of the most abundant repeated sequences and the fastest evolving part of the eukaryotic genome. To date, studies have been primarily focused on satellite DNAs in (peri)centromeric heterochromatin. Although there is clear evidence that some roles have been assigned to satDNAs, primarily in centromere structure, understanding of their organization, evolutionary dynamics, and molecular mechanisms driving their spread across the genome, especially in euchromatic regions, is still rather limited. One of the main reasons for the current lack of global and in-depth studies of satDNAs is certainly the fact that satDNAs are the most difficult part of the genome to sequence and assemble, and therefore they are underrepresented or even absent in the best genome assemblies.

The main objective of this work is investigation of evolutionary dynamics, mechanisms of propagation and transcriptional potential of ten different euchromatic satDNAs abundant which represent even 4,6% of the genome of insect model organism *T. castaneum*. First, the most contiguous genome assembly of *T. castaneum* to date with the significant improvement in the representation of the repetitive genome portion will be generated using Oxford Nanopore long-read sequencing. To this end, the high-molecular-weight (HMW) DNA of appropriate quality and length for Nanopore sequencing, which is essential for accurately assembling the complete euchromatic satDNA regions will be optimized for *T. castaneum*. The hypothesis is that a new, enhanced genome assembly will provide an excellent platform for studying the organization and evolutionary dynamics of euchromatic satDNAs. Comprehensive analyses of the new genome will also shed light on how these euchromatic satDNAs spread and diverge across different regions of the genome focusing on their relationship to genes and other repetitive sequences. Finally, the effect of recombination on the expansion and spread of the satDNA arrays will also be investigated.

Another important aspect of this research is the investigation of the transcriptional potential of satDNA during embryogenesis and development. As there is increasing evidence for satDNA transcription in different species, it is hypothesized that transcribed satDNAs in *T. castaneum* may play a significant role in genome regulation and other essential cellular functions. For this purpose, the expression profile of ten euchromatic satDNAs DNAs in *T. castaneum* will be determined to reveal the patterns of their expression throughout different developmental stages, from embryogenesis to later life cycles.

These results will provide a deeper understanding of how euchromatic satDNAs contribute to genome evolution, regulation, and structural integrity and shed light on their broader influence on the genome.

3. Material and methods

3.1 DNA isolation and sequencing

Insect samples

Laboratory cultures of the red flour beetle, *T. castaneum*, specifically the highly cultured Georgia 2 (GA2) strain, were routinely maintained in whole wheat flour, supplemented with whole rye flour and oats. The rearing conditions were optimized for faster reproduction, set at 32 °C and 70% relative humidity, and kept in darkness. Different life stages of the beetles were separated by sieving through a 0.71-mm sieve, and individual beetles were picked with tweezers in different quantities in order to achieve best DNA yield. Therefore, to ensure efficient DNA isolation for genome assembly, larvae and pupae were collected in sufficient quantities, with 200 mg of pupae and 500 mg of larvae used for each DNA extraction.

Nuclei isolation

The nuclei were isolated following a modified version of the Brown and Coleman protocol [129]. Several changes were made to optimize the process. Instead of using −80 °C, the mortar and spatula were precooled with liquid nitrogen. Fresh NIB buffer was prepared immediately before use, and an additional washing step was introduced for the isolated nuclei. Centrifugation times were adjusted, and standard plastic tubes were used for convenience. To begin, 20 mL of freshly prepared NIB buffer per reaction was chilled on ice. The mortar and spatula were filled twice with liquid nitrogen to ensure adequate cooling. During the second nitrogen evaporation, the sample, as specified in Table 4.1, was added to the mortar and ground into a fine powder using increasing pressure and speed. The powder was then scraped into a 50 mL tube containing 8 mL of chilled NIB buffer using the precooled spatula. The tube was gently swirled to mix the suspension, and if any residue stuck to the tube walls, a wide bore tip was used to flush it down, ensuring maximum efficiency. Care was taken to avoid shaking the tube, as this could disrupt the suspension. The mixture was filtered through a 100 µm cell strainer into a new chilled 50 mL tube. Next, the solution was divided into six chilled 1.5 mL tubes and centrifuged at 100× g for 30 seconds at 4 °C. The supernatant was carefully transferred into six new tubes without disturbing the loosely attached pellet of cell debris. These tubes were centrifuged again at 1800× g for 3 minutes at 4 °C to pellet the nuclei. After the supernatant was discarded, any remaining liquid was removed with a pipette. The compact nuclei pellet was resuspended in 1 mL of cold NIB buffer using a wide bore tip, being careful not to introduce air bubbles into the mixture. This step was repeated to ensure thorough resuspension.

DNA isolation

Lysis buffer was prepared by adding 500 µL of protease or 95 µL of proteinase K, along with 10 µL of RNase A, to 5 mL of G2 buffer. After the final centrifugation, the nuclei pellet was resuspended in 800 µL of G2 buffer. Complete resuspension was achieved by pipetting gently with a wide bore tip, again taking care to avoid introducing air bubbles. The tubes were incubated at 50 °C for 1 hour at 300 rpm in a thermomixer, with intermittent gentle inversion or pipetting to ensure complete digestion. The properly digested nuclei had a visible, stringy, milky texture. If clumps of nuclei remained, they were further broken by additional pipetting. The genomic DNA was extracted using a Qiagen Genomic Tip 100/G column, following the manufacturer's instructions with slight modifications. Pressure was applied at all stages to ensure efficient flow, and the QF buffer was prewarmed. The column was equilibrated with 4 mL of QBT buffer, and the digested sample was applied to the column. The column was washed twice with 7.5 mL of QC buffer, and DNA was eluted with 5 mL of prewarmed QF buffer. The eluted DNA was precipitated by adding 3.5 mL of isopropanol at room temperature. The solution was allowed to stand for 30 seconds, during which the upper phase turned whitish. The tube was inverted multiple times, causing white strands of DNA to appear. These strands formed a sticky DNA "jelly," which was then spooled onto a thin glass rod. The spooled DNA was transferred to a 1.5 mL DNA LoBind tube containing 100 µL of elution buffer. After incubation at 50 °C for up to 2 hours, the DNA was mostly dissolved. It was then left overnight with gentle shaking to achieve full relaxation before being stored at 4 °C, where it remained stable for several months.

Assesment of quality and length

DNA concentration was consistently measured using both fluorometric and spectrophotometric methods. The quality of the DNA was evaluated with a spectrophotometer, with acceptable A260/280 and A260/230 ratios being around 1.8 and 2.2, respectively, in line with ONT's official guidelines. The length of isolated DNA, sheared DNA, and the prepared library was assessed using pulsed-field gel electrophoresis (PFGE). DNA fragments were separated on a 1% agarose gel in 0.5× TBE buffer, run at 6 V/cm, 14 °C, with a 120° included angle, and switch times ranging from 1 to 10 seconds over 14 hours using a Bio-Rad CHEF-DR III PFGE system. The gel was subsequently stained with 1 µg/mL ethidium bromide solution at room temperature for 30 minutes on a shaker.

DNA shearing and size selection

The homogenized DNA solution was sheared 10–30 times using a 30-gauge needle. Its concentration was measured in triplicate and adjusted to 150 ng/µL using TE buffer or water. For size selection, the Short Read Eliminator (SRE) XS kit was applied according to the manufacturer's instructions. The final resuspension was carried out in 50 µL of EB buffer from the SRE kit, and the concentration was measured twice to ensure reproducibility.

Sequencing and basecalling

The Oxford Nanopore library was prepared using the SQK-LSK110 kit, following the manufacturer's instructions with specific modifications. The sheared and purified DNA was used for library preparation at double the amount recommended by the ONT protocol. Additionally, all elution and incubation times were extended to twice the suggested duration to prevent library loss and increase the final concentration. This ensured the library concentration was over 100 ng/µL, allowing multiple loads on MinION flow cells. The flow cells were washed and reloaded 2–5 times to maximize data output. A total of twelve MinION flow cells (versions 10.3.4 and 9.4.1) were used for method development and assembly data generation, resulting in a cumulative data output of 89.9 GB with an N50 of 20.1 kb. The sequencing was managed using the Oxford Nanopore MinKnow software version 20.10.3. Basecalling was performed using Guppy v5.0.1.

3.2 Genome assembly

Assembly

The basecalled reads were utilized in the assembly process using Canu v2.2 [87], with parameters specified in Table 3.1. Adjustments were made according to Canu documentation to account for the genome's high repetitiveness [87] and the elevated AT content in the reference Tcas5.2 genome assembly [114]. To manage computational demands and the small size of the *T. castaneum* genome, reads were filtered to those greater than 20 kb. The Canu assembly was carried out using the Isabella computer cluster at the University Computing Centre (SRCE), University of Zagreb.

Contig placement

To arrange the Canu contigs into chromosomes based on the Tcas5.2 (GCF_000002335.3) assembly, pre-existing gaps in the Tcas5.2 assembly needed to be bridged. This was accomplished using TGS-Gapcloser software [98] with default gap-filling settings and the corrected reads from the Canu pipeline. Gap filling

addressed small and medium gaps in the Tcas5.2 assembly, preventing interruptions in the long contigs generated by Nanopore sequencing. After gap filling, the RagTag software tools [97] were used to further refine the assembly. Canu contigs were used as the query sequence, while the gap-filled Tcas5.2 assembly served as the reference with the "scaffold" parameter. RagTag aligned and placed the Canu contigs onto the gap-filled Tcas5.2 assembly, incorporating previously missing repetitive regions and filling gaps with successfully aligned contigs. This approach helped integrate repetitive elements absent from the original Tcas5.2 assembly. RagTag mapped high-confidence genomic regions onto chromosomes and placed contigs ending or beginning with repetitive regions into gaps, revealing previously unknown regions. The result was an unpolished assembly that served as a template for subsequent polishing.

Polishing and gene completeness analysis

To enhance the assembly quality and reduce the error rate, correction of the TcasONT assembly based on the Canu contigs was performed. Two rounds of RACON [130] polishing were carried out using short reads (<20 kb) that were excluded from the initial assembly. These excluded reads, totaling approximately 50 Gb, provided significant additional genomic information. Polishing followed the RACON documentation, which involves mapping the reads onto the assembled genome with minimap2 [131] and using the mapped reference reads for polishing. The polished assembly, named TcasONT, was then used for downstream analysis. Benchmarking Single Copy Orthologs (BUSCO) [132] analysis was conducted using the BUSCO v5.0.0 module on the Galaxy web platform (usegalaxy.org), with the same settings as listed in Table M1 applied for all assembly validations.

Repeat annotation

RepeatMasker, a widely used tool for identifying and masking repeat elements in target sequences [133], was employed to obtain GFF/GTF formatted data detailing the position and orientation of classified RepBase repeat elements. This data provided information on the quantity, size, and distribution of various repeat elements within the genome assemblies. Assemblies were annotated with repeat elements using RepeatMasker on the Galaxy web platform (usegalaxy.org), utilizing repeat data from the latest RepBase database (RELEASE 20181026) and the "Hexapoda" species listing for clade-specific repeats. RepeatMasker was also rerun for the Tcas5.2 assembly to update repeat annotations, as the original annotations were based on an earlier version of the RepBase database. For quantifying three classes of

satDNAs (with defined monomer lengths >50 bp, 50-500 bp, and >500 bp) in the TcasONT and Tcas5.2 assemblies, the Tandem Repeat Finder (TRF) program [100] was used with default parameters.

Table 3.1 Command line arguments used in key steps of assembly generation.

STEP	SOFTWARE	VERSION	PARAMETERS
SEQUENCING AND BASECALLING	MinKnow	40350	GUI
	Guppy	5.0.11	--config dna_r10.3_450bps_hac.cfg -x cuda:0
ASSEMBLY AND POLISHING	Canu	2.2	genomeSize=200m minReadLength=20000 corMaxEvidenceErate=0.15 ovlMerThreshold=500 gridEngineResourceOption="-l mem=MEMORY"
	TGSGapfiller	v1.0.1	--thread 14 --min_match 500 --ne
	RagTag	v2.1.0	scaffold -f 50000 -t 16
	minimap2	r1101	-ax map-ont
	RACON	v1.4.3	default
ANNOTATION	Liftoff	37043	-g 52.gff -chroms -copies
	RepeatMasker	4.0.9_p2	Galaxy settings: "repeat source species = hexapoda" "output=ggf"
	BUSCO	5.0.0	Galaxy settings (lineage=Insecta, Augustus species=Tribolium castaneum)

Transfer of gene annotations

To map genes in Canu contigs (filtering) and the TcasONT assembly, the LiftOff package [99] was used in conjunction with gene annotations from the Tcas5.2 assembly. LiftOff first maps the entire TcasONT assembly to the reference Tcas5.2 and then aligns the gene sequences from the Tcas5.2 reference to the target TcasONT based on these overlaps. Although this method is limited in finding potential new genes in the improved TcasONT assembly, it ensures that the comprehensive annotations of the Tcas5.2

assembly, based on an extensive RNA-seq database and gene prediction methods, are accurately mapped and transferred onto the TcasONT assembly.

3.3 Identification and analysis of satDNAs

Identification of satDNA repeats

Satellite repeats within the genomes were annotated usin the standalone NCBI BLAST algorithm and the *metablastr* package [134] in R. The subject sequences were the analyzed assemblies (Tcas5.2 and TcasONT), and the queries were the previously characterized Cast1-Cast9 [120]. All of the detected hits were retained in a database of hits, and following the analysis, the database was filtered to identify trends and arrays for Cast1-Cast9.

Analysis of satDNA arrays

All Cast1-Cast9 monomers were identified from the BLAST result table and filtered according to the parameters described in Figure 4.10. To avoid fragmentation due to potential short sequence variations within the arrays, it was essential to establish optimal parameters for satDNA array detection. Total arrays for each Cast satDNA were analyzed to determine the best neighboring window length that would connect continuous repeating monomers into a single array. This method was implemented to account for errors and insertions and to accurately link all monomers of a given satellite. Basic filtering was then performed to define arrays and remove short, interspersed monomers using custom parameters for each satDNA family to ensure that arrays contained at least 3 repeat units for each satDNA, except for the Cast2' array (Cast2 monomer interspersed with the newly discovered sequence Cast2'), which included three different length monomers, with the 1100 bp Cast2' mixed with 170 bp Cast2.

Detection of array edges

To accurately determine the edges of Cast1-9 arrays in the genome, a refined strategy was employed. Traditional monomer detection methods, which typically rely on a fixed cutoff based on monomer similarity, often struggle with the degenerate nature of array edges, making it challenging to identify small homology regions and junctions. Therefore, several steps were taken: first, a database of all monomers for each satDNA was created, along with a database of all arrays and their flanking regions (500 bp). K-mers of 32 bp were extracted from both the monomers and the extended arrays with flanking regions. For each position within the extended array, the closest k-mer match from the monomer database was identified based on Hamming distance, and the score was recorded. A rolling mean position score was

then calculated by averaging scores from ±5 positions. The true edges of the arrays were determined by identifying the minimum and maximum positions for each array where the distance was less than 5. Based on these newly defined edges, surrounding and microhomology regions were extracted.

Analysis of gene content

A 50 kb region upstream and downstream of each Cast1-Cast9 array was selected to define gene profiles around these arrays. In each 100-kb region (50 kb upstream and 50 kb downstream), the area was divided into 100 bins of 1 kb each. The number of exons was counted in each bin to profile genes around the different Cast1-Cast9 arrays. Expected exon densities were determined by calculating the median, 1st quartile (1Q), and 3rd quartile (3Q) exon densities across the genome in 100-kb sliding windows using a custom R script.

Multiple sequence alignment and clustering

MAFFT [135] was used to perform multiple sequence alignments of Cast1-9 monomers in the assembly. After alignment, the "F81" genetic distance evolutionary model from the *ape* package [136] was applied on the alignments to generate genetic distance matrices. These matrices were then used for PCA analysis, which was conducted using the PCA function from the FactoMineR package [137]. The first two dimensions of the PCA results for each satDNA were visualized using *ggplot2* [138].

Visualizations and statistics

All plots and calculations were generated in R using custom data processing notebooks. In addition to standard libraries, the circlize package was employed to create circular visualization plots illustrating global genome patterns. To construct the complex heatmaps used for analyzing the similarity of neighboring regions, the ComplexHeatmap package was utilized. A graph-based visualization method was implemented to tackle the low variation among satDNA monomers and their tendency for intra- and interchromosomal exchange, as seen in the mixing in PCA plots. To generate the graph networks, for each monomer in each array, we identified the five closest monomers outside the same array using the dist.dna function from the ape package in R, applying the "F81" genetic distance model. The resulting data was visualized as a graph network with the networkD3 package. In these visualizations, clustered and connected nodes represented potential satDNA arrays involved in frequent exchange, while disconnected nodes suggested lower interaction. Homology in 20 bp regions flanking the arrays was visualized with the ggseqlogo package, following alignment using MAFFT.

3.4 Extrachromosomal circular DNA

Extrachromosomal circular DNA on agarose gels

Two-dimensional agarose gel electrophoresis was conducted following the method described in [139], with several modifications. Total DNA was extracted from 500 mg of *T. castaneum* pupae using standard phenol-chloroform extraction and dissolved in Tris-EDTA buffer (pH 8.0). The DNA concentration was measured using a Qubit 4 fluorometer (Invitrogen). To shear the linear DNA, 20 µg of the extracted DNA was passed through a 0.33 mm hypodermic needle 25 times. Since the linear double-stranded DNA (dsDNA) fragments greatly outnumber potential extrachromosomal circular DNA (eccDNA) molecules in the total genomic DNA (gDNA) isolate, the gDNA was treated with exonuclease V to selectively degrade the linear dsDNA. Exonuclease V (New England Biolabs) digests linear dsDNA from both the 5' and 3' ends. This overnight digestion at 37 °C was intended to remove as much linear dsDNA as possible while preserving the circular DNA. The reaction was halted by adding 11 mM EDTA (pH 8.0) followed by incubation at 70 °C for 30 minutes. The DNA was then purified using the Monarch PCR & DNA Cleanup Kit (NEB). The first dimension of electrophoresis was run in 0.7% agarose gel in 1× TBE buffer at 0.7 V/cm for 18 hours. After this, the gel was stained in 1× TBE buffer containing 0.2 µg/mL ethidium bromide. A lane with the separated DNA was excised, and 1.5% agarose containing 0.2 µg/mL ethidium bromide was poured around the lane, positioning it at a 90° angle relative to the first run. The second dimension of electrophoresis was carried out at 4 V/cm for 3 hours.

Southern blot hybridization

To ensure efficient DNA transfer from the agarose gel to the membrane, the gel was first rinsed in 0.25M HCl for 30 minutes, followed by a 30-minute rinse in 0.4M NaOH. The DNA was then transferred overnight onto a positively charged nylon membrane (Roche Life Science) using capillary transfer. Hybridization probes for Cast1, Cast2, Cast5, and Cast6 satellite DNA were labeled with biotin-16-dUTP (Jena Biosciences) through PCR amplification of cloned plasmids containing the respective satellite DNA sequences. Specific primers were used for each satellite:

- Cast1: 5' AAGTCGGCTACGACTAACCGTTC 3' and 5' TTGCAAATTTGGATTCCGCCCGG 3'
- Cast2: 5' TATACGCAAAATGAGCCGC 3' and 5' AAAGTCGTAGAGCAATGCGG 3'
- Cast5: 5' GGTGTTGAAAAGTCATAARTTGAGTG 3' and 5' AGAGCCGGTGTACACAACATT 3'
- Cast6: 5' CGACGCATGGGTCAATCTAAGACA 3' and 5' ATTCGAAACTTTTCAAAAAAATTGG 3'.

Hybridization followed the protocol outlined in [120]. Detection was performed using streptavidin-alkaline phosphatase and the chemiluminescent substrate CDP-star (Roche Life Science), with visualization conducted on the Alliance Q9 Mini (Uvitec) imager.

3.5 Small RNA sequencing and analysis

Isolation of small RNAs from life stages

The first step in RNA isolation involved sorting the various life stages of *T. castaneum*. A 0.71 mm sieve was used for initial sorting, and beetles were manually picked. Three life stages (larvae, pupae, and adults) were sorted, with pupae and adults further separated by sex, resulting in five distinct samples. For each sample, only the heads were collected by cutting approximately 30 mg of heads (100-200 depending on life stage) on ice, then immediately transferring them to a vial in liquid nitrogen. Total RNA was extracted using the Quick RNA Miniprep Plus Kit (Zymo) with 30 mg of starting material per reaction, and two biological replicates were prepared for each sample. The collected tissue was placed in 500 µL of RNA/DNA Shield Solution (Zymo) and homogenized using an electric homogenizer and pestle. Lysis was performed with 15 µL of proteinase K and 30 µL of PK digestion buffer, followed by incubation at 26°C for 2 hours. Subsequent steps followed the manufacturer's protocol, with final elution in 50 µL of RNase-free water (Invitrogen). RNA quantity was assessed using gel electrophoresis and a Nanodrop spectrophotometer (>100 ng), while RNA integrity was validated (RIN >9) using the Qubit IQ RNA Kit to confirm the presence of different RNA sizes.

RNA sequencing

Sequencing was performed using RealSeq Biosciences Inc. (CA, USA) provider. Small RNA library was prepared with the RealSeq-AC kit and sequenced on an Illumina NextSeq 500 v2 device and High-Output - SR 75 Cycle with read lengths of 75 pb in one direction. Average number of reads passing filter per sample was 10M.

Public data

RNA sequencing data from Ninova et al. [140] were accessed via the NCBI Gene Expression Omnibus (GEO) under accession number GSE63770. The miRNA and target regions were downloaded from iBeetleBase [141] . Subsequently, these regions were extracted from the Tcas5.2 genome [114], deduplicated, and mapped to satDNA sequences using Bowtie [142].

Read mapping and analysis

For read mapping and analysis, all reads—both public and newly generated—were processed by the same pipeline to insure data compatibility. First the reads were trimmed of adapter sequences using Trim Galore [143], retaining those longer than 18 nt and applying the –small_rna preset. Following adapter and quality trimming, the reads were aligned to satDNA sequences using the Bowtie aligner with the parameters: -p 8 -S --no-unal. The resulting alignments were sorted, filtered, and processed using Samtools [144] with the functions “samtools depth” and “samtools bedcov.” Further processing of the alignment files was conducted in R using the Rsamtools package [145] .

4. Results

4.1 Development of new HMW DNA isolation protocol

The starting point for the isolation of HMW DNA are commercial kits with species/tissue-specific modifications. The commercial kits and their official protocols that were tested and optimised during the DNA extraction process were the E.Z.N.A kit (Omega BioTek, Norcross, GA, USA), the Monarch HMW DNA extraction kit for tissue (New England Biolabs), the Blood and Cell Culture DNA Mini and Midi kit (Qiagen), and the standard phenol-chloroform extraction, which were initially tested for isolation and library preparation and the final sequencing step. Different problems were encountered that were specific to each of the available kits. For example, the E.Z.N.A. kit had the limitation that the extracted HMW DNA was relatively short and often produced DNA less than 50kb in length; the Monarch kit was found to be non-reproducible and the DNA had low absorbance ratios due to the very mild washing step. The Qiagen columns were often blocked even at the purest pupal stage as chitin residues reduced yields and lengths, and the phenol-chloroform extraction, while able to produce HMW of sufficient quality, is very labour intensive, and DNA purity as measured by absorbance ratios was suboptimal. When this DNA was introduced into standard ONT library preparation protocols, clumping of magnetic beads and large losses occurred after each step of library preparation, resulting in poor quality downstream libraries with insufficient read lengths in nanopore sequencing and rapid pore death. Since commercially available kits could not produce HMW in sufficient quality and quantity, a protocol for the extraction of HMW from purified cell nuclei was developed that includes a purification step using commercially available Genomic Tip columns followed by DNA shearing and size selection.

The optimized procedure was tested on all *T. castaneum* developmental stages (larvae, pupae, adults), along with two other Tribolium species (T. freemani, T. confusum) (Table 4.1). The DNA obtained had absorption ratios in the proposed range for Oxford Nanopore sequencing (Table 4.1).

Table 4.1 Results of the DNA isolation using developed protocol performed on different Tribolium species as well as on their various developmental stages.

Species	Stage	Starting material (mg)	DNA concentration (ng/µL)	DNA yield (µg)	$A_{220/260}$	$A_{260/280}$
T. castaneum	Pupae	200	172	17.2	1.89	1.95
		200	130	13.0	1.85	2.14
		200	138	13.8	1.84	2.25
		200	154	15.4	1.88	1.78
	Larvae	1100	512	51.2	1.87	2.35
	Adults	1000	643	64.3	1.85	2.14
T. freemani	Adults	1050	327	32.7	1.83	2.24
	Larvae	920	540	54.0	1.88	2.05
	Adults	620	78	4.7	1.94	2.00
T. confusum	Pupae	340	213	12.8	1.87	2.41

Due to the presence of large amounts of non-cellular material in the adults, mainly chitin in the form of the beetle cuticle, and large amounts of fat and intestinal tissue in the larval DNA, higher amounts of starting material (>600 mg) were required to produce sufficient amounts of HMW DNA, whereas only 200 mg of starting material was needed for isolation from relatively pure pupae. The size distribution of the isolated HMW DNA was analysed by Pulsed Field Gel Electrophoresis (PFGE). The extracted DNA from *T. castaneum* had the highest number of gDNA fragments distributed between 50 and 150 kb (Figure 4.1a). The DNA isolated from the pupal stage even showed an additional band at 200 kb. In addition, the same procedure to isolate HMW DNA was also tested for two congeneric species, T. confusum and T. freemani, which yielded gDNA with a length of up to 100 kb (Figure 4.1b). To increase sequencing efficiency, library preparations of sheared DNA were also tested. After shearing, both pupal and larval DNA showed a reduction in the ultra-long DNA fraction, with most DNA falling in the 30–80 kb range. Further testing of shear intensity revealed that 30 passes through the G30 needle resulted in the most compact band, with the majority of DNA still above 48 kb (Figure 4.3c). Notably, gel electrophoresis showed no significant increase in the abundance of shorter fragments, which is critical for subsequent sequencing. Size selection

on the sheared DNA had only a slight negative impact on the DNA length in the PFGE, probably due to the additional centrifugation and handling steps, as shown by a slight downward shift.

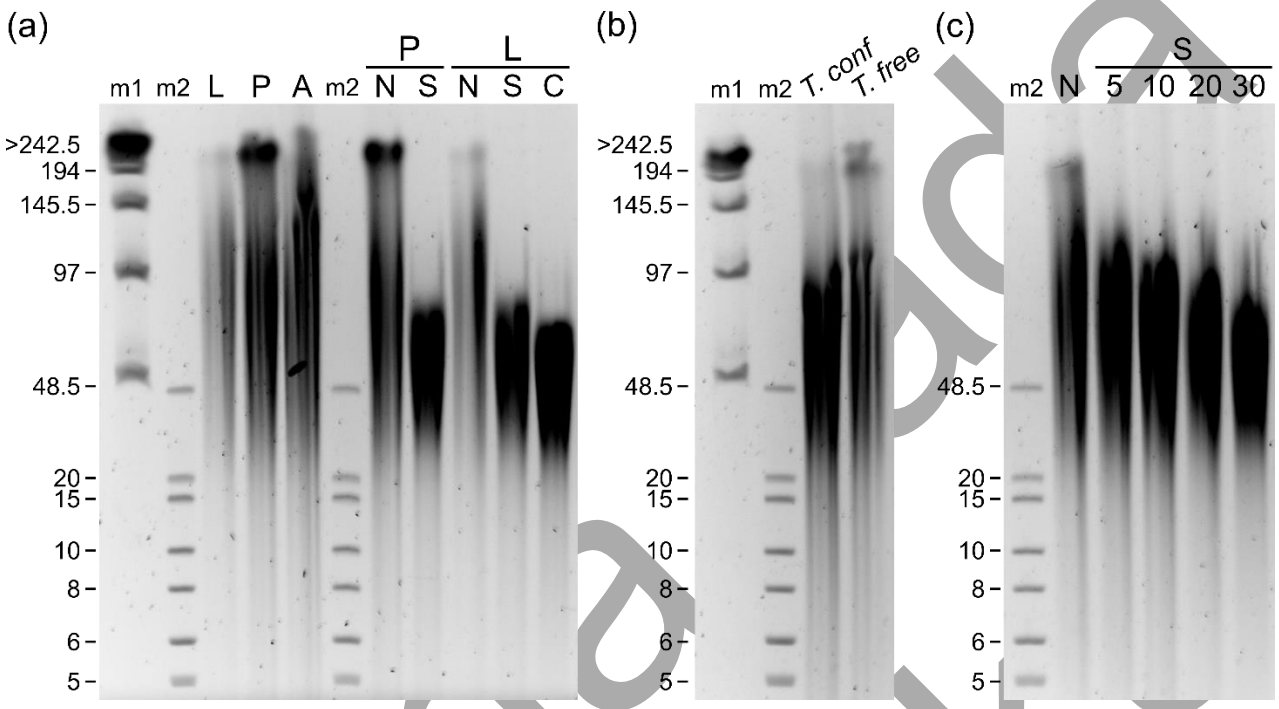


Figure 4.1 Pulsed field gel electrophoresis PFGE performed on isolated genomic DNA, along with sheared and cleaned fractions from various developmental stages of *T. castaneum*, *T. confusum* (T. conf), and *T. freemani* (T. free) beetles. Lambda DNA (m1) and Extend DNA ladder (m2) used as molecular weight markers. Approximately 1 µg of DNA from each sample was mixed with loading dye and loaded per well. **A** genomic DNA isolated from *T. castaneum* at different developmental stages (L-larvae, P-pupae, A-adults), alongside sheared (30×) and size-selected fractions. N represents non-sheared DNA, S is sheared DNA using a G30 needle, and C is the size-selected DNA using the Short Read Eliminator Kit XS. **B** displays genomic DNA isolated from *T. confusum* pupae and *T. freemani* adults. **C** effect of increasing needle shear passes (indicated by numbers) on *T. castaneum* adult genomic DNA.

4.2 Evaluation of HMW DNA by Nanopore sequencing

After the newly developed protocol was tested with PFGE, it was further evaluated on the Nanopore sequencing platform. An additional modification was introduced by doubling all wait times specified in the official protocol. The results of the sequencing runs are depicted on Figure 4.2 and Figure 4.3. As can be seen, the protocol combined with the adjustment resulted in a threefold increase in N50 compared to the Blood and Cell Culture kit, and a twofold increase in N50 compared to the Monarch kit. The read length distribution was also improved, when comparing the BCCD length distribution Figure 3a to the non-size selected output of newly developed protocol Figure 4.3b and Figure 4.3c. The E.Z.N.A. kit was not

tested, as it had previously produced the shortest DNA fragments. Additionally, the newly developed protocol showed the best reproducibility in the sequencing runs when compared, with each new sequencing run producing an equal or better N50 value for the sequenced reads.

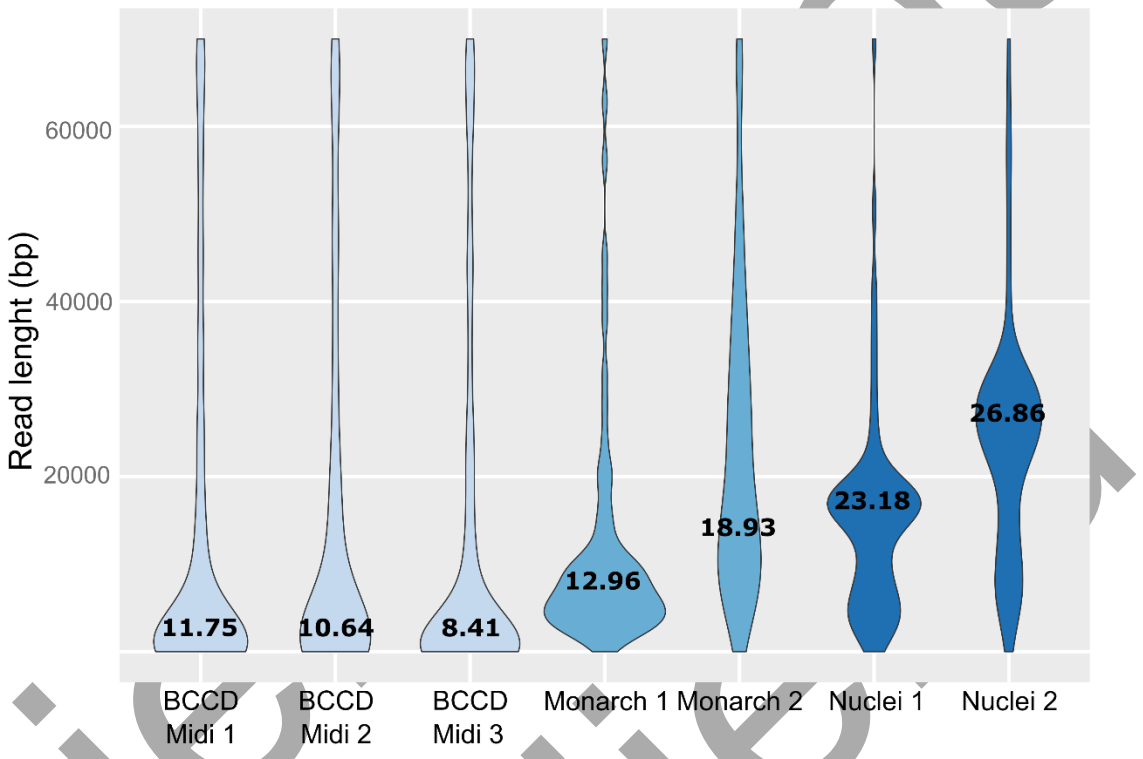


Figure 4.2 Violin plot of read length distribution of Nanopore sequencing data using different sequencing protocols, Blood and Cell Culture DNA Midi kit (BCCD Midi), the Monarch HMW DNA extraction kit for tissues (Monarch), and the newly developed protocol (Nuclei). Appended numbers represent experimental replicates, with N50 values (in kilobases) indicated by the numbers.

An additional step was later introduced which focused on filtering short reads present in the sequencing output by using the Circulomics XS short read eliminator kit which uses a centrifugation-based size selection process. This method effectively removed the majority of DNA reads under 10 kb, a crucial step for improving the quality of sequencing data. Although the removal of these shorter fragments was not visibly apparent in PFGE, it had a significant impact on the sequencing run. The effectiveness of this size selection can be clearly observed in the sequencing data, specifically in the length histogram. The absence of the leftmost peak, which corresponds to the shortest reads, highlights the successful elimination of

these fragments (Figure 4.3d). This adaptation plays a crucial role in improving the overall performance of sequencing, as shorter reads usually result in poorer quality and less efficient data output.

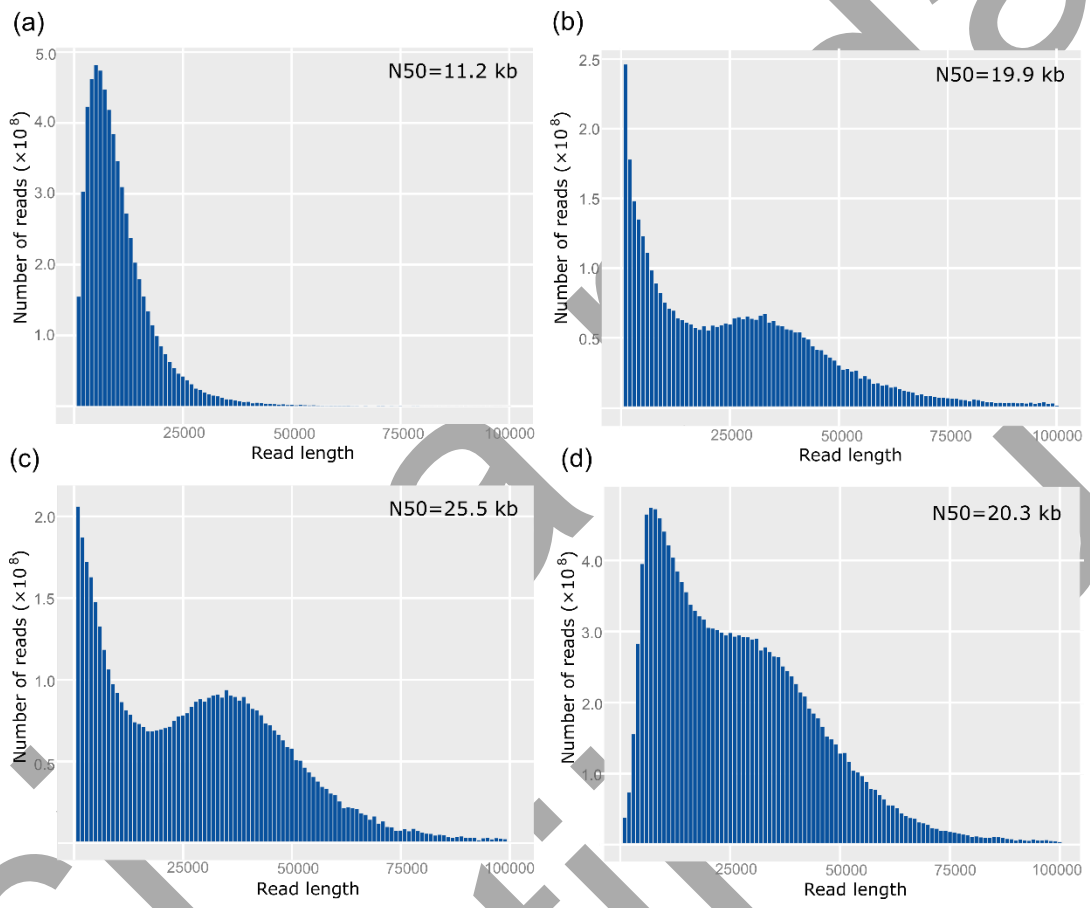


Figure 4.3 Read length distribution graphs Length distribution graphs show the correlation between genomic DNA (gDNA) shearing and size selection in Nanopore sequencing, with corresponding N50 values displayed in the top right corner of each graph. **A** Unsheared DNA **B** DNA sheared with 20 passes through a G30 needle **C** DNA sheared with 30 passes through a G30 needle **D** DNA sheared with 30 passes through a G30 needle followed by size selection using the Short Read Eliminator kit.

The cumulative output of Oxford Nanopore sequencing using the new protocol showed stable continuous growth throughout the sequencing experiment, which can last up to 72 hours. By washing the flow cell multiple times, with each new wash allowing near-perfect recovery of the pores, five consecutive loads could be performed within 48 hours, yielding 13.17 Gb of data (Figure 4.4a). The distribution of the Phred quality score (Q) shows that the majority of reads have a quality score above Q20, which means an error

rate of less than 1% (Figure 4.4b). There is also a positive correlation between quality and read length, with the longest reads having particularly high accuracy. In fact, most of the longest reads achieved a Phred score of Q24, corresponding to an accuracy rate of 99.6%, while the low-quality reads correlated mainly with their shorter length. This relationship between read length and quality emphasises the overall reliability of the sequencing data, especially for the longer fragments.

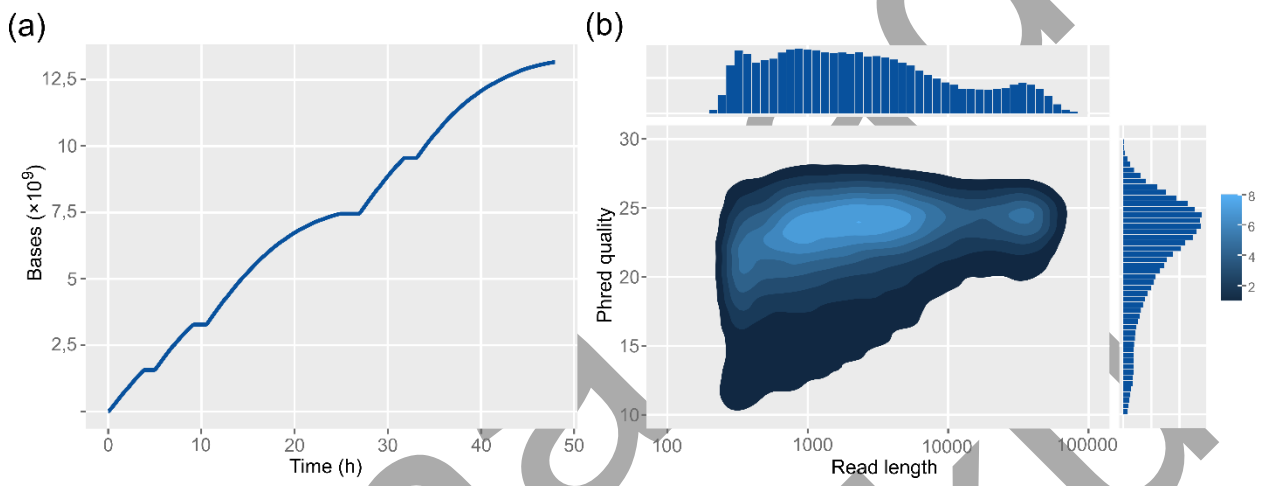


Figure 4.4 Summary of Nanopore sequencing run output. **A** The cumulative base output after a 48-hour run on a MinION flow cell, involving five consecutive library loads and four intermediate DNase washes. **B** 2D density plot showing the distribution of overall read Phred quality scores and read lengths, where lighter color shades indicating the higher cumulative fractions of reads with specific lengths and quality scores.

The sketch of the newly developed protocol can be seen in Figure 4.5. It is explained in detail in the section "Material and methods". Briefly, the nuclear isolation protocol is based on Brown and Coleman's method, with modifications including the use of liquid nitrogen to pre-chill the mortar and spatula, the preparation of fresh NIB buffer immediately prior to use, and the adjustment of centrifugation times. The isolated nuclei were carefully prepared with additional washing steps and the suspension is passed through a cell strainer. The nuclei are pelleted, resuspended in G2 buffer with protease and digested at 50 °C to produce a milky, stringy solution. After digestion, the genomic DNA is purified using a Genomic Tip column, applying pressure to maintain flow, followed by elution with pre-warmed QF buffer. The DNA is precipitated with isopropanol, spooled onto a glass rod and transferred to a DNA LoBind tube. The DNA is then incubated at 50 °C for up to 2 hours for homogenization and left at room temperature overnight to allow it to relax completely, resulting in a clear, viscous solution. The isolated DNA is stable at 4 °C for

months. The DNA is then sheared with a 30-gauge needle and size-selected using the Short Read Eliminator Kit. Quality should be assessed by spectrophotometry, with acceptable absorbance ratios and DNA lengths checked by pulsed-field gel electrophoresis. For nanopore sequencing, 3–3.5 µg of DNA is used per reaction and incubation times are doubled according to the beads-free library preparation protocol. Each flow cell is loaded with 400–600 ng of library DNA to allow multiple sequencing runs.

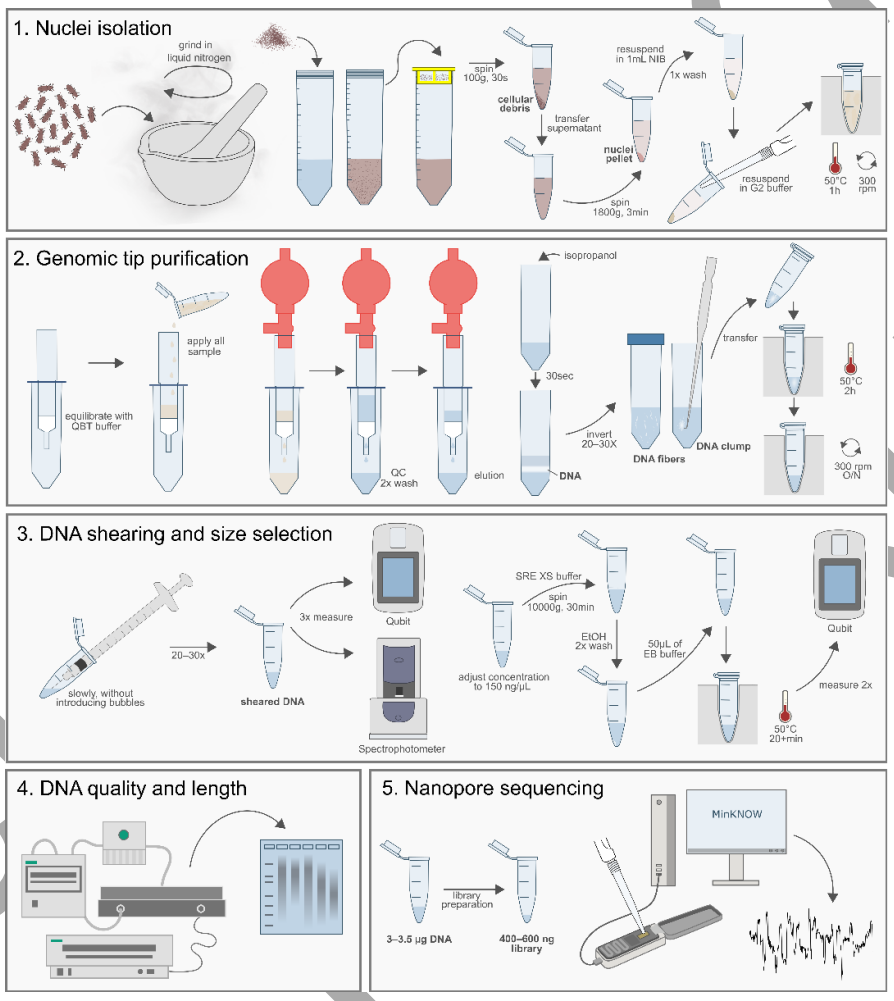


Figure 4.5 The workflow of the newly developed sequencing protocol for hard-cuticled beetles for Nanopore sequencing. Courtesy of Evelin Despot Slade.

4.3 Nanopore sequencing and genome assembly of *T. castaneum*

4.3.1 Nanopore sequencing

The HMW DNA prepared with the new protocol from different life stages was sequenced on MinION flow-through cells. The summary of the sequencing runs on a total of 6 flow cells is shown in Table 1. The sequencing experiments yielded a total of 5,688,065 analyzed reads covering a total of 89.87 billion bases, which corresponds to a 436X coverage of the estimated 204 MB genome size of *T. castaneum*. Furthermore, the mean read length of all generated reads is 15,799.3 bases with a standard deviation (STDEV) of 11,872.4 and a mean read length of 11,768 bases. The N50 value, a critical measure indicating the length at which 50% of the total bases are contained in reads of that length or longer, is 20,119 bases, which means that at least 22x coverage is achieved at >20kb, allowing proper assembly. The mean read quality score is 12.8, with the median read quality slightly higher at 12.9. Regarding read quality, 88.9% of the reads have a quality score greater than Q10 (<10% error rate), 65.7% exceed Q12 (<6.2% error rate), and 15.5% surpass Q15 (<3.1% error rate). As for read length distributions, 5,076,810 reads are longer than 1,000 bases, 4,123,341 reads exceed 5,000 bases, and 3,395,112 reads are greater than 10,000 bases. Most importantly 945,171 reads are longer than 25,000 bases, and 117,567 reads exceed 50,000 bases. These long reads account for a substantial portion of the total base count, with 35.29 billion bases in reads above 25,000 bases and 7.26 billion bases in reads above 50,000 bases representing a 40x coverage from >50kb reads alone. These statistics of Nanopore output data suggest that any genome assembly using these reads will be of the highest quality.

Table 4.2 Summary statistics for Nanopore sequencing sequencing of *T. castaneum*

Statistic	Value
Mean read length:	15,799.30
Mean read quality:	12.8
Median read length:	11,768.00
Median read quality:	12.9
Number of reads:	5,688,065.00
Read length N50:	20,119.00
STDEV read length:	11,872.40
Total bases:	89,867,276,950.00
>Q10:	88.90%
>Q12:	65.70%
>Q15:	15.50%
# reads (>= 0 bp)	5,688,065
# reads (>= 1000 bp)	5,076,810
# reads (>= 5000 bp)	4,123,341
# reads (>= 10000 bp)	3,395,112
# reads (>= 25000 bp)	945,171
# reads (>= 50000 bp)	117,567
Total length (>= 0 bp)	89,867,276,950.00
Total length (>= 1000 bp)	86,515,280,369.00
Total length (>= 5000 bp)	83,196,677,762.00
Total length (>= 10000 bp)	73,362,556,790.00
Total length (>= 25000 bp)	35,294,400,317.00
Total length (>= 50000 bp)	7,263,554,761.00

4.3.2 Chromosome scale assembly using a hybrid assembly approach

As mentioned above, the reference genome assembly of *T. castaneum* (Tcas5.2) comprises 165.9 Mb [114]. However, after the removal of placeholders and sequencing gaps, the assembly is reduced to 136 Mb. Considering the experimentally estimated genome size of 204 Mb confirmed by the in silico genome size estimators CovEst and FindGSE, which estimated the genome size to be approximately 204–208 Mb (Supplementary Table 1), it is evident that 68 Mb (33 %) of the genome is potentially missing in the Tcas5.2 reference assembly. Additionally, FindGSE identified a repeat content of 27%, confirming experimentally determined genome's high repetitiveness. To improve the assembly, particularly in repetitive regions, our sequencing data obtained from Oxford Nanopore long-read sequencing and hybrid assembly approach was utilized. The workflow of the assembly approach used to generate the new *T. castaneum* assembly (TcasONT) is shown in Figure 4.6. The Nanopore reads were initially divided into two categories according to their length short reads (<20 kb), totaling 52.7 Gb with 258X genome coverage, and long reads (>20 kb), totaling 36.3 Gb with 178X coverage. The long reads (>20 kb) were then used for the initial assembly with Canu. This assembly resulted in 1,479 contigs with a total length of 321 Mb and an N50 of 835.5 kb (Table 4.3). The longest contig measured 16.4 Mb. The Canu assembly was approximately 117 Mb larger than the experimentally estimated genome size of 204 Mb.. Given that (peri)centromeric satDNAs, TCAST, make up 17% of the genome and poses challenges for accurate assembly, the additional 117 Mb in the Canu assembly is likely due to these repetitive TCAST arrays. To mitigate the negative impact of TCAST satDNA on the assembly, an additional filtering step was performed. Contigs lacking at least 1,000 bp of unique gene-coding sequence were removed, resulting in the successful filtering of 471 out of 1,479 total contigs representing a total of 223 Mb (Table 4.3).

Table 4.3 Summary statistics of *T. castaneum* assembly using Canu and subsequent filtering and orienting using RagTag

	All Canu contigs	*Filtered Canu contigs*	*Successfully oriented contigs*
# contigs	1479	471	244
Total length	313409266	223143008	189081091
# contigs	1479	471	244
Largest contig	16371545	16371545	16371545
Total length	321044737	223451783	189223737
GC (%)	31.21	33.01	33.37
N50	835545	3660290	5280288
N90	76926	114221	218032
auN	3757725.148	5335218.9	6249616.776
L50	45	15	11
L90	903	205	65

The 471 filtered contigs were used to create a new chromosome-scale assembly by a reference-guided approach, using the improved Tcas5.2 assembly as the reference. Two main factors drove this choice: 1) the availability of the high-quality Tcas5.2 reference genome where jumping library technology was used to generate the chromosome-scale assembly, and 2) the ability of this approach to rank input sequences based on mapping quality. Since Tcas5.2 contained 3,669 unresolved gaps totaling 11 Mb with an average gap size of 3,125 Kb, the next step was to close these gaps using TGS-GapCloser with 8.6 Gb of Canu-corrected long reads (>30 kb). With this approach, 3,607 gaps (98.3 %) were successfully closed, increasing the genome size by 10.5 Mb, mainly in the repetitive part. The gap-filled Tcas5.2 assembly was then used to determine the alignment of 471 Canu contigs in the 10 chromosomes using RagTag software. Of these, 244 contigs were unambiguously mapped to the reference genome, resulting in a new ONT-based genome assembly called TcasONT (Table 4.3). The remaining 227 contigs could not be accurately mapped and were classified as unassembled sequences (Supplementary Table 2). Further analysis revealed that approximately half of these unplaced sequences (14 Mb) were largely composed of (peri)centromeric TCAST satDNA, which accounted for more than 50% of the sequences (Supplementary Table 2). To better

understand the representation of TCAST satDNA, the TcasONT assembly was mapped with TCAST, identifying 9,446 TCAST monomers, which accounted for 3.6 Mb, or 1.7% of the genome. This means that most of the estimated 17% TCAST satDNA is still missing from the genome assembly. Finally, using the <20kb reads discarded from the assembly procedure, te RagTag oriented genome was polished 2X, producing the final TcasONT genome assembly which is used in all subsequent analysis.

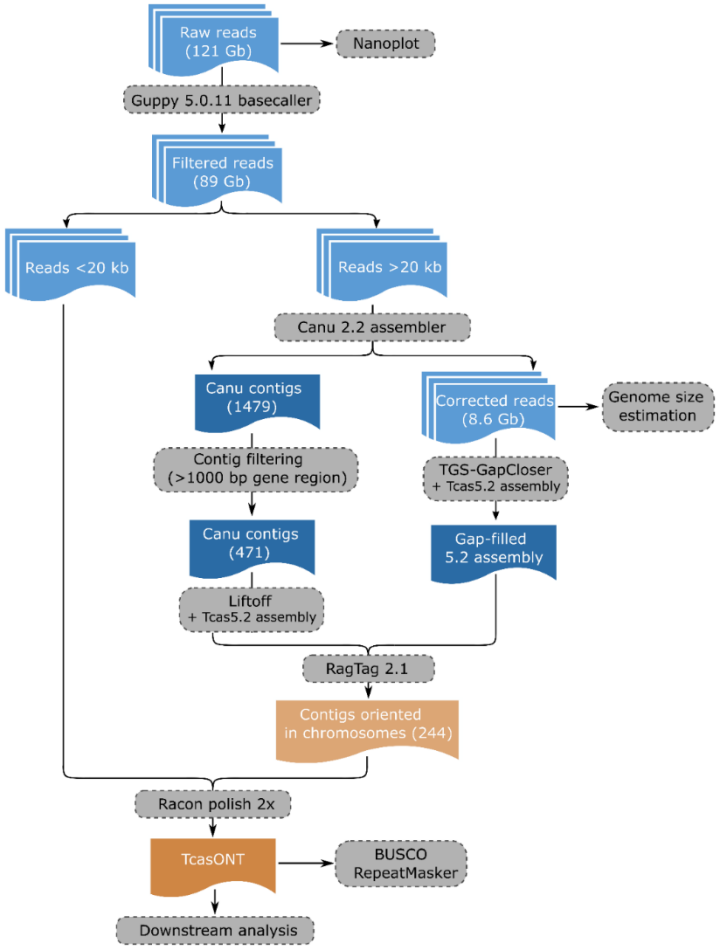


Figure 4.6 Workflow of the hybrid assembly approach used in the TcasONT assembly. The difference between all generated reads (121Gb) and reads filtered after basecalling (89Gb) is based on quality filtering performed by the Guppy basecaller.

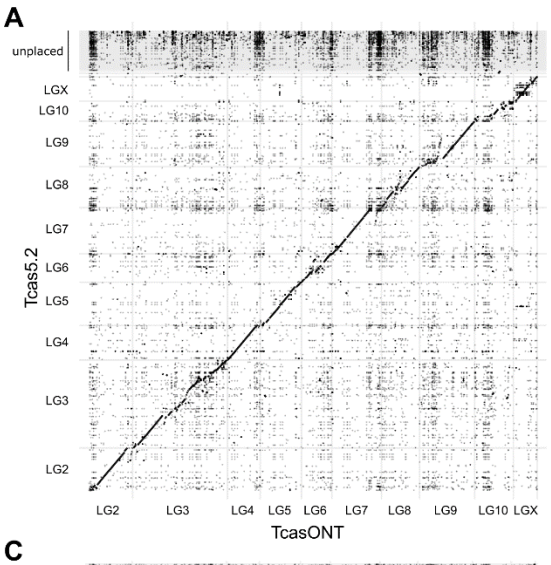
The final polished TcasONT genome assembly comprises 225.9 Mb, with 191 Mb assembled into ten chromosomes and the remainder in unassembled contigs. Compared to the Tcas5.2 reference genome, TcasONT has a 45 Mb larger total chromosome length (Table 4.4).

Table 4.4 Chromosome length comparison between the long-read improved TcasONT and the reference Tcas5.2 assembly of *T. castaneum*.

Chromosome	*Tcas5.2*	*TcasONT*	*Difference (%)*
LG10	7,222,678	16,519,013	56.27657657
LG2	15,265,516	18,604,846	17.94871078
LG3	31,381,287	40,533,075	22.57856824
LG4	12,290,766	13,994,349	12.17336369
LG5	15,459,558	17,646,621	12.39366449
LG6	10,086,398	12,970,738	22.23728519
LG7	16,482,863	21,226,280	22.34690676
LG8	14,581,690	16,306,430	10.57705457
LG9	16,184,580	23,519,639	31.18695402
LGX	8,676,460	10,258,873	15.42482298
total length (bp)	**147,631,796**	**191,579,864**	**22.93981585**

The increase in chromosome length is between 10.6% and 56.3%, demonstrating a substantial improvement in genome continuity at the chromosome level (Table 4.4). Additional dot-plot analysis of genome-to-genome mappings between TcasONT and Tcas5.2 revealed high levels of macrosynteny and collinearity across all chromosomes (Figure 4.7a), with strong sequence identity between the genomes. Additionally, 88% of the previously unplaced contig sequences in Tcas5.2 are now correctly integrated into the chromosomes of TcasONT with the remaining contigs mainly belonging to the TCAST satDNA. Gene completeness was assessed using BUSCO analysis with insect universal orthologs from the odb10 database. TcasONT identified 1,329 of 1,367 genes, corresponding to 97.2% single-copy completeness. It also included 17 duplicated genes, 13 fragmented, and only 8 missing genes. This represents a significant improvement over Tcas5.2, with 32 more complete BUSCOs detected in TcasONT (Figure 4.7b). To provide genome wide insight into the improvement of the repetitive genome content by the TcasONT genome assembly, a dot-plot analysis of the self-to-self mappings of both genome assemblies was performed (Figure 4.7c,d) using >70% sequence identity as a criterion to filter significant matches from *minimap2* output. As evident from the dot-plot the TcasONT genome has achieved a much higher level of self-

similarity, as evidenced by large dark blocks of self-similar sequences that sre repeated on all chromosomes, whereas such blocks are absent in the dot-plot of the Tcas5.2 assembly. Self-to-self mappings were processed to quantify these differences and TcasONT had 44,671 self-similar regions, while this number was only 2,202 in Tcas5.2, representing a 20-fold increase in the proportion of repetitive genome fraction. Additionally, gene annotations from Tcas5.2 were transferred to TcasONT using the Liftoff tool. Of the 14,467 genes annotated in Tcas5.2, only 48 (0.3%) remained unmapped, mainly genes with no known biological function (Supplementary Table 3).



B

Genes (N=1367)	**Tcas5.2**	**TcasONT**
Complete	1297	1329
Duplicated	6	17
Fragmented	4	13
Missing	60	8

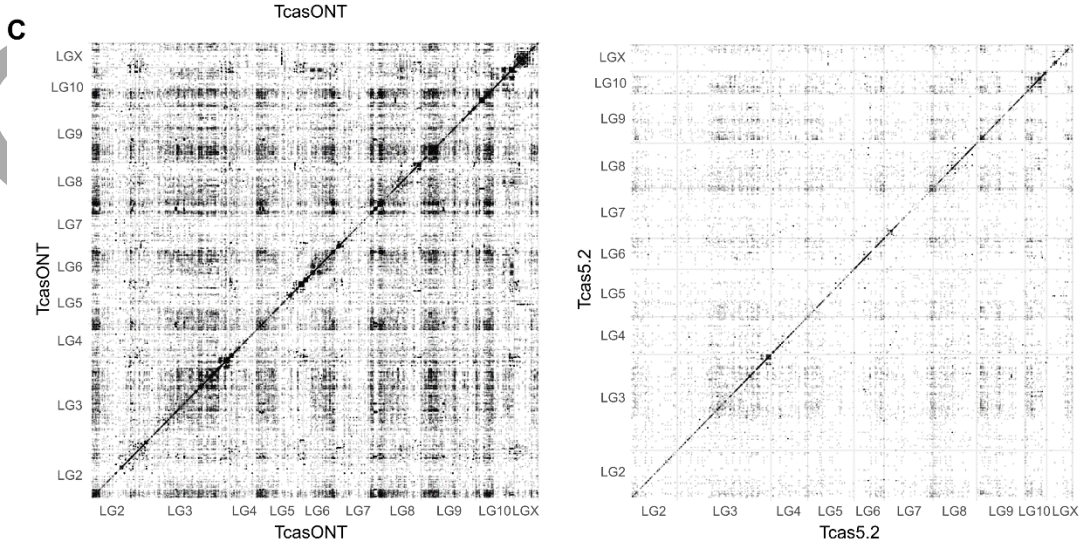


Figure 4.7 Assessment of *T. castaneum* assemblies using Dot-Plot and BUSCO Analysis. **A** Dot-plot comparison of the TcasONT and Tcas5.2 assemblies. The horizontal axis represents intervals along the TcasONT assembly, while the vertical axis corresponds to intervals along the Tcas5.2 assembly. Dots near the diagonal indicate co-linearity between the two assemblies. **B** Gene completeness analysis using BUSCO based on insect universal orthologs. Results are shown as absolute counts for complete and single-copy genes, complete and duplicated genes, fragmented genes, and missing genes. **C** Whole genome-to-genome dot-plot analysis for both TcasONT (left) and Tcas5.2 (right) assemblies. Each dot represents a region of at least 1,000 base pairs mapped to another part of the genome, with dot density reflecting the number of highly similar regions.

4.3.3 Improvement of the repetitive genome fraction

To specifically define the improvement of the repetitive genome fraction, two main classes were analysed; transposable elements (TEs) and tandem repeats (TRs) in both Tcas5.2 and TcasONT assemblies using the RepBase database and RepeatMasker (Figure 4.8). The TEs were classified into 4 main types, including DNA transposons, LINEs, LTRs and SINEs. In the comparison between the TcasONT and Tcas5.2 genome assemblies, there is a substantial increase in the number and length of various genomic TEs. TcasONT contains 92615 identified TEs, nearly double the 41437 TEs identified in Tcas5.2, reflecting a more than 2-fold increase in their number and a 3-fold increase in cumulative length (Supplementary Table 5). The largest increase can be seen in the LINE elements, with 32,237 (16.06 Mb) TEs in TcasONT compared to 4,684 (1.57 Mb) in Tcas5.2, which corresponds to an almost 10-fold increase (Figure 4.8). Similarly, LTR elements show a 5.7 fold increase in number, with TcasONT containing 14,861 elements compared to 2,593 in Tcas5.2, while their cumulative lengths increased by a factor of 3.3 in TcasONT assembly. Number and length of DNA transposons and SINE elements remained roughly the same in both assemblies, with visible increases in both categories for TcasONT (Figure 4.8a,b,Supplementary Table 5).

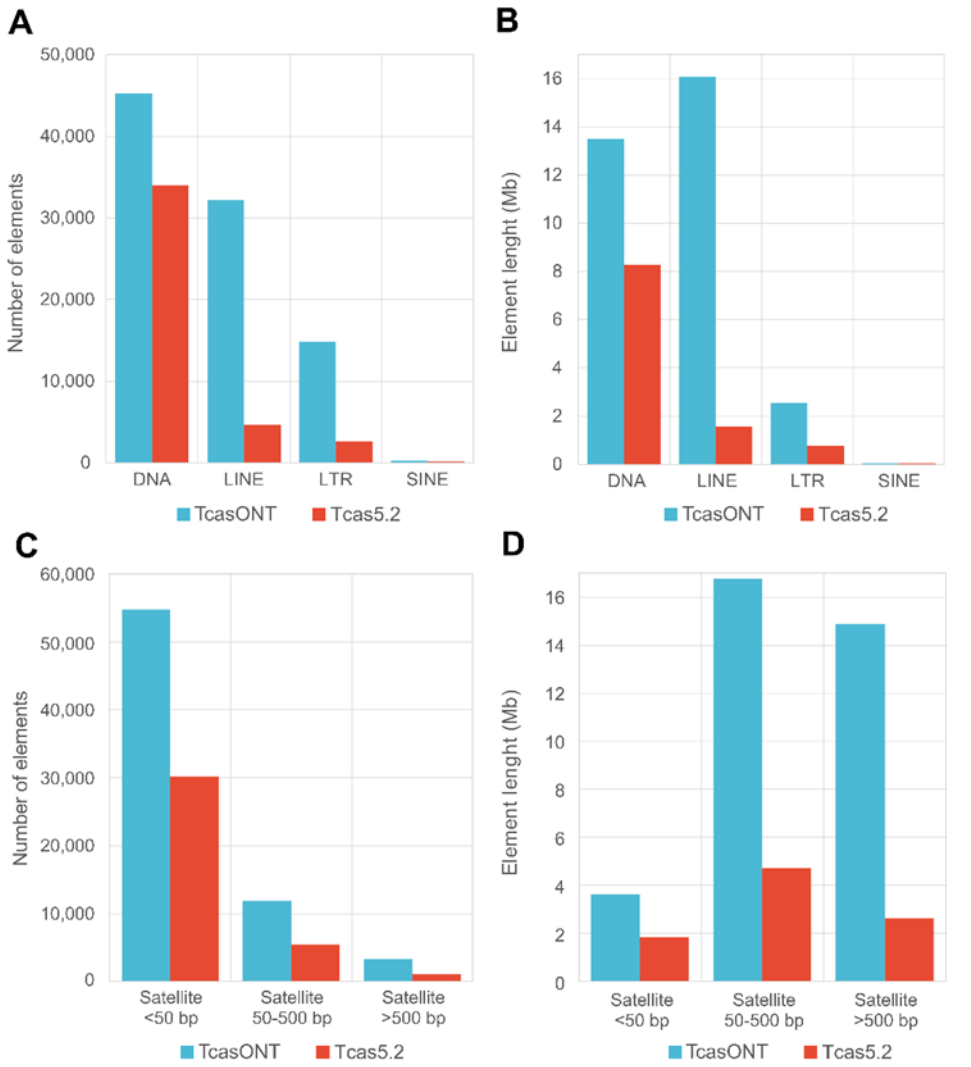


Figure 4.8 Bar charts comparing different types of genomic elements in two assemblies: TcasONT (blue) and Tcas5.2 (red). **A** The number of elements by type, including DNA transposons, LINEs, LTRs, and SINEs **B** The total cumulative length (in megabases) of the elements **C** The number of satellite DNA elements, categorized by length: satellites smaller than 50 bp, between 50 and 500 bp, and larger than 500 bp **D** Total length (in megabases) of satellite elements from C.

In addition, the other the most abundant class of repetitive DNA, tandem repeats (TRs), was roughly examined in Tcas52 and TcasONT using Tandem Repeat Finder (TRF) [100]. TRs were categorized into three groups based on monomer length: <50 bp, 50–500 bp, and >500 bp. Overall, the analysis revealed a total of 35.3 Mb of TRs in the TcasONT assembly, which represents a significant increase compared to 9.1 Mb in Tcas5.2 (Figure 4.8c.). A closer examination showed that the number of TR elements in TcasONT

had doubled in all three size classes (Figure 4.8c). The most notable contribution came from the 50–500 bp and >500 bp TRs, which include "classical" satellite DNAs, that significantly increased the genome length (31.7 Mb in TcasONT compared to 7.3 Mb in Tcas5.2) (Figure 4.8d) In summary, the 45 Mb difference in size between genome assemblies of Tcas5.2 and TcasONT is primarily due to the enrichment of repetitive regions, which account for 45.8 Mb of the increase (Supplementary Table 6). The most enriched repetitive fractions in the TcasONT assembly were transposable elements (21.5 Mb) and tandem repeats (26.2 Mb). Remarkably, (peri)centromeric TCAST satellite DNA contributes only 3.6 Mb, suggesting that the TcasONT assembly has an additional 22.6 Mb of tandem repeats outside of (peri)centromeric regions consisting of "classical" satDNAs, characterized by monomer units longer than 50 bp.

4.3.4 Enrichment of Cast1-Cast9 satDNAs in the TcastONT assembly

To detect the monomers of 9 classes of Cast1-Cast9 satDNAs in Tcas5.2 and TcastONT genome assemblies, NCBI BLAST was used. In addition, for verification of the TcatONT assembly credibility, the same analysis of Cast1-Cast9 satDNAs monomer detection was performed on randomly subsampled >20kb reads which represent 4x genome coverage and were used in the assembly process. The results are presented in Table 4.5. The TcasONT assembly shows a substantial enrichment in the abundance of Cast1-Cast9 satDNA compared to Tcas5.2, with the total genome abundance of 4.811% in TcasONT versus 1.141% in Tcas5.2, For most Cast satDNAs, the abundance (3,839%) in the randomly subsampled >20kb reads closely matches their representation in the TcasONT genome. The slight discrepancy in Cast1-Cast9 abundance between TcastONT and subsampled reads is due to the fact that 15% of the genome, which consists of pericentromeric satDNA, is missing in TcatONT. In addition, several Cast satDNAs, in particular Cast5 and Cast7, show a very strong increase in abundance in TcasONT compared to Tcas5.2 (10.97-fold and 5.04-fold, respectively). Cast5, with a monomer length of 334 bp, has the highest genome abundance in TcasONT at 1.407%, a significant increase from 0.128% in Tcas5.2. Cast1, Cast2 and Cast6 also show significant increases in genome abundance in TcasONT by 3.22-fold, 2.74-fold and 3.38-fold, respectively. Though Cast3 and Cast9 have more modest differences, most satDNAs are far more abundant in TcasONT. Comparing TcasONT to estimated genome abundancies, Cast5 shows the highest increase with 1.407% in TcasONT compared to 0.906% in the reads showing significant increase. Cast1 and Cast2 also follow this trend, with their abundances nearly doubling from the reads (0.205% and 0.295%) to TcasONT (0.431%

and 0.449%). Cast6 is similarly enriched, with 0.374% in TcasONT versus 0.264% in the reads. On the other hand, some satDNAs are more closely represented in both datasets, such as Cast4, Cast3, and Cast9, which show similar genome abundances between TcasONT and the reads.

Table 4.5 Enrichment of Cast1-Cast9 repetitive elements in TcasONT when compared to Tcas5.2 and 4x subsampled corrected reads generated by the Canu assembly algorithm

	TcasONT			Tcas5.2		Reads (corrected, 4x coverage)		
	Monomer Length	Monomer number	Genome abundance (%)	Monomer number	Genome abundance (%)	Monomer number	Abundance in reads (%)	**TcastONT/Tcas5.2 (fold increase)**
Cast1	172	5258	0.431	1149	0.134	10080	0.205	**3.217036**
Cast2	172	4997	0.449	1407	0.164	14481	0.295	**2.736779**
Cast3	227	1292	0.153	898	0.138	3868	0.104	**1.108691**
Cast4	179	2129	0.199	814	0.099	9222	0.196	**2.015467**
Cast5	334	8073	1.407	567	0.128	22903	0.906	**10.97176**
Cast6	180	3980	0.374	908	0.111	12372	0.264	**3.377705**
Cast7	121	1967	0.124	301	0.025	49995	0.717	**5.035729**
Cast8	169	534	0.047	248	0.028	1485	0.030	**1.659258**
Cast9	350	496	0.091	377	0.089	1501	0.062	**1.013829**
TOTAL			**4.811**		**1.141**		**3.839**	

Given the significant improvements in genome assembly, particularly in the representation of satDNAs regions, the new TcasONT assembly provides an exceptional platform for in-depth analysis of both structure location and genomic organization of the Cast1-Cast9 satDNA sequences, located outside of the (peri)centromeres.

4.3.5 Identification of Cast1-Cast9 satDNA arrays in the TcasONT assembly

Due to the well-documented variability of monomer sequences within a satellite DNA (satDNA) family, it was crucial to establish parameters for sequence similarity and sequence coverage to ensure the detection of the vast majority of Cast1-Cast9 satDNA arrays in the TcasONT assembly. Additionally, these parameters were essential for assessing sequence variation and variability across the genome. To achieve this, a detailed BLAST search was performed on both the raw Nanopore sequencing data and the TcasONT

assembly using the consensus sequences of the Cast1-Cast9 monomers. Two key parameters were measured for all sequences found: sequence coverage and similarity. The results were visualized through density plots, where color intensity indicated the number of monomers for each Cast satDNA (Figure 4.9). Most of the Cast satDNAs showed a high intensity, or aggregation, of monomers in the region corresponding to a sequence coverage more than 75% and sequence identity more than 70%.

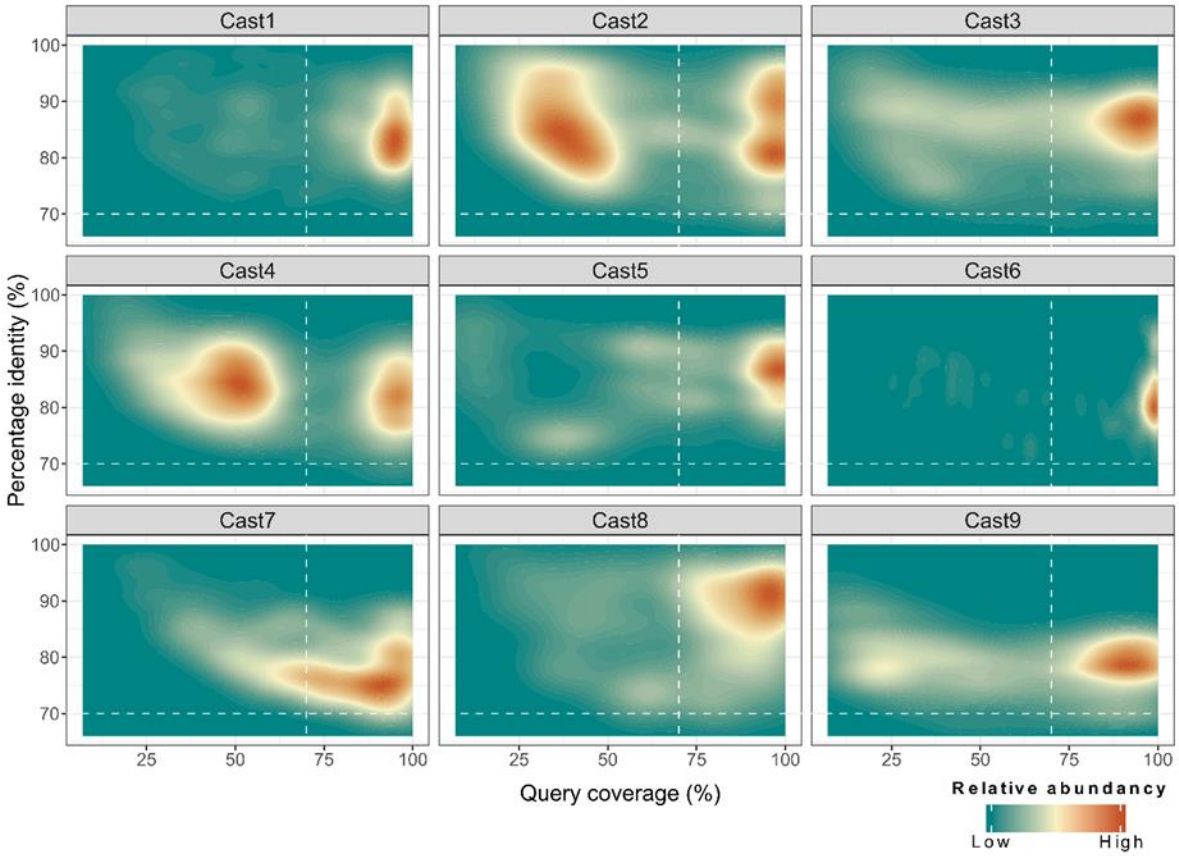


Figure 4.9 The density plot of percentage identitiy and query coverage for Cast monomers of nine cast satDNAs (Cast1-Cast9) identified through a BLAST search of the TcasONT assembly. The relative abundance of the monomers is represented by a color gradient ranging from green to red. Dashed white lines illustrate the established parameters for satDNA detection in subsequent analyses to capture as many high quality monomers.

Notably, Cast2 and Cast4 showed a distinct pattern with two areas of high density, likely due to partial sequence similarity between these satDNAs (Figure 4.10). Despite this overlap, they were classified as separate satDNAs because their consensus sequences differ by 50% of their total sequence length.

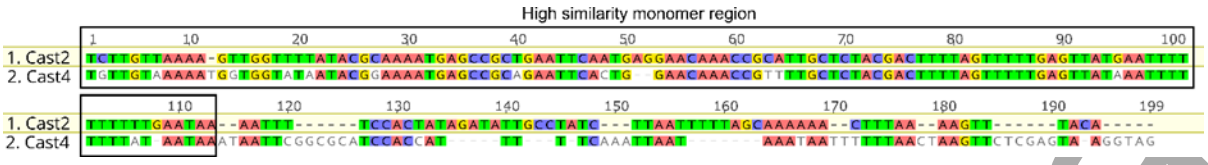


Figure 4.10 Pairwise alignment of Cast2 and Cast4 sequence, with the gray box denoting the region of high similarity between the satDNA consensuses.

Based on the density plots for all Cast satDNAs, a threshold of >70% sequence coverage and >70% identity was established to effectively map the majority of monomers from all nine Cast satellite DNA families (Cast1-Cast9) onto the TcasONT assembly. This approach enabled a comprehensive capture of sequence variation while ensuring that the vast majority of satDNA monomers were successfully identified in the genome.

Since satDNAs form large arrays that can span several kilobases, the next step was to determine the properties of the arrays of individual Cast satDNAs to better understand their organization. To investigate the organization of the Cast1-Cast9 arrays, we analyzed the distances between the monomers within each Cast array (Figure 4.11). In particular, we wanted to determine whether the arrays were organized in continuous tandem arrays (consisting only monomers of specific Cast satDNA) or exhibited a mixed tandem organization (with different sequences in the arrays). The results showed that most satDNA arrays had a typical satDNA organization, with a continuous arrangement of monomeric variants, as evidenced by the lack of sharp increases in the curve as in Cast1, Cast3 and Cast9. This pattern suggests that these arrays are primarily composed of monomers organized in tandem without interruptions. However, Cast2, Cast5, and Cast7 exhibited disrupted tandem continuity, as evidenced by a sharp increase in the probability of finding monomer at certain distances. The most prominent increase was observed in Cast2, suggesting the presence of a different sequence within the Cast2 monomers, warranting further investigation.

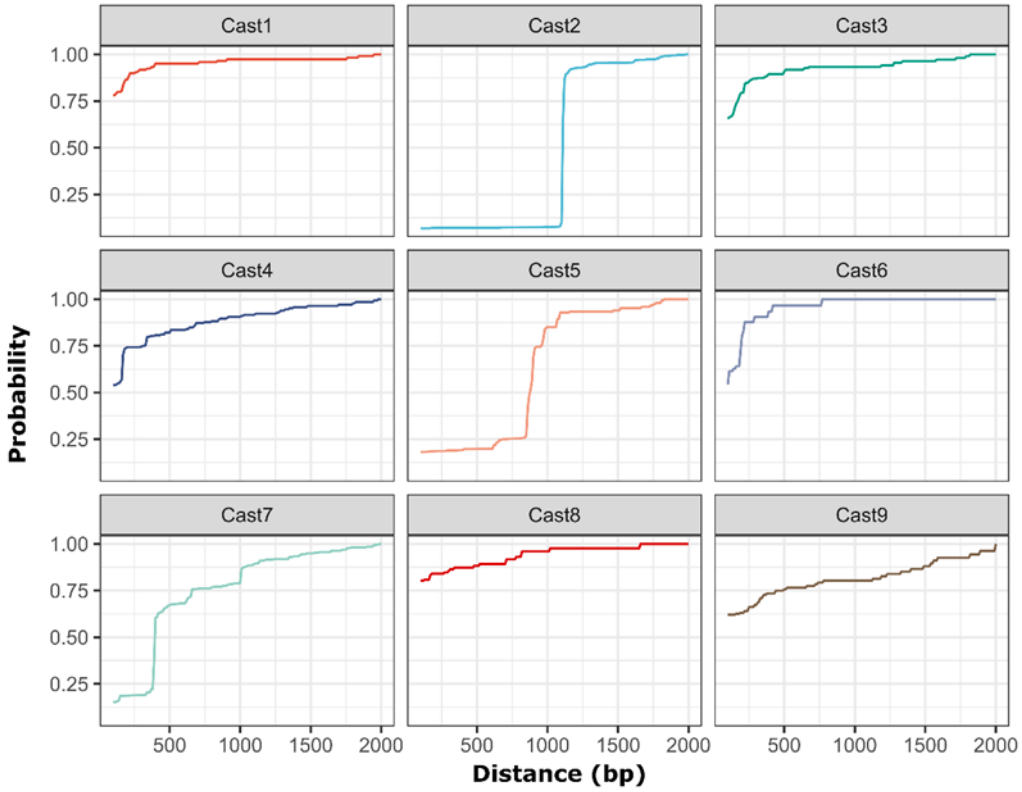


Figure 4.11 The analysis of the monomer organization for Cast1-Cast9 reveals the probability of finding another Cast monomer at a specified distance from an existing one. Notable spikes in Cast2, Cast5, and Cast7 indicate a distinct pattern of satDNA organization that incorporates additional sequences. In contrast, the other satDNAs show no sharp increases, suggesting a tandem organization composed solely of satellite monomers, as evidenced by the gradual rises in the graph and their high starting points. For instance, Cast3 demonstrates that 60% of all monomers in the genome are located immediately after another monomer. The steep increase in Cast2 is attributed to the formation of a new repeat unit, Cast5 is frequently intercalated with R66-like elements, while Cast7 occasionally exhibits a improper tandem organization with TCAST.

Detailed analysis revealed that, in addition to the homogeneous Cast2 arrays described previously, almost 90% of Cast2 monomers are predominantly found as part of a new, longer repeat unit approximately 1270 bp in length (Figure 4.11, Figure 4.12a). This new repeat family was named Cast2', and in subsequent analyses, these two forms of Cast2 arrays were analysed separately. For Cast5, further investigation showed that the observed disruption in tandem organization of arrays was due to the insertion of previously described R66-like sequences, which were interspersed within the continuous Cast5 arrays (Figure 4.12c). Similarly, analyses of Cast7 arrays revealed a mixed organization, with Cast7 monomers frequently associated with (peri)centromeric TCAST satDNA. However, this association exhibited low sequence length and similarity (Figure 4.12d), suggesting a complex structural arrangement for these

particular arrays. Considering monomer length and mixed array organization of Cast2, Cast5 and Cast7, the best window length that ensures detection of the maximum number of arrays was evaluated for each Cast satDNA.

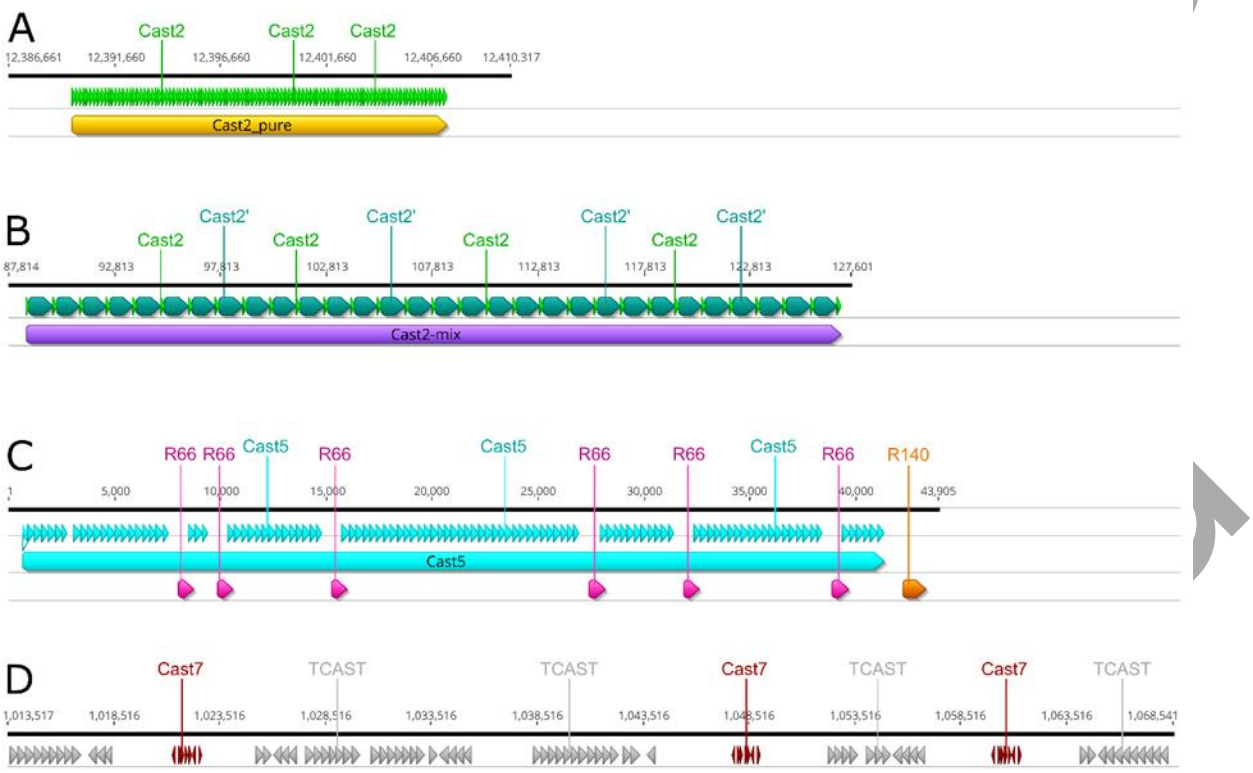


Figure 4.12 A Cast2 arrays organized in a classical tandem head-to-tail arrangement **B** Formation of the new satellite DNA family, named Cast2', composed of Cast2 monomers (light green) and intermediate sequences (dark green) **C** Organization of Cast5 arrays with adjacent regions in which R66-like and R140-like transposable elements are present on 2/3 of arrays **D** Cast7 arrays lacking true tandem organization associated with (peri)centromeric TCAST satellite DNA.

The parameters determined for valid array detection in the genome assembly allow comparative studies of reference Tcas5.2 and TcastONT assemblies for Cast satDNAs. Among other things, the analysis showed that a minimum of three consecutive monomer units should be a criterion for correct array characterization for each Cast satDNA. Comparative analysis of the Cast1-Cast9 satDNA arrays, taking into account the number of arrays, the mean value of the arrays, their total length and their abundance, reveals several notable differences between the Tcas5.2 and TcasONT genome assemblies (Figure 4.13).

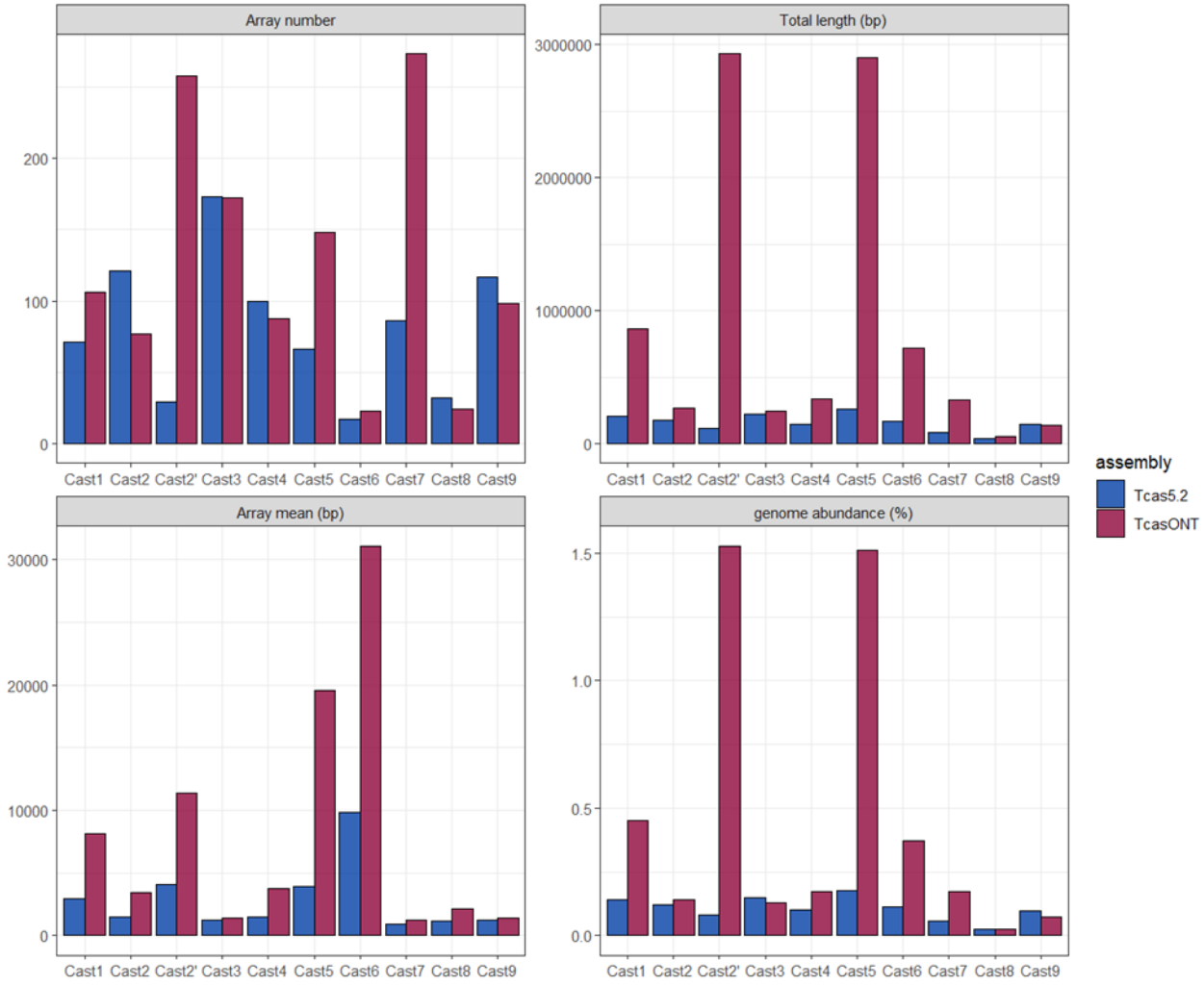


Figure 4.13 Summary statistics of various Cast1-Cast9 properites between TcasONT (red) and Tcas5.2 assemblies (blue). The statistics shown are array number (top left), array total length (top right), array mean length (bottom left) and total genome abundance comprised in the arrays (bottom right)

The number of arrays shows a significant increase in TcasONT, especially for Cast2', Cast5, and Cast7, indicating that these sequences are more comprehensively represented in the new assembly (Figure 4.13). In particular the newly defined Cast2'showed the largest increase, with the number of arrays increasing by more than 9-fold. This indicates that many arrays were either missing or fragmented in the previous Tcas5.2 genome assembly. In addition, the total arrays length is considerably larger in TcasONT, with Cast1, Cast2', Cast5, and Cast6 showing the largest increases with 24-fold for Cast2', thus revealing a much better representation of these satDNA regions in the new genome assembly (Figure 4.13). This increase in total array length suggests that the satDNA regions were more thoroughly captured and

assembled in new Nanopore generated TcastONT assembly, especially for those specific satDNA families. The array mean lengths have also increased dramatically in the new assembly, particularly for Cast6 and Cast5, where the mean array lengths have increased 3.16-fold and 5-fold respectively presenting a higher degree of contiguity in the Cast satDNA regions, implying that the arrays in TcasONT have fewer interruptions and are assembled more completely. Finally, the genome abundance plot confirms the results from monomer analysis, that the proportion of the genome occupied by these satDNA arrays has increased significantly in TcasONT, with Cast1, Cast2', and Cast5 now occupying significantly larger portions of the genome. Overall, the enrichment of both the array number and their cumulative length shows a substantial improvement in coverage the Cast satDNAs in the new assembly, demonstrating that the TcasONT assembly has much more comprehensive in its representation compared to Tcas5.2.

The final check that the TcastONT assembly represents an excellent platform for the overall analysis of Cast1-Cast9 satDNAs was the comparison of TcastONT assembly and raw data. Given the challenge of accurately assembling satDNA sequences, we next explored whether the array lengths and organization of Cast1-Cast9 satDNAs in the TcasONT assembly truly reflect the actual structure of the genomic loci containing these repeats. A key concern is that the repetitive nature of satDNAs can often lead to assembly collapse, resulting in an underestimation of the number of monomer units (or array length) in a genome assembly compared to real abundance in the genome. Since raw reads provide a more accurate representation of what is actually present in the genome, without being subject of an assembly process, we conducted a comparative analysis of the Cast1-Cast9 sequences between the individual raw reads and the newly generated genome assembly. In this analysis, we used previously established optimal parameters for detecting arrays (Figure 4.14).

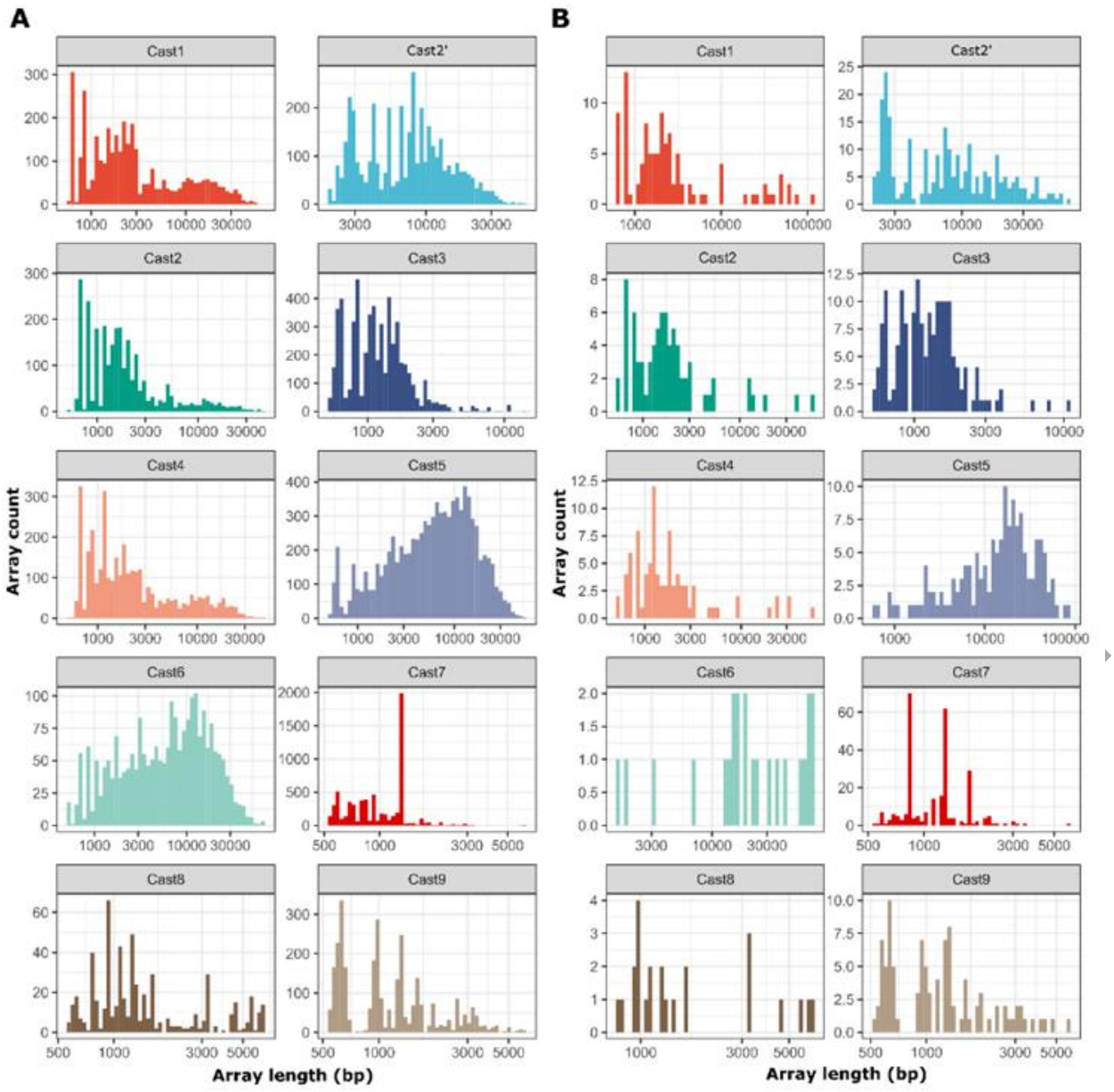


Figure 4.14 A Distribution of array lengths in corrected Nanopore reads for Cast1-Cast9 satDNAs B Distribution of array lengths in TcasONT for Cast1-Cast9 satDNAs. The length of the arrays is log10-scaled.

The results of comparison revealed a significant level of similarity in array patterns between the datasets of individual raw reads and the TcasONT assembly for most of the Cast satDNAs. For most Cast satDNAs, the array length distributions is comparable, with peaks typically found around 1,000 to 5,000 bp, which is especially true for satDNA families with shorter mean arrays. This indicates that the core features of the

arrays, especially those of shorter lengths, are preserved. However, longer arrays (over 20,000 bp) are underrepresented in the assembly across longest Cast sequences. This is particularly evident for Cast6, where the assembly struggles to capture the full range of array lengths seen in the raw data. Despite these differences, the shared general patterns between the two datasets reflect a level of consistency in assembling shorter repetitive arrays, even though the possibility that the assembly loses detail for longer repeats persists.

4.4 Cast1-Cast9 satDNAs chromosome distribution and genomic environment

Once the Cast satDNA arrays had been precisely determined, the next step was to examine their genomic distribution and regions surrounding them. The results of the chromosomal distribution are shown as a heat map in Figure 4.15a. It shows the scaled frequency of Cast1 to Cast9 satDNAs on different chromosomes, with values normalized relative to the chromosome with the most abundance for each satDNA family and also normalized the the total length of the chromosomes and gray color indicating that a satDNA familiy is missing from the chromosome. Cast1, Cast2 as well as Cast8 and Cast9 show low to moderate levels of abundance across most chromosomes, with high peaks for LG6. Cast5 and Cast2' show more uniform distributions, with Cast2' occurring with similar frequency on all chromosomes. Cast3, and Cast4 appear to have a more variable distributions being almost completely absent in LG5 and LG7 and completely absent on LGX (Cast4). In addition, Cast6 exhibits highly variable pattern of chromosome distribution being strongly representated on LG3 and LG9 but completely absent on LG4, LG7, LG10 and LGX. This heatmap reveals the heterogeneous distribution of the Cast families, suggesting that certain Cast families, such as Cast2' and Cast5 occur on multiple chromosomes, while some other families tend to localize in large amounts on one chromosomal subset while being present in low abundance on others. This points to the evolutionary drive of certain satDNA families to either spread throughout the genome or to amplify on specific genomic loci.

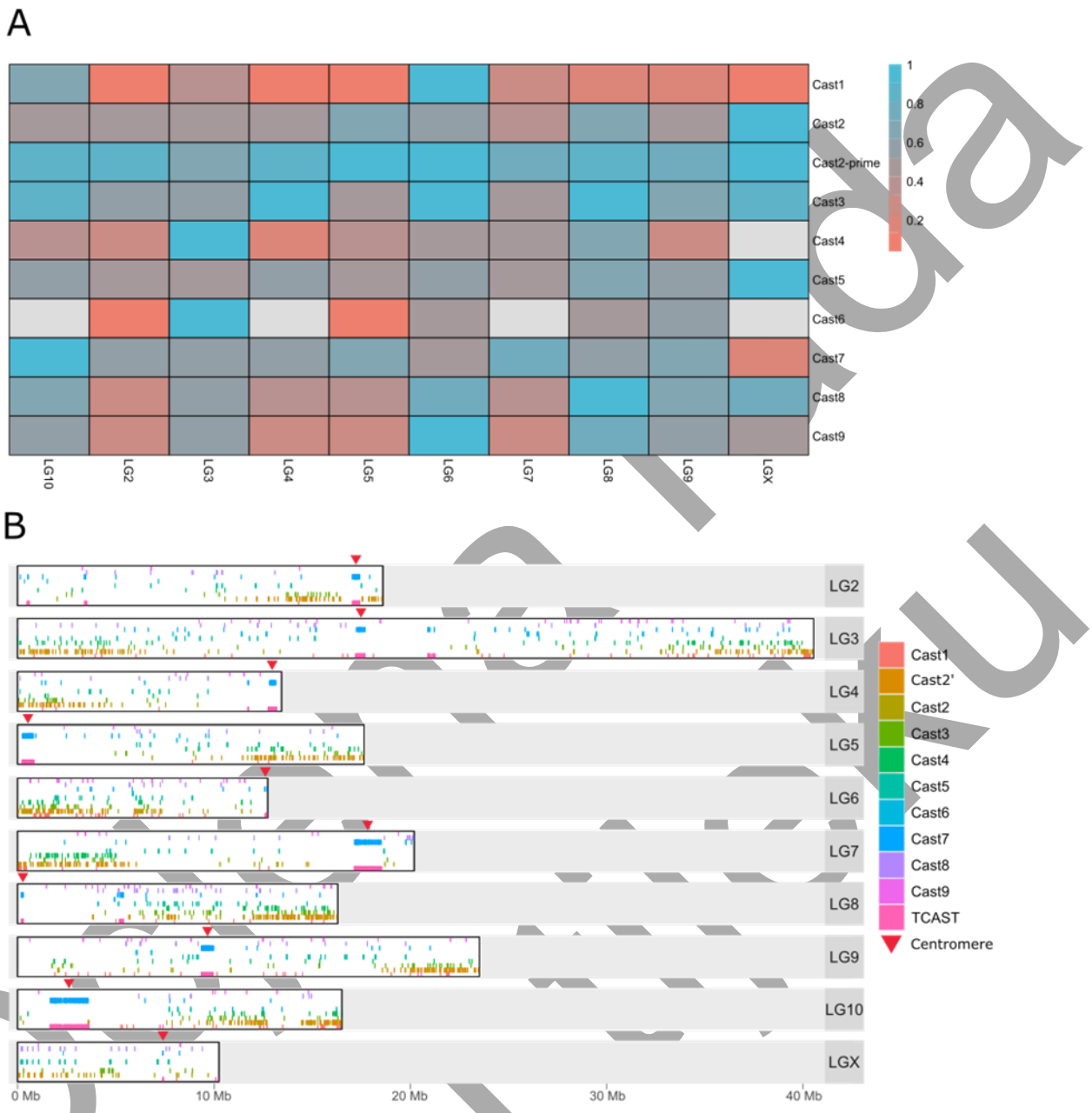


Figure 4.15 A Heatmap of Cast1-Cast9 presence and total content on chromosomes in TcasONT assembly. Each satDNA family is scaled to both chromosome length and its highest content per megabase of chromosome length. B Locations of Cast1-Cast9 satDNA arrays on chromosomal arms in the TcasONT assembly. The location of the centromere is denoted by the red arrow.

This was further confirmed by examining the distribution of Cast1-Cast9 along the length of the chromosomes (Figure 4.15b). Given the known presence of large blocks of (peri)centromeric heterochromatin in *T. castaneum*, the presumtive locations of the (peri)centromeres on each chromosome were marked. A detailed analysis of the distribution revealed that Cast satDNAs are broadly distributed along the chromosomes, with no clear tendency to cluster around the (peri)centromeric regions where satDNA is frequently located. The only exception is Cast7, which is preferentially located near the (peri)centromeric regions marked by the satDNA TCAST. This is due to the complex structure it forms with TCAST main satDNA (Figure 4.12). Additionally, some Cast satDNAs were found to have a slight tendency to cluster in the distal regions of the chromosomes, or in specific genomic regions generally localizing away from the (peri)centromere.

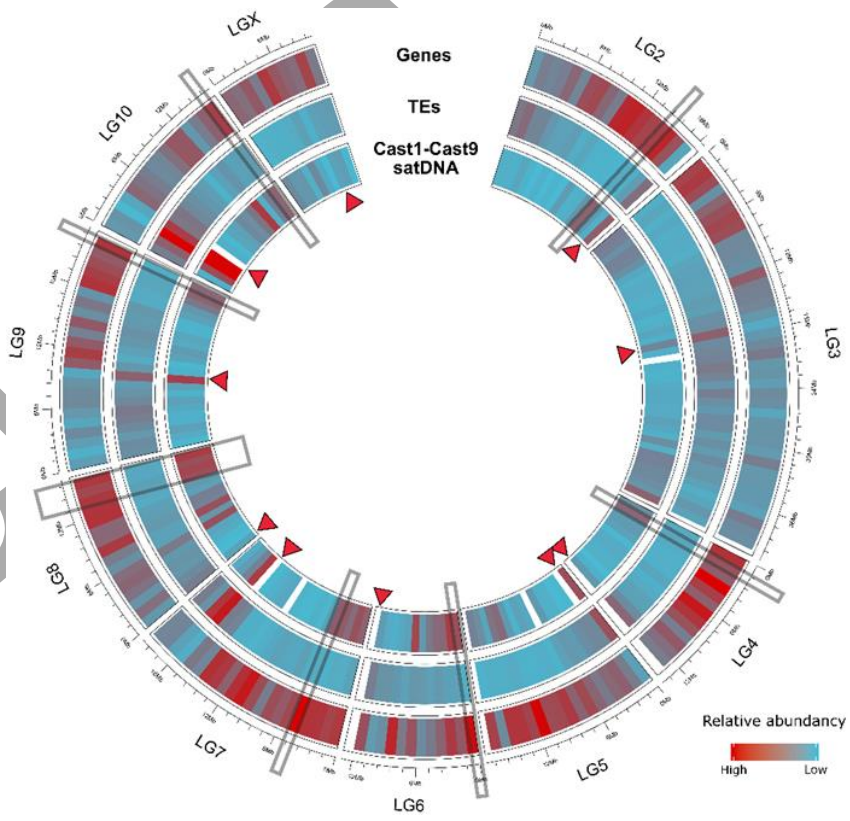


Figure 4.16 Circular plot of the genomic distribution of genes (outer) transposable elements (middle) and Cast1-Cast9 satDNAs in TcasONT assembly. The genome is divided in equal 500kb bins and the total number of each element is counted per each window. High abundancy is marked with red, while low abundancy with blue. The location of the centromere is marked with the red arrow.

The chromosome distribution of Cast1-Cast9 satDNAs compared to genes and transposable elements was further analyzed, and the results are shown in a circular plot (Figure 4.16). The analysis was performed using 500 kb bins, which can contain multiple genes and different Cast families, underscoring the potential complexity and diversity within these regions. The results show that the Cast1-Cast9 satDNAs are frequently found in gene-rich regions, and often overlap with genes, while they have little to no overlap with TEs. This lack of co-occurrence between Cast elements and TEs serves as an example of their genomic independance, with the only exception being the (peri)centromeric regions, where Cast7 satDNA coincides with TEs. The highlighted regions in the plot emphasize the clear separation between Cast satDNAs and TEs, and reinforce the idea that Cast elements and TEs occupy largely non-overlapping genomic regions (Figure 4.16).

Since the circular plots of the entire genome suggest that Cast1-Cast9 elements are embedded in gene-rich regions, the next step was to determine the precise locations of these regions and whether these satDNAs are indeed distinct from transposable elements (TEs). Since circular plots provide a rough estimate of gene and TE amount within large 500kb windows, a more detailed analysis of the flanking regions was conducted. Specifically, genes and TEs within 50 kb flanking regions (±) of Cast1-Cast9 arrays were counted, and the gene and TE content across the genome assembly was used to assess the relative density near the Cast arrays (Figure 4.17a, b).

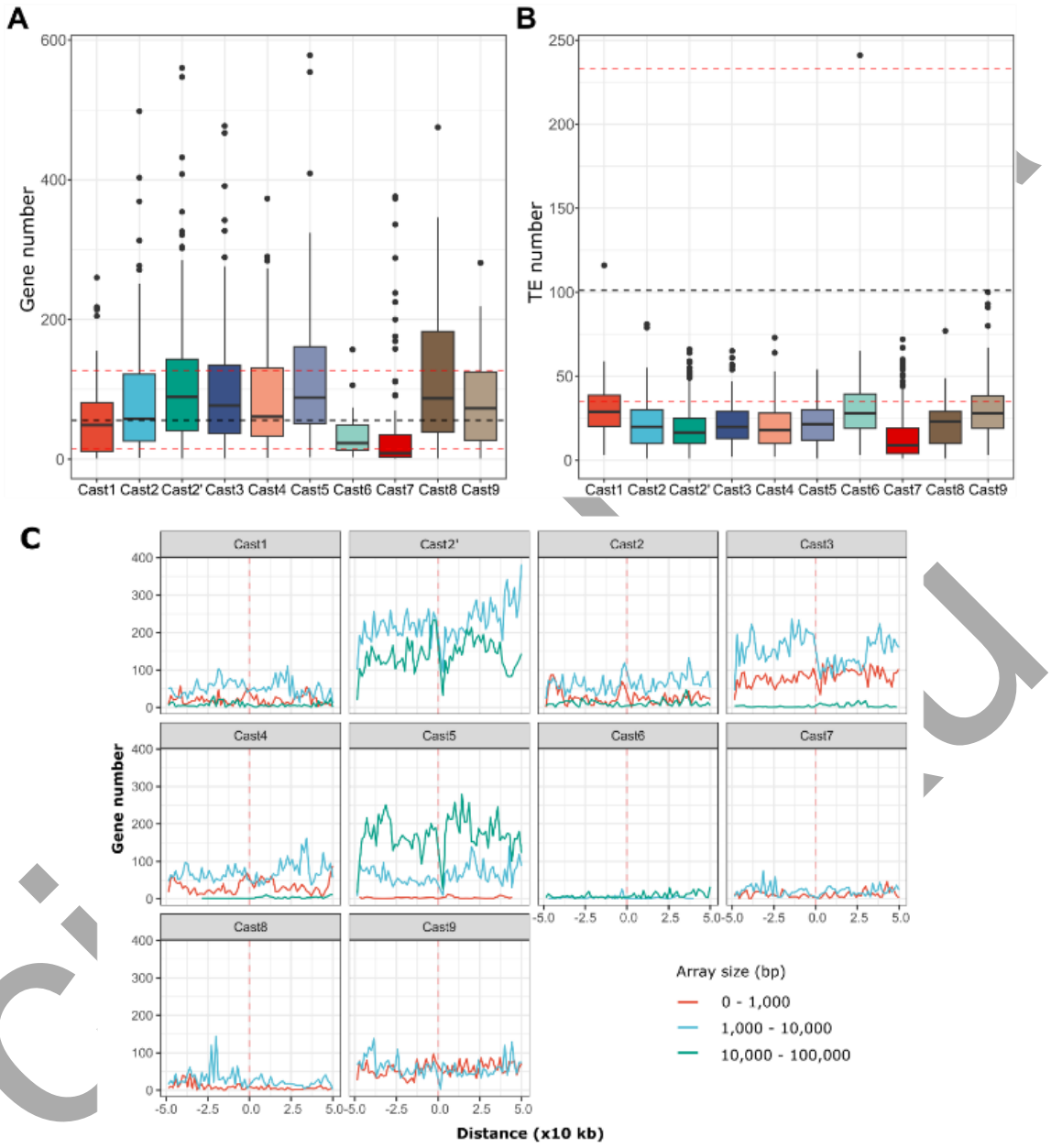


Figure 4.17 A Boxplot of total gene (exon) content in +/- 50kb flanking regions of Cast1-Cast9 satDNA arrays. The genome median exon content in 100kb windows is denoted by the dashed black line while the genome 1Q and 3Q content is denoted by the dashed red line. **B** Boxplot of total transposable element content in +/- 50kb flanking regions of Cast1-Cast9 satDNA arrays. The genome median transposable element content in 100kb windows is denoted by the dashed black line while the genome 1Q and 3Q content is denoted by the dashed red line. **C** Line plot of exon content in 1kb sliding windows of +/- 50kb flanking regions of Cast1-Cast9 satDNA arrays. Arrays are divided based on their length to short (<1kb), intermediate (1-10kb) and long (>10kb). The location of arrays is denoted on the graph by the dashed red line and total exon count in the window on the Y axis.

The scaled values based on the total number of arrays show that most Cast satDNAs have a higher median gene content than the median of the genome, with the exception of Cast1, Cast6, and Cast7. Furthermore, Cast2', Cast3, Cast5, and Cast8 are flanked by a significantly larger number of genes (Kolmogorov-Smirnov test, $p < 0.01$) than the genome average (Supplementary Table 7). These satDNAs also exhibit distributions above the third quartile of the genome, indicating that many arrays are embedded in highly gene-rich regions. Cast7 is the only satDNA with a significantly lower number of adjacent genes, corresponding to its intermingled arrangement with (peri)centromeric TCAST satDNA (Figure 4.12b). In contrast, Cast satDNAs are surrounded by significantly fewer TEs than the genome as a whole (Kolmogorov-Smirnov test, $p < 0.01$) (Supplementary Table 7), further highlighting the separation of these arrays from the dynamics of TEs. To further explore the precise location of Cast1-Cast9 satDNAs, a rolling window analysis was conducted (Figure 4.17c), counting the number of exons within each window and categorizing satDNA arrays into three size classes: short (<1 kb), intermediate (1–10 kb), and long (>10 kb). Interestingly, there was no discernible trend towards shorter arrays. In fact, in Cast5, Cast2', and Cast3 most arrays of intermediate size are deeply embedded in gene-rich regions and surrounded by hundreds of exons. Even less abundant satDNA families, such as Cast4 and Cast9, show a substantial number of exons in their vicinity, suggesting that gene structure allows of large satDNA arrays to coexist in the same genomic environment.

4.5 Mechanisms of propagation and evolution of Cast satDNAs

To examine the junction regions of Cast arrays accurately, it is essential to define their boundaries precisely. Given that the monomers at the edges of the array tend to have higher variability due to reduced recombination efficiency [108], k-mer similarity-based approach was implemented to overcome this challenge (see Methods). This new method significantly improves the detection and merging of arrays, as can be seen in Figure 4.18. The red shaded areas represent broken monomer fractions, that cannot be detectable by the conventional BLAST search but can interfere with the micro/microhomology search. As a result, array edges were successfully redefined to 4bp accuracy, and some arrays that were previously considered separate were now merged. Using these edge-refined arrays, both closely homologous regions (20 bp) and larger genomic segments (2 kb) were extracted and aligned using MAFFT for more detailed analysis.

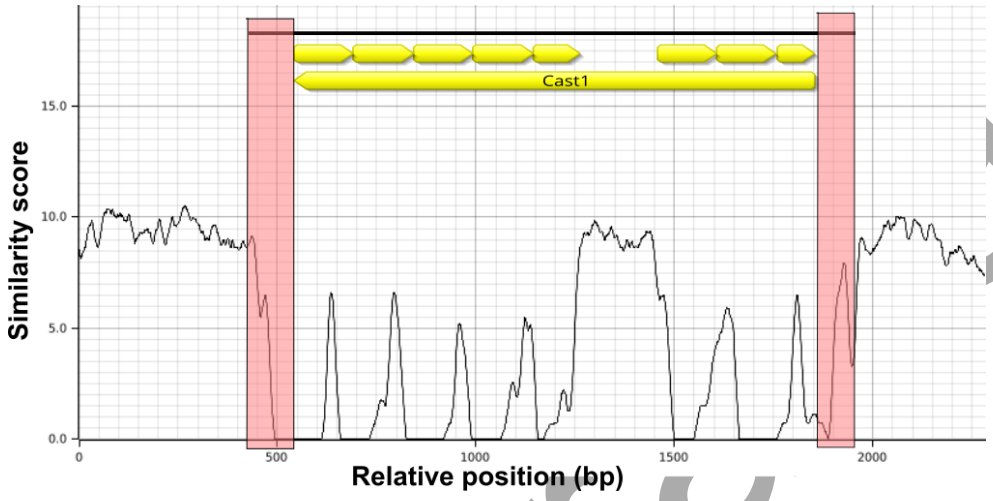


Figure 4.18 Example of improved annotation by using the newly developed k-mer similarity counting strategy. The location of previous arrays is presented in yellow; the output of the program is visualized by the black line. The improvement in annotation is presented as the red shaded area on array borders.

The macrohomology junction region analysis (Figure 4.19a) revealed that, of the ten Cast satDNAs analyzed, only Cast5 and Cast7 exhibit a consistent similarity in their surrounding regions for most arrays. For Cast5 arrays, two dominant regions with high sequence similarity were identified. One side of arrays often contains an R66-like sequence, which can also be scattered within Cast5 arrays (see Figure 4.12c), while the other side predominantly contains an R140-like sequence. Of the 150 Cast5 arrays analyzed, two-thirds had R140-like sequences at their ends, and one-third had R66-like sequences. Similarly, most Cast7 arrays were found to be flanked by (peri)centromeric TCAST (Figure 4.12d). In contrast, the remaining Cast satDNAs showed only partial similarities in their surrounding regions, affecting a smaller subset of arrays. For example, a subset of the Cast1 arrays, all from chromosome LG7, had the same transposon-like sequence at their array ends (Supplementary Figure 5). Additionally, microhomology analysis of 20 bp sequence motifs near array boundaries revealed that Cast1, Cast3, and Cast9 have poly A/poly T tracts in these regions, while other Cast satDNAs lacked a common motif (Figure 4.19b)

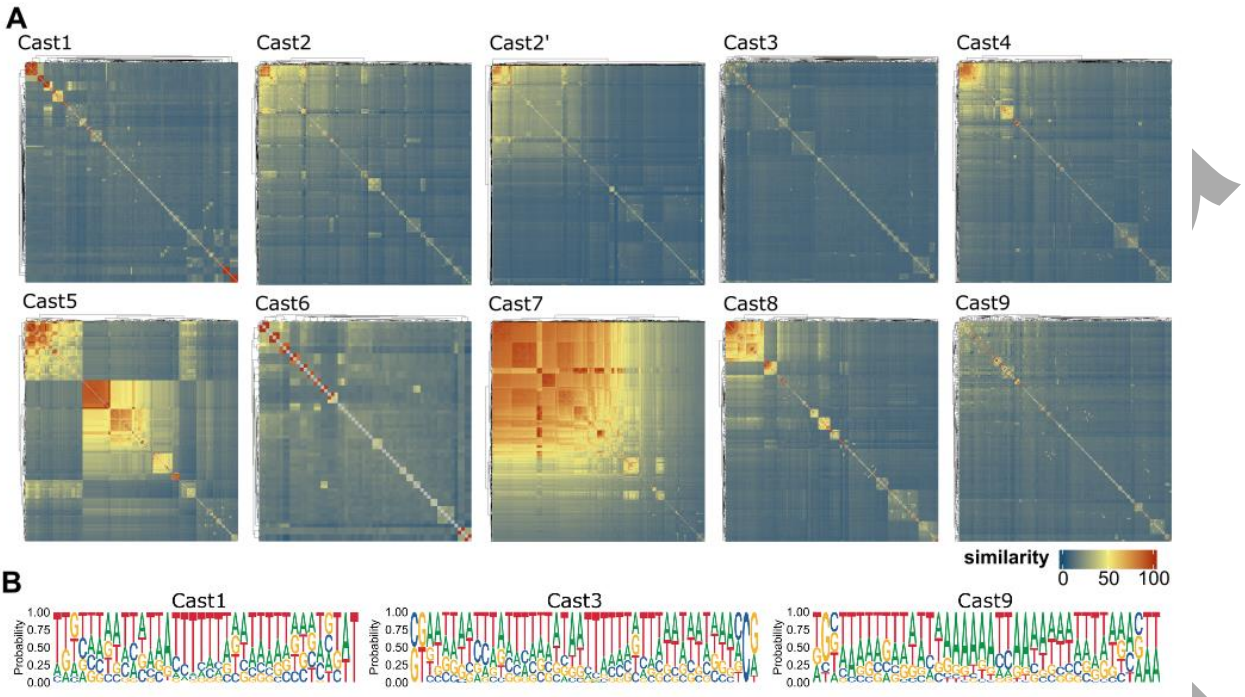


Figure 4.19 A Heatmap clustering and visualization of +/- 2kb flanking regions of Cast1-Cast9 satDNA arrays and their inter-similarity. High similarity flanking regions are shown in red while low similarity regions are shown in blue. **B** SeqLogo visualization of discovered motifs MAFFT alignment in microhomology junction regions of 3 satDNA arrays, Cast1, Cast3 and Cast9.

Following the junction region analysis, it became clear that the forces driving satDNA evolution likely originate from the satDNAs themselves, suggesting that they are components of the genome that are self-propagating. Consequently, analyses comparing the mutual variability of monomers and arrays, together with their chromosomal positions, can provide valuable insights into the genomic dynamics of Cast satDNAs. Given the large dataset and relatively low overall variability within monomer families (Figure 4.9), traditional phylogenetic approaches, which would require analyzing of large numbers of similar sequences, were deemed insufficiently sensitive and too time-consuming to fully capture the trends in Cast satDNA genome dynamics (Supplementary Figure 2).

Therefore, principal component analysis (PCA) and UMAP embeddings were applied to genetic distance matrices generated from monomer alignments (Figure 4.20). A genome-wide database of Cast monomers, annotated with their chromosomal positions, was created. The PCA results revealed a scattered distribution pattern for most Cast satDNAs, especially for Cast1, Cast2, Cast2', Cast3, and Cast4,

where regions with a high density of monomers from different chromosomes were observed. An exception was Cast6 and Cast5, which showed clustering of monomers from the same chromosome, reflecting their long, homogenized arrays which tend to have high intra-array similarity. To validate the accuracy of the PCA embeddings, graphs displaying the percentage of explained variance were included (Supplementary Figure 3), confirming that dimensionality-reduction techniques effectively captured the variation within the monomer alignments with up to 94% of the observed variance in Cast4 being explained by the first principal component alone.

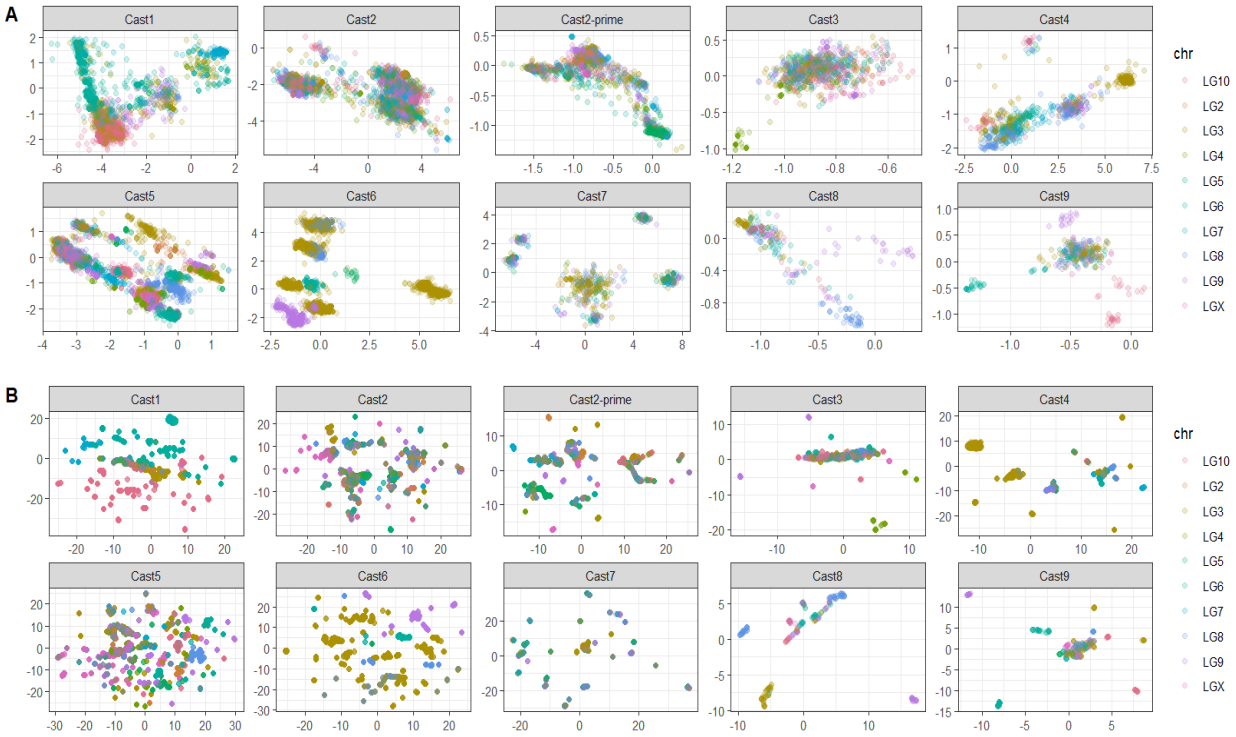


Figure 4.20 A First 2 principal component PCA of the distance matrix generated by all extracted monomer alignment of Cast1-Cast9 satDNAs colored based on their chromosome of origin. **B** UMAP embeddings of the distance matrix generated by all extracted monomer alignment of Cast1-Cast9 satDNAs colored based on their chromosome of origin.

To disclose relationship between Cast arrays, database of arrays with their corresponding chromosomal annotations was created and comparative analysis of arrays was performed for in order to examine the sequence variability of arrays within each Cast satDNA family. The relationships between arrays based on

sequence similarity are visualized as graph networks (Figure 4.21). The Cast6 and Cast7 were excluded from this analyses due to their small number of arrays. In these graphs, the distance and interconnectivity between dots correlate with the sequence similarity between arrays, where closer dots indicate higher similarity between monomers of different arrays. It is assumed that the relationship between arrays reflects their genomic spread. Three patterns of satDNA evolution can be deduced from the graphs. First group is characterized by one dominant cluster of relatively closely related arrays from nearly all chromosomes, the most prominent example of this pattern is Cast3 network, but also can be observed in Cast8, and Cast9 (Supplementary Figure 4e,f). The second group of Cast satDNAs show several distinct array clusters of intensive interchromosomal expansion, where related arrays are spread across different chromosomes, suggesting that such expansion events occurred several times throughout Cast2 evolution. Notably, only one cluster shows intrachromosomal expansion, while the others indicate extensive interchromosomal exchange. A similar pattern, with several distant clusters containing related sequences from different chromosomes, is also observed in Cast4, Cast2' and Cast5 (Supplementary Figure 4a,g and c). Finally, the last model of high divergence and homogenization for certain arrays is present on Cast1 network is characterized by greater distance between clusters, with some array sets completely separated due to sequence divergence. Cast1 also contains two distinct subgroups of sequences completely separated from the main cluster, one of which is directly linked to the transposon element Polytron (Supplementary Figure 5), further emphasizing the complex relationship between satDNA arrays and genomic architecture. Interestingly, these three different patterns of Cast satDNA propagation events correlate with the average lengths of the arrays. For example, satDNAs for which only one expansion event can be observed (Cast3, Cast4, Cast8 and Cast9) have a relatively short array length (mostly around 4000 bp). SatDNAs with several expansion events, as seen in Cast2, Cast2' and Cast5, have a moderate array length of about 15000 bp. Finally, Cast1, for which no recent expansion centers were observed, also tends to have several very long arrays (up to 112kb).

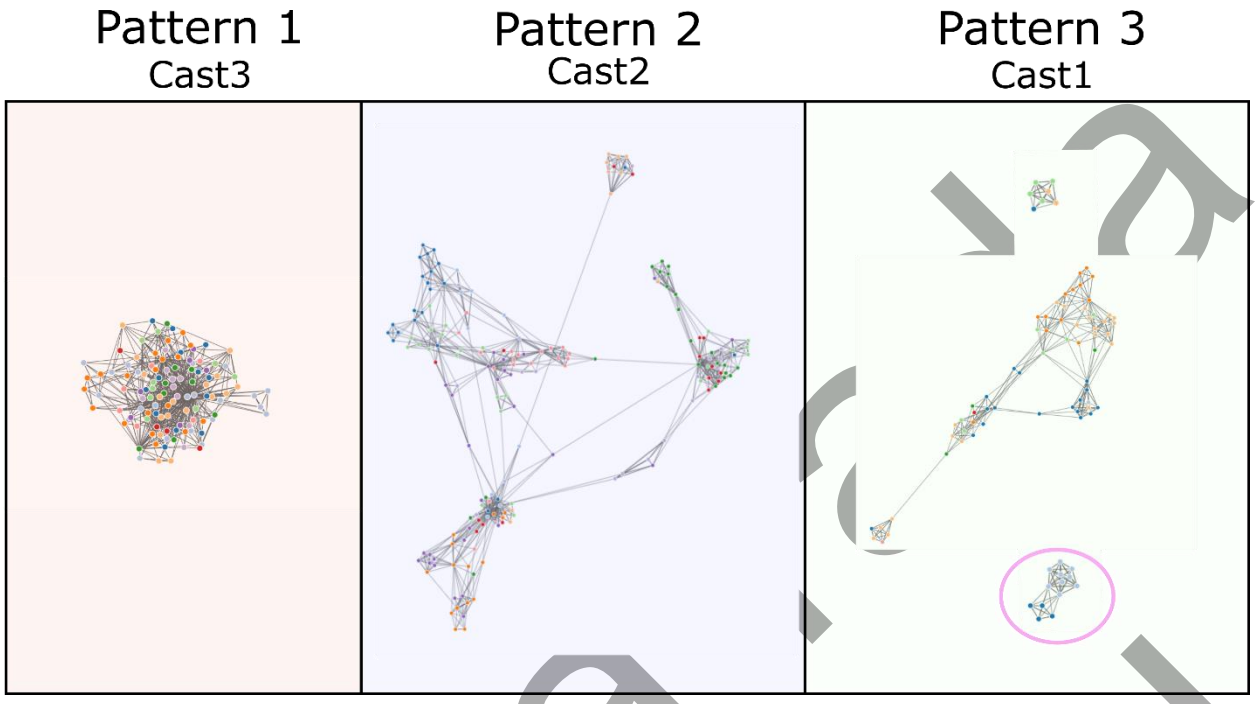


Figure 4.21 Three observed patterns of satDNA evolutionary trends. Clusters represent monomers of high intra and intra chromosomal exchange, while extended nodes relative isolation and divergence. The purple circle for Cast1 in Pattern 1 represents sequences on LG7 which are associated with Polinton-2 sequence and are evolutionary distant from the main cluster (Supplementary Figure 5)

Given the extensive inter- and intrachromosomal exchange observed in all Cast satDNAs, one possible mechanism for this phenomenon could be the insertion of satDNA arrays mediated by extrachromosomal circular DNA (eccDNA). To investigate whether Cast satDNAs are present in the eccDNA fraction, and whether they even possess the capacity for genomic expansion via eccDNA, two-dimensional (2D) agarose gel electrophoresis followed by Southern blot hybridization was conducted. Probes were developed for the most abundant Cast satDNAs—Cast1, Cast2', and Cast5 while Cast6 served as a less abundant but satDNA with long arrays. The results confirmed the presence of eccDNA molecules containing these specific satDNAs (Figure 4.22), supporting the potential role of eccDNA in facilitating satDNA spread throughout the genome.

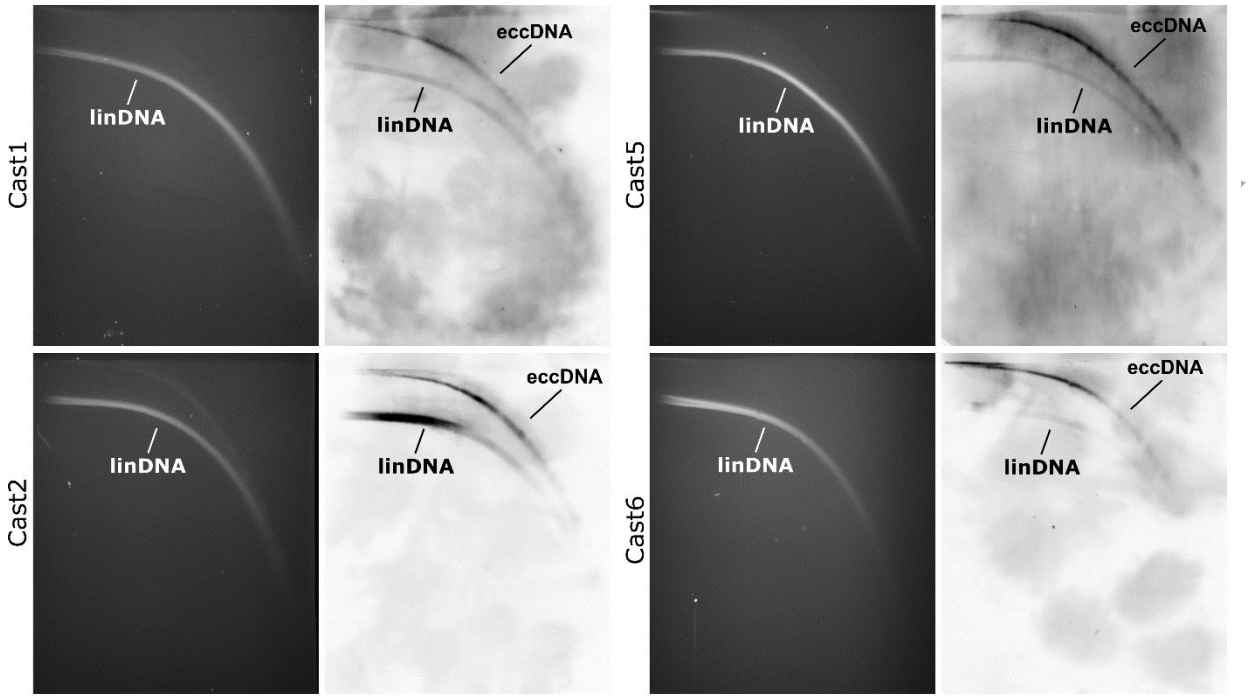


Figure 4.22 Agarose gels of extrachromosomal circular DNA for 4 satDNA families Cast1 (top left), Cast5 (top right), Cast2 (bottom left) and Cast6 (bottom right). For each family, some remnants of linear DNA remained after purification and 2D electrophoresis, additionally Cast2 and Cast5 had such fractions of eccDNA that they were visible upon normal gel inspection. All 4 satDNA had successful Southern blot staining of eccDNA. Courtesy of Damira Veseljak.

4.6 Suppression of recombination on the X chromosome

It is known that satDNAs accumulate in chromosomal regions with reduced or absent recombination [23], as these regions lack the repair mechanisms necessary to prevent integration. Since suppressed recombination often occurs in sex chromosomes because their chromosome pair is missing in one sex, we analyzed the number and length of Cast1-Cast9 arrays on the X chromosome and compared to those on autosomal chromosomes (Figure 4.23, Supplementary Figure 6). Although the Y chromosome, which is mostly non-recombining, would provide valuable insights for this analysis, it was not available in either the previous Tcas5.2 or the new TcasONT assembly due to problems in assembly and linkage mapping. When mapping the Cast1-Cast9 arrays, we found that the X chromosome does not exhibit a significantly higher average number of arrays per megabase compared to the autosomes (Supplementary Table 8).

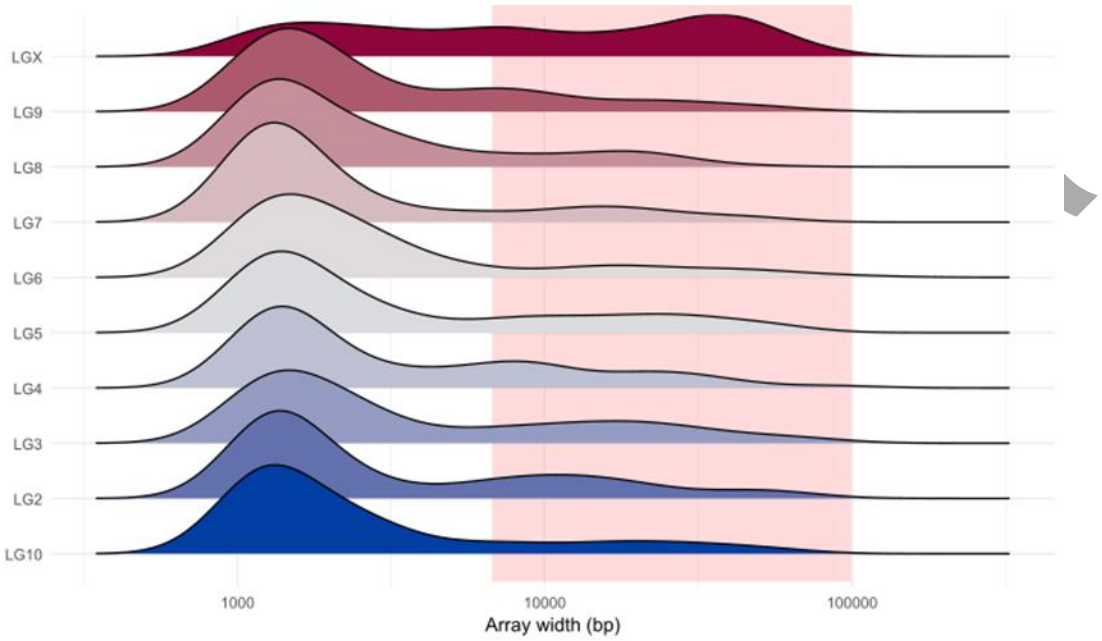


Figure 4.23 Distribution of array lengths per chromosome in TcasONT assembly. The red shaded box represents the lengths in which there is a largest increase in array lengths on the LGX chromosome.

Additionally, the sequence variability of Cast satDNA monomers on the X chromosome did not show substantial differences from those on autosomes nor significant clusters in the PCA and UMAP visualizations or graph networks. However, the array lengths on the X chromosome were statistically longer than those on the autosomes (Wilcoxon test, $p<0.05$) (Figure 4.23). This trend was particularly evident in Cast2, Cast2', Cast5, and Cast9, where array lengths on the X chromosome were up to 10 times longer than those on the autosomes, as seen especially in Cast2' (Supplementary Figure 6) providing evidence to the importance of autosome repair mechanisms in regulating satDNA propagation and elongation.

4.7 Transcription levels of Cast1-Cast9 satDNAs

The transcription of euchromatic satDNAs Cast1-Cast9 during embryonic development was analyzed using small RNAseq libraries from *T. castaneum* at various stages: oocytes, early embryos prior to zygotic transcription (0–5h), transcriptionally active early blastoderm (8–16h), differentiating blastoderm (16–20h), gastrulation (20–24h), germband elongation (24–34h), fully-extended germband (34–48h), and late-stage development up to hatching (48-144h), as retrieved from the study by Ninova et al., 2016. Additionally, transcription of Cast1-Cast9 satDNAs during *T. castaneum* development was assessed using small RNAs isolated from heads at different stages (larval, male and female pupae, male and female adults). Since the head primarily contains brain tissue, these small RNA analyses likely reflect transcription during brain development at various stages.

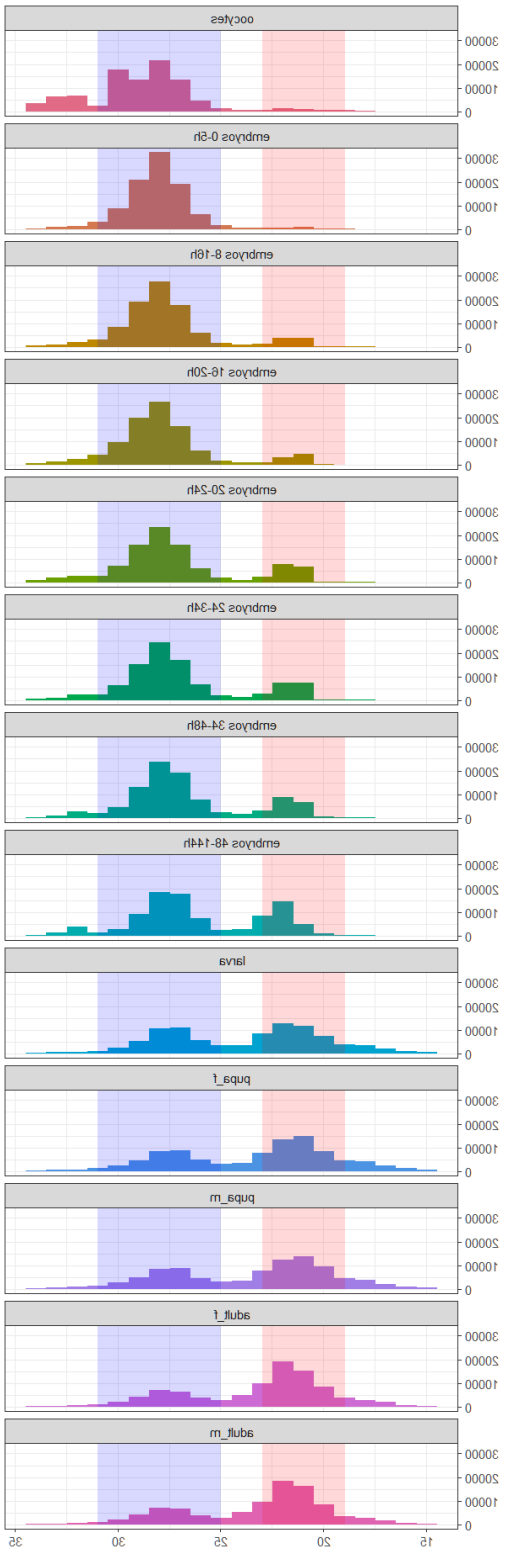


Figure 4.24 The length distribution of sequenced small RNAs (<35 nt) was analyzed across whole libraries during embryogenesis and brain development. Pink shading highlights the miRNA fraction, while blue shading represents the piRNA fraction. The X-axis shows read size, and the Y-axis displays read count per million.

The analysis of small RNA profiles during embryogenesis and brain development revealed an interesting trend (Figure 4.24). In oocytes and early embryos (0-5h), the small RNA population is almost entirely composed of piRNAs. As embryogenesis progresses, miRNAs start to appear alongside piRNAs, with their proportion steadily increasing toward the later stages of embryogenesis. This rise in miRNA levels continues through brain development, reaching equal levels with piRNAs during the pupal stage. In adults, the balance shifts further, with miRNAs becoming more abundant than piRNAs in both females and males. To explore the transcriptional activity of euchromatic satDNAs (Cast1-Cast9) during these stages, small RNA reads were mapped to the consensus sequences of Cast monomers (Figure 4.25) and the number of hits was standardized according to library size and genome abundancy.

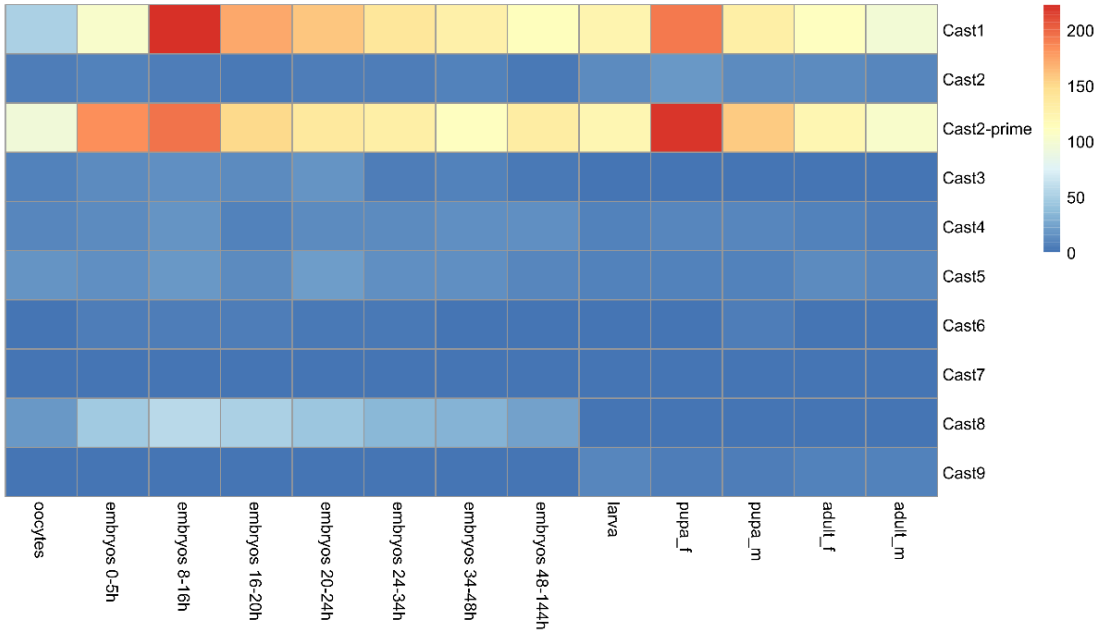


Figure 4.25 Library and genome size standardized expression levels of small RNAs in various Cast satDNAs (Cast1-Cast9) measured across different stages of embryogenesis (oocytes and embryos from 0–144h) and brain development (larva, male pupae, female pupae, adult male, and adult female).

The results show that a considerable number of expressed miRNAs are associated with Cast1 and Cast2' satDNAs, both of which show distinct transcriptional patterns throughout development. In contrast, Cast8 shows weak and relatively uniform expression during embryogenesis and almost no expression during

brain development. The expression patterns of Cast1 and Cast2' satDNAs are quite similar, with a notable increase in transcription from the oocyte stage to the 8-16h embryo, where a peak in transcription is observed. Following this peak, transcription gradually decreases toward the end of embryogenesis. During brain development, both Cast1 and Cast2' exhibit differential transcription, with the highest expression observed in the female pupal brain and the lowest in the early oocytes and late male adult phases.

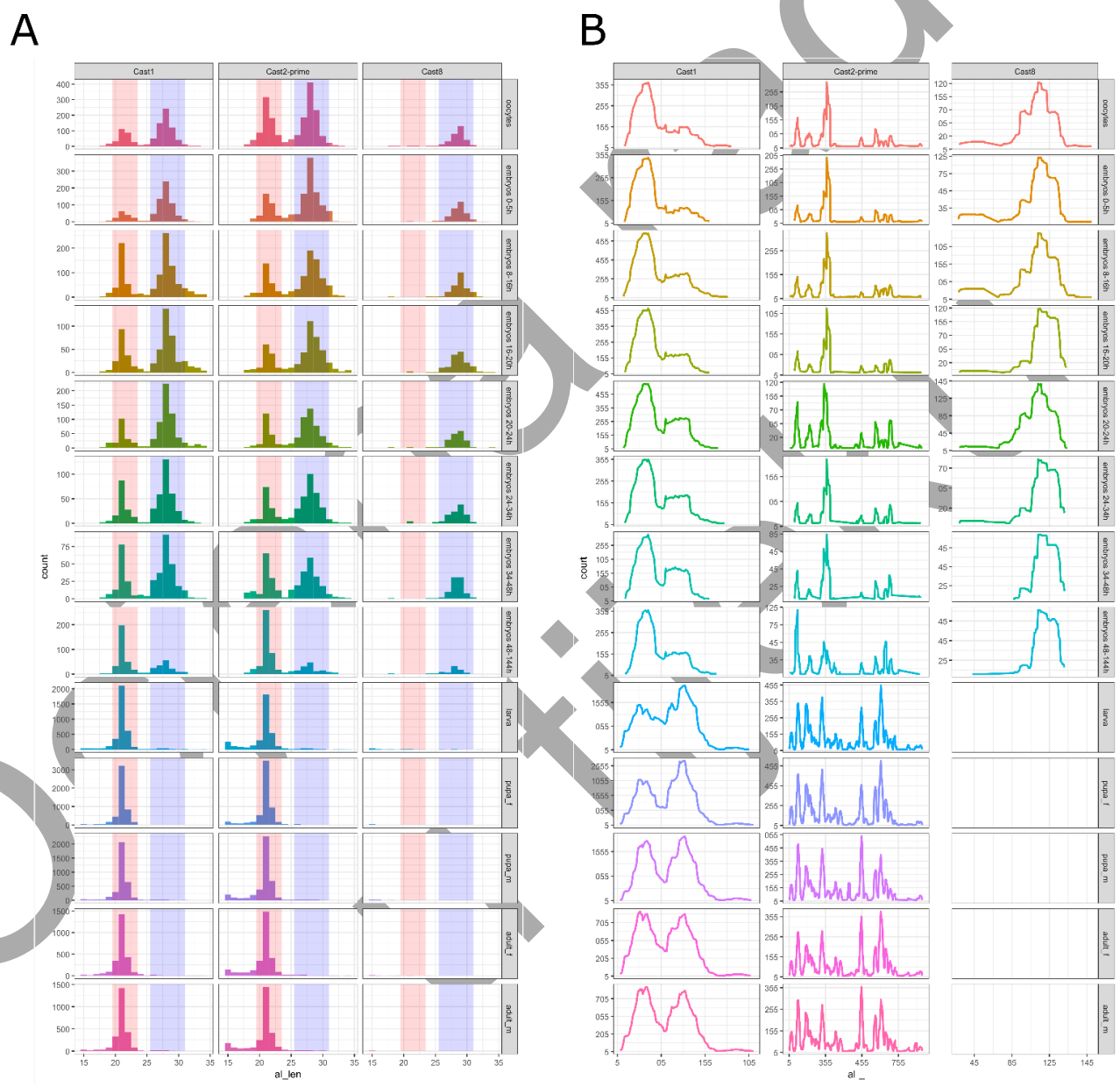


Figure 4.26 A Small RNA library read length distribution mapped to euchromatic satDNAs, Cast1, Cast2' and Cast8. Pink shading on the distribution denotes read lengths associated with miRNA profiles (19-23bp) while the blue shade represents piRNA fraction (26-32bp). **B** Coverage depth of Cast1, Cast2' and Cast8 monomers by small RNA reads during embryogenesis and brain development.

In addition, we analyzed the populations of small RNAs associated with these differentially transcribed Cast satDNAs (Figure 4.26A). During embryonic development, the Cast1 satellite DNA produces abundant small RNAs mainly from a 14–47 bp region of its monomer (Figure 24B). As development progresses, small RNAs also map to the 52-84 bp region, particularly during brain development, where both regions serve equally as precursors of small RNAs. For the Cast2' monomer, which is much longer (1102 bp) than Cast1 and Cast8 (~170 bp), a larger number of regions containing small RNAs are mapped. In the first third of the Cast2' monomer (1-300 bp), three dominant regions persist throughout brain development, while four additional regions emerge in the last third (600-900 bp) during brain development. In contrast, Cast8 shows a single prominent region between 90-130 bp during embryogenesis when it is transcribed. To identify potential genomic target sequences of these small RNAs, we mapped 462,079 predicted miRNA target sequences with miRanda, and the results revealed that these target sequences are exclusively mapped to the three Cast satDNAs. This indicates that there are no sequences outside of these satDNAs that could be potential targets for Cast-specific miRNAs.

5. Discussion

Due to substantial progress in sequencing technologies and bioinformatics tools, the burden of generating new genome assemblies has become a lot easier to overcome. However, satDNA remains one of the most challenging parts of the genome to assemble. A clear example of this difficulty is seen in the effort to assemble the relatively small but highly repetitive genomes of holocentric nematodes [110]. The assemblies were fragmented, with the authors highlighting that the abundant and dispersed satDNA within holocentromeres is the primary factor causing fragmentation and preventing chromosome-level assembly [43], [110]. Given that the *T. castaneum* genome contains numerous families of repetitive sequences, particularly satDNAs [116], [146], it's unsurprising that the official reference assembly exhibited significant gaps in its representation of these repetitive sequences, including euchromatic satDNAs. Although evidence suggests that euchromatic satDNAs have some functional roles, our understanding of their organization, evolutionary dynamics, and the molecular mechanisms driving their dispersal, movement, and rearrangement within euchromatin remains limited. The primary aim of this research was to conduct a comprehensive study of satDNAs within the euchromatin of *T. castaneum*, which requires an assembly enriched in repetitive regions. Since the reference assembly Tcas5.2 did not meet this requirement, our first step was to enhance the assembly of repetitive DNA regions in *T. castaneum*. To achieve this, we generated a high-quality genome assembly at the chromosome level by combining nanopore long-read sequencing with a reference-guided approach.

5.1 Newly developed isolation protocol

Considering that the most critical factor for successful nanopore long-read sequencing is the extraction of high molecular weight (HMW) DNA in sufficient purity and quantity, the first task was to optimize the isolation protocol for *T. castaneum*. Due to the problems of using conventional isolation methods and commercial kits and their application for ONT sequencing, which did not provide DNA of sufficient quality or quantity for Nanopore sequencing, a new combined isolation and sequencing protocol was developed. While the nuclei isolation protocol presented by Brown and Coleman [147] was a useful starting point, it required further improvement, particularly for the then-unexploited application of Nanopore sequencing. Although the commercial kits tested provide DNA of sufficient length, the DNA pellets were difficult to dissolve, causing problems during purification and centrifugation.

Our results of optimizing the isolation of HMW DNA showed that the amount of starting material proved to be critical, especially in larvae and adult beetles, where non-cellular components such as fat and chitin required more starting material, leading to nonlinear relationships between weight of starting material and DNA yield. The major drawback of this method therefore proved to be the huge amount of starting tissue needed for proper nuclei isolation and subsequent DNA spooling. It ranged from 200mg for pupae and eggs to >1g for larvae and adults. Although it is possible to achieve even larger DNA of higher molecular weight using this protocol, N50 >1Mb could lead to clumps of hard-to-dissolve DNA and consequently rapid pore death during sequencing, especially for AT and repeat rich genomes such as the *Tribolium* species. Based on this DNA isolation with N50 <200kb fragment length turned out to be optimal for nanopore sequencing. Additionally, mechanical shearing using 31-gauge needles improved sequencing output significantly, since the number of ultra-long fragments was reduced, and the subsequent cleanup of short fragments using Circulomics XS had a larger effect on final library and sequencing output.

Key steps added in the newly developed protocol include fully resuspending the nuclear pellets, usage of gentle wide-bore pipetting, ensuring that the isolated DNA is spooled in isopropanol rather than centrifugated. Additionally, we found that High nucleic DNA input can result slow flow rates during purification requiring manual pressurization, but ultimately does not affect DNA quality. The increased viscosity of the DNA eluted from the columns indicates higher molecular weight and quantity. Spooled DNA in EB buffer formed a “jelly-like” mass that required prolonged relaxation at increased temperatures (up to 50 ^{0}C) indicating a high degree of entanglement and high molecular weight. The sequencing output of Nanopore libraries prepared from such DNA combined with elongated waiting times proved to be vastly better than all other possible variations of commercial kits and their outputs, and given the method’s success in isolating HMW DNA from three related beetle species, it is recommended as a reliable starting point for isolation from other Coleoptera species and even beyond.

5.2 New genome assembly of *T. castaneum* using Oxford Nanopore Sequencing technology

At the time of this study, the official assembly of *T. castaneum* was Tcas5.2, which was incomplete, as more than 25% of the estimated genome size of 204 Mb was missing, as confirmed by in silico analyses. Approximately 27% of the *T. castaneum* genome is repetitive, and the Tcas5.2 assembly, created using Illumina short-read sequencing and optical mapping, had significant problems in assembling repetitive

regions, especially satDNAs. This posed a problem for this study, which aimed to investigate the structural, evolutionary, and putative biological roles of nine highly abundant satellite DNAs (satDNAs), Cast1-Cast9, most of which were absent from the Tcas5.2 assembly.

The long-read sequencing via Nanopore technology was employed to achieve the necessary continuity for a more complete genome assembly. To this goal, a high-quality *T. castaneum* genome assembly at the chromosome level was generated by combining nanopore long-read sequencing and a reference-guided approach. The output of 89 Gb of Nanopore data enabled the creation of a long-read assembly. The TcasONT assembled chromosomes lack only 13 Mb of the estimated *T. castaneum* genome sequence of 204 Mb, previously determined experimentally [148] and also in silico in our study. The missing 13 Mb could primarily be attributed to (peri)centromeric regions, due to assembly-impeding highly repetitive TCAST satDNA regions [149]. This gap in (peri)centromeric regions is consistent with the challenges faced by even large research consortia in capturing the entire (peri)centromeric regions of genomes such as *A. thaliana* and *H. sapiens* [3], [5].

Regarding gene completeness, only 8 genes were missing in TcasONT, while 60 genes were missing in Tcas5.2. Furthermore, repeat content analysis showed that TcasONT added 47.8 Mb of repetitive sequences, almost completely capturing the repetitive elements except for the (peri)centromere of the *T. castaneum* genome and achieving a 20-fold enrichment of repetitive regions. The TcasONT assembly revealed a remarkable increase in satDNA representation, especially for repeats longer than 50 bp, and accounting for 10% of the genome, making TcasONT a suitable platform for in-depth analysis of satDNAs. This significant improvement allowed for a more detailed analysis of euchromatic Cast1-Cast9 satDNAs. The abundance of Cast1-Cast9 satDNAs in TcasONT was quantified at 8.8 Mb, accounting for 4.6% of the genome, a figure consistent with experimental data. In addition, a TcastONT assembly enabled the detection of a new, highly abundant euchromatic satDNA related to Cast2, Cast2'. The in-depth analysis of their distribution revealed that Cast2' and Cast5 show the largest increases in TcasONT relative to Tcas5.2, due to their large repeat length (~1100bp Cast2' and 340bp for Cast5) and the ability to form large arrays, which were previously omitted.

To understand the genomic organization of these satDNAs, a new algorithm was developed to precisely detect satDNA arrays at the whole genome scale. This automated method replaced the laborious manual inspection of the arrays, and made it possible to obtain detailed information about the genomic landscape

of Cast1-Cast9 satDNAs. Given the potential loss of redundant overlaps in long noisy reads, particularly in satDNA arrays, it was essential to validate the representation of these arrays in the TcasONT assembly. A comparative analysis of array profiles in the TcasONT assembly and a random subsample of sequencing reads confirmed that the assembly accurately reflects the genome's satDNA landscape. This validation provided a strong foundation for in-depth analysis of the organization and evolution of Cast1-Cast9 satDNAs, enabling further exploration of their structural and functional roles in the *T. castaneum* genome.

5.3 Genomic organization of Cast1-Cast9 satDNAs

The findings from this study, confirm the presence of the ten "classical" satDNAs (Cast1-Cast9, along with the newly identified Cast2') in the form of long tandem arrays within euchromatic regions. These regions, while less permissive to the accumulation of satDNAs compared to (peri)centromeric heterochromatin, still support and accommodate these arrays. Moreover, our gene density analyses showed that the surrounding regions of the arrays of almost all Cast satDNAs correlated positively with the gene-rich regions compared to the average gene density in the genome. This discovery challenges the earlier assumption that euchromatic satDNAs would primarily localize to distal regions of the centromere and in regions bounded by centromeric satDNA [26]. Instead, these satDNAs are distributed distally on the chromosomal arms, away from centromeric heterochromatin. The hypothesis that these euchromatic satDNAs might accumulate in genomic regions of lesser importance, such as those consisting of other repetitive elements such as transposons, was tested by analysing gene and transposon density in the vicinity of the Cast1-Cast9 arrays. In contrast to this hypothesis, the results revealed that these satDNAs reside in gene-rich regions do not overlap with transposons. Notably, 950 of the total 2900 arrays overlapped with lifted gene annotations, indicating that a significant portion of these arrays is embedded within intron bodies. Furthermore, the distribution analysis showed that satDNAs are positioned in transposon-poor regions, rarely associating with transposable elements or regions linked to them. The distances between satDNA arrays and nearby exons were consistently small, further supporting their localization within gene-dense euchromatic areas in arrays of different sizes. Furthermore, the sharp drop in exon densities relative to array starts and ends, particularly in long array, suggests that these arrays are

often located at gene boundaries, potentially serving as regulators of gene activity and have impact on chromatin formation.

The results from our study also challenge previous models that satDNA accumulation is primarily a feature of genomic regions characterized by suppressed recombination and genetic repair, suggesting that recombination suppresses array expansion and may even lead to array loss [150]–[152]. Instead, our results suggest that satDNAs are capable to integrate into gene-rich regions where recombination does not prevent their spread but inhibit the elongation of these arrays. This is especially evident in the analysis of satDNA length distributions across autosomes and the X chromosome. The suppressed recombination on the X chromosome appears to stimulate the formation of longer arrays, but has no significant effect on the number of arrays or their sequence variability. This suggests that recombination in euchromatic regions limits the elongation of satDNA arrays, but it does not prevent their integration. Once integrated, the lengths of arrays appear to become "fixed," establishing a balance between satDNA propagation and the genomic mechanisms that limit their expansion.

5.4 Evolutionary trends and propagation mechanisms of Cast1-Cast9 satDNAs

Due to widespread distribution of satDNAs in euchromatin and their potential impact on genome evolution, the study raises the question of how these satDNAs propagate. While the mechanisms of TE propagation are quite well understood, such as independent retrotransposons LINE elements in human and the non-autonomous MITE and SINE elements which utilize the machinery of other elements [18]; the propagation of satDNAs, particularly within euchromatin, remains elusive. To explore this, we analyzed the distribution patterns, genome dynamics, and junction regions of Cast1-Cast9 satDNAs. The results revealed that these satDNA arrays are dispersed across all chromosomes, extending along their entire lengths without any regional preference except from a noticeable trend of their placement being on distal parts of the chromosome rather than the pericentromere. Dimensionality reduction analysis confirmed previous findings that long arrays tend to homogenize [153] which is evident in the large clusters for Cast5 and Cast6 which belong to the same chromosome and array. Additionally, results from this analysis confirmed the frequent inter and intrachromosomal exchange events involving small and intermediate arrays, as evidenced by the absence of clustering in Cast3, Cast9 and Cast8. Furthermore, junction region analysis revealed that these satDNAs, with the exception of Cast5 and Cast7 are rarely associated with

other repetitive sequences. There are several examples in other species where dispersed satDNAs have been discovered as short arrays integrated into central repeats of non-autonomous transposon elements. For example, tandem repeats within the Tetris transposon in *Drosophila virilis* have been reported to form the basis for the formation of long satellite arrays, that eventually lose the transposon features in adjacent regions[154]. Similarly, our findings suggest that the expansion of Cast5 satDNA is likely associated with the Mariner transposon element, given the presence of transposon-like sequences in its surrounding regions. However, besides Cast5, the other Cast satDNAs do not appear to be consistently associated with mobile elements, but they also show extensive propagation. Furthermore, conserved microhomology regions characterized by the presence of poly-A/T tracts are found in junction regions of some Cast satDNAs which could represent a preferential insertion site. These findings suggest presence of an efficient self-propagation mechanism that operates both within and across chromosomes.

Regarding their evolutionary history, graph-network analysis of sequence similarities among arrays for each Cast satDNA showed that they group into three distinct patterns. Under the presumption that these patterns represent snapshots of satDNA activity at specific time points, it is possible to construct a timeline that explains the genome dynamics and propagation of euchromatic satDNAs (Figure 5.1). Initially, a single expansion event may originate from one center, resulting in short arrays spreading rapidly across different chromosomes (t1) retaining high sequence similarities. As these arrays are localized in different chromosomal regions, they start to diverge in sequence due to reduced inter-array homogenization, and some arrays continue to elongate if located in favorable environments (t2). At a later stage, short arrays could serve as new expansion centers, initiating further dispersal events (t3). Over time, satDNA arrays may enter a dormant phase in which they don't spread further, but continue to expand in length and homogenize(t4). Although the exact triggers for satDNA dispersal are still unclear, the observed patterns suggest an efficient mechanism that drives the widespread distribution of these sequences across the genome.

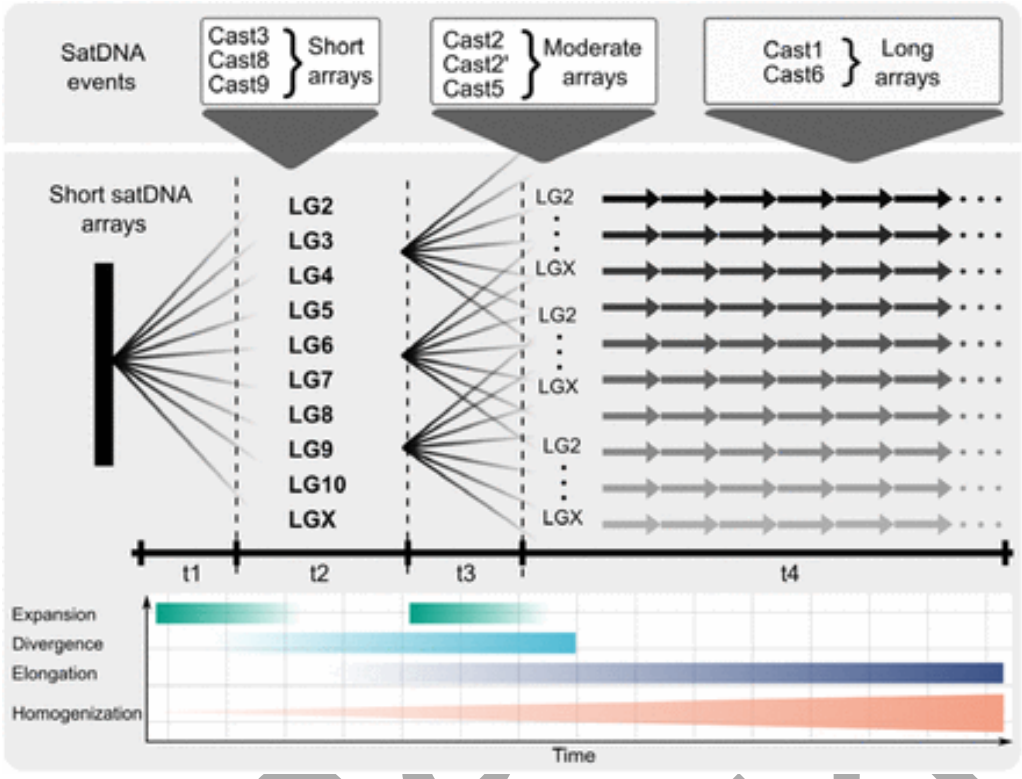


Figure 5.1 Timeline of satDNA expansion events from a single locus of origin. Initially, a sudden burst leads to the rapid spread of satDNA across different chromosomes, followed by a secondary expansion event that further propagates the sequences. The timeline of events includes distinct phases marked by different colors: expansion can occur in multiple discrete bursts, characterized by a sharp increase in intensity, which is followed by a slower decline. Divergence occurs over a longer period, becoming more pronounced with time as long arrays get fixed into their genomic locations. Elongation begins later but has a progressively greater effect, while homogenization remains an ongoing process, particularly influencing the longest arrays.

In general, three main mechanisms for satDNA propagation have been proposed so far: (i) dispersion in short arrays, potentially integrated as central repeats within non-autonomous transposable elements; (ii) spread through long distances via extrachromosomal circular DNAs (eccDNAs); and (iii) interlocus gene conversion via 3D interactions between loci in the interphase nucleus [108]. Although satDNAs were traditionally considered less mobile than TEs, this study reveals that satDNAs also possess a significant ability to spread throughout the genome. TEs are known to proliferate in periodic bursts, often linked to stressful conditions like heat, irradiation, or chemical exposure, as observed in *D. melanogaster*[21]. Our findings suggest that euchromatic satDNAs exhibit a similar pattern of genomic dynamics, with repeated bursts of expansion. This suggests that satDNAs may also initiate expansion cycles triggered by external stressors. Further support for this comes from the observation that *T. castaneum* euchromatic satDNAs,

which are counterparts to pericentromeric satDNAs, also show increased expression in response to heat stress [155].

Recent studies in *D. melanogaster* confirmed two potential mechanisms driving satDNA spread throughout euchromatin: reintegration via eccDNA and interlocus gene conversion, particularly on the X chromosome [108]. Presence of Cast satDNAs in the eccDNA fraction suggests that eccDNA-mediated reintegration may play a significant role in the spread of euchromatic satDNA in *T. castaneum*. In summary, the results of clustering patterns, neighboring regions and junction regions analysis as well as the presence of satDNAs in the eccDNA fraction, two key mechanisms responsible for the genome dynamics and evolution of euchromatic satDNAs could be proposed: transposition and eccDNA insertion.

5.5 Transcriptional activity of satDNAs

We investigated the expression of small RNAs during embryonic and brain development in *T. castaneum*, focusing on understanding the transcription patterns of the most abundant euchromatic satDNAs, Cast1-Cast9, in embryogenesis and in the development of highly differentiated organs such as the brain. The small RNA profile during embryogenesis and brain development in *T. castaneum* revealed a notable trend: piRNAs dominated in oocytes and throughout early embryogenesis, while miRNAs steadily increased towards the end of embryogenesis and became the dominant small RNA type in the adult brain. Furthermore, our study analyzed the transcription of ten euchromatic satDNAs during these stages, with three satDNAs found to produce small RNAs in significant quantities. Among them, Cast1 and Cast2' showed stage-specific transcriptional peaks in early blastoderm during embryogenesis and the female pupal brain. Notably, Cast1 and Cast2' exhibit a similar transcriptional processing mechanism: during embryogenesis, transcripts are processed into both miRNAs and piRNAs, but in brain development, they are exclusively processed into miRNAs. The lack of predicted genomic targets of Cast-specific small RNAs suggest a self-regulatory role for these sequences, in contrary to functions typically associated to piRNAs such as stem cell maintenance and meiosis in *D. melanogaster* [156] with their highest intensity in the germline [157]. Although piRNAs are typically linked to transposon regulation, their involvement in tandem repeat regulation, as seen in Bombyx mori female embryos where piRNA-mediated signaling affects sexual differentiation [158], indicates that Cast1 and Cast2' could be promising candidates for further RNAi knockdown experiments to investigate potential phenotypic effects, particularly since *T. castaneum*

efficiently transmits RNAi effects to offspring. Furthermore, given that 80% of miRNA knockouts in Drosophila result in visible phenotypes [159], it is likely that miRNAs in *T. castaneum* also play crucial roles in cellular processes during late embryogenesis and brain development.

5.6 Potential biological roles of Cast1-Cast9 satDNAs

Given what we know about the evolution, propagation, and transcriptional activity of satDNAs, the next crucial step is to disclose their impact on the genome. Previous research on euchromatic satDNA with short arrays and monomer-length repeats suggests that these satDNAs may be involved in gene regulation, acting as "evolutionary tuning knobs" by modulating chromatin [38] and playing a role in processes like X chromosome recognition and dosage compensation [160]. SatDNAs located in euchromatic regions might regulate gene expression by influencing local chromatin structure or through transcripts derived from the repetitive sequences. For instance, contractions of the human subtelomeric satellite D4Z4 can modify the chromatin state of adjacent genes, leading to disease such as like muscular dystrophy [25].

In Drosophila, introns-containing satDNA have been shown to be transcribed along with their associated genes, requiring specific mechanisms to overcome the challenges posed by long stretches of repetitive DNA, such as R-loop formation [161]. Recent studies have also shown that euchromatic satDNA-derived transcripts play a role in the control of embryonic development in mosquitoes through sequence-specific gene silencing [34]. Additionally, the transcription of α-satellite DNAs is regulated by Topoisomerase I (TopI) in response to double-strand breaks, a process conserved across species such as mouse 3T3 cells and Drosophila S2 cells, as well as Drosophila larval imaginal wing discs and tumors [162] and may also apply to euchromatic satDNAs. Furthermore, pericentric satellites in mice, exhibit a transient peak in expression during chromocenter formation, that follows a developmental clock; when replication is inhibited, chromocenter formation is halted underscoring the importance of satellite DNA in development and chromatin organization[163]. Moreover, considering that satDNAs are located in gene-rich regions, their epigenetic regulation, such as the presence of repressive histone marks like H3K9me3, may impact neighboring gene expression. A genome-wide analysis in humans demonstrated that euchromatic satDNAs are associated with such repressive marks, suggesting their influence on gene regulation [35].

In addition to directly affecting gene expression, large-scale genomic rearrangements involving long arrays of Cast1-Cast9 satDNAs scattered throughout the genome are highly probable. The rapid evolutionary

turnover of euchromatic satDNAs could contribute to a rapid change in the genes' landscape, and affect gene function and overall genome dynamics. In summary, the widespread presence of Cast1-Cast9 satDNAs in the euchromatic regions of the *T. castaneum* genome likely exerts significant influence on gene expression, genome organization, and evolutionary dynamics, making them important targets for further investigation.

6. Conclusions

We have produced the most contiguous genome assembly of *T. castaneum* to date with the significant improvement in the representation of the repetitive genome portion by Oxford Nanopore long-read sequencing. The new genome is enriched by up to 1/4 of the genome size, especially in the repetitive part such as transposable elements and satellite DNAs. In addition to enrichment in the repetitive part, predicted genomic completeness also increased compared to the former Tcas5.2 assembly.

We found that our approach has been extremely efficient in bridging highly repetitive regions in *T. castaneum*. We believe that our approach could be useful for all species for which reference genomes have been published but whose assemblies are significantly deficient and unassembled in repetitive regions. In particular, it could be important for genomes that are highly repetitive even outside the (peri)centromere. Our genome assembly enriched with repetitive genome parts will provide a highly reliable data point for future comparative analyses of the repetitive genome fraction in related species to find putative conserved traits in these extremely variable genome parts. This will be a crucial step in understanding the evolution of the genome as a whole.

We have shown that enhanced genome assembly provides an exceptional platform for in-depth genome-wide analyses of different and the most abundant satellite DNAs in euchromatin. We provided significant insights into the behavior and organization of these euchromatic satDNAs in *T. castaneum*, challenging previously assumptions about their localization and propagation. Contrary to earlier hypotheses, which assumed that satDNAs are mainly located in gene-poor regions, such as (peri)centromeric regions or regions abundant with transposable elements, our study reveals that satDNAs can also be embedded in gene-rich regions, even in the form of long tandem arrays.

From an evolutionary perspective, this study provided evidence of highly efficient mechanism of self-propagation and homogenization of satDNA arrays in gene-rich regions. The long arrays tend to homogenize, with frequent inter- and intrachromosomal exchanges. Most analyzed satDNAs did not associate with other repetitive elements but their presence in the eccDNA fraction strongly suggested that eccDNA-mediated reintegration is probably a major force in the spread of these sequences. We proposed a new model of their genome dynamics characterized by repeated bursts of satDNAs spreading through euchromatin, followed by a process of elongation and homogenization of arrays. Recombination

appears to limit the elongation of satDNA arrays, but has no impact on the frequency of their integration into gene-rich regions.

Examination of transcriptional activity of euchromatic satDNAs during embryogenesis and brain development revealed interesting trends. Of the 10 euchromatic satDNAs analyzed, three were transcribed and processed into small RNAs. In embryogenesis, transcripts are processed into both miRNAs and piRNAs, whereas transcripts in the brain were exclusively processed into miRNAs. The absence of other genomic Cast-specific small RNAs suggests that the processed RNAs play a role exclusively in a self-regulatory mechanism. Two of them showed differential transcription with peaks in the early blastoderm during embryogenesis and in the female pupal brain. The presence of piRNAs in brain tissue indicates a unique regulatory system in *T. castaneum* with Cast1 and Cast2' satDNAs as promising candidates for RNAi experiments to uncover their potential roles in *T. castaneum* development and genome evolution.

Finally, such dynamical sequences with transcriptional potential embedded in euchromatin, which are subject to changes and rearrangements, would have an extraordinary potential for rapid evolution of the genome and consequently of the species itself. This opens a new perspective on satDNAs by considering them as inevitable parts of euchromatin, thus stimulating new research involving epigenetic studies, which could disclose their role and putative influence on gene content.

7. References

[1] N. Dia *et al.*, "Subtelomere organization in the genome of the microsporidian Encephalitozoon cuniculi: Patterns of repeated sequences and physicochemical signatures," *BMC Genomics*, vol. 17, no. 1, pp. 1–20, 2016, doi: 10.1186/s12864-015-1920-7.

[2] P. Fernández *et al.*, "A 160 Gbp fork fern genome shatters size record for eukaryotes," *iScience*, vol. 27, no. 6, 2024, doi: 10.1016/j.isci.2024.109889.

[3] M. Naish *et al.*, "The genetic and epigenetic landscape of the Arabidopsis centromeres," *Science (80-.).*, vol. 374, no. 6569, pp. 1–23, 2021, doi: 10.1126/science.abi7489.

[4] J. Yang *et al.*, "Chloroplast phylogenomic analysis provides insights into the evolution of the largest eukaryotic genome holder, Paris japonica (Melanthiaceae)," *BMC Plant Biol.*, vol. 19, no. 1, pp. 1–11, 2019, doi: 10.1186/s12870-019-1879-7.

[5] N. Sergey *et al.*, "The complete sequence of a human genome," *Science (80-.).*, vol. 376, no. 6588, pp. 44–53, Apr. 2022, doi: 10.1126/science.abj6987.

[6] V. Wood *et al.*, "The genome sequence of Schizosaccharomyces pombe," *Nature*, vol. 421, no. 6918, pp. 94–94, 2003, doi: 10.1038/nature01203.

[7] J. M. Carlton *et al.*, "Draft genome sequence of the sexually transmitted pathogen Trichomonas vaginalis," *Science (80-.).*, vol. 315, no. 5809, pp. 207–212, Jan. 2007, doi: 10.1126/SCIENCE.1132894.

[8] S. Nowoshilow *et al.*, "The axolotl genome and the evolution of key tissue formation regulators," *Nature*, vol. 554, no. 7690, pp. 50–55, 2018, doi: 10.1038/nature25458.

[9] I. Y. Choi, E. C. Kwon, and N. S. Kim, "The C- and G-value paradox with polyploidy, repeatomes, introns, phenomes and cell economy," *Genes and Genomics*, vol. 42, no. 7, pp. 699–714, 2020, doi: 10.1007/s13258-020-00941-9.

[10] M. Pertea and S. L. Salzberg, "Between a chicken and a grape: estimating the number of human genes," *Genome Biol.*, vol. 11, no. SUPPL. 1, 2010, doi: 10.1186/gb-2010-11-s1-I1.

[11] "» How big are genomes?" https://book.bionumbers.org/how-big-are-genomes/ (accessed Oct. 16, 2024).

[12] T. A. Elliott and T. R. Gregory, "What's in a genome? The C-value enigma and the evolution of eukaryotic genome content," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 370, no. 1678, 2015, doi: 10.1098/rstb.2014.0331.

[13] J. N. Wells and C. Feschotte, "A Field Guide to Eukaryotic Transposable Elements," *Annu. Rev. Genet.*, vol. 54, pp. 539–561, 2020, doi: 10.1146/annurev-genet-040620-022145.

[14] C. R. Beck, J. L. Garcia-Perez, R. M. Badge, and J. V. Moran, "LINE-1 elements in structural

variation and disease," *Annu. Rev. Genomics Hum. Genet.*, vol. 12, pp. 187–215, 2011, doi: 10.1146/annurev-genom-082509-141802.

[15] E. M. McCarthy and J. F. McDonald, "Long terminal repeat retrotransposons of Mus musculus," *Genome Biol.*, vol. 5, no. 3, 2004, doi: 10.1186/gb-2004-5-3-r14.

[16] C. Feschotte and E. J. Pritham, "DNA Transposons and the Evolution of Eukaryotic Genomes," *Annu. Rev. Genet.*, vol. 41, p. 331, 2007, doi: 10.1146/ANNUREV.GENET.40.110405.090448.

[17] J. K. Pace and C. Feschotte, "The evolutionary history of human DNA transposons: Evidence for intense activity in the primate lineage," *Genome Res.*, vol. 17, no. 4, pp. 422–432, 2007, doi: 10.1101/gr.5826307.

[18] S. Ayarpadikannan and H.-S. Kim, "The Impact of Transposable Elements in Genome Evolution and Genetic Instability and Their Implications in Various Diseases," *Genomics Inform.*, vol. 12, no. 3, p. 98, 2014, doi: 10.5808/gi.2014.12.3.98.

[19] J. M. Aury *et al.*, "Long-read and chromosome-scale assembly of the hexaploid wheat genome achieves high resolution for research and breeding," *Gigascience*, vol. 11, pp. 1–18, 2022, doi: 10.1093/gigascience/giac034.

[20] G. Haberer *et al.*, "European maize genomes highlight intraspecies variation in repeat and gene content," *Nat. Genet.*, vol. 52, no. 9, pp. 950–957, 2020, doi: 10.1038/s41588-020-0671-9.

[21] V. Mérel, M. Boulesteix, M. Fablet, and C. Vieira, "Transposable Elements in *Drosophila melanogaster*," in *Mobile DNA II*, Wiley, 2020, pp. 484–518. doi: 10.1186/s13100-020-00213-z.

[22] V. Peska and S. Garcia, "Origin, Diversity, and Evolution of Telomere Sequences in Plants," *Front. Plant Sci.*, vol. 11, no. February, pp. 1–9, 2020, doi: 10.3389/fpls.2020.00117.

[23] J. Thakur, J. Packiaraj, and S. Henikoff, "Sequence, chromatin and evolution of satellite DNA," *Int. J. Mol. Sci.*, vol. 22, no. 9, 2021, doi: 10.3390/ijms22094309.

[24] C. Ramel, "Mini- and microsatellites," *Environ. Health Perspect.*, vol. 105, no. SUPPL. 4, pp. 781–789, 1997, doi: 10.1289/ehp.97105s4781.

[25] G. Dumbovic, S. V. Forcales, and M. Perucho, "Emerging roles of macrosatellite repeats in genome organization and disease development," *Epigenetics*, vol. 12, no. 7, pp. 515–526, 2017, doi: 10.1080/15592294.2017.1318235.

[26] M. A. Garrido-Ramos, "Satellite DNA: An evolving topic," *Genes (Basel).*, vol. 8, no. 9, 2017, doi: 10.3390/genes8090230.

[27] S. Ohno, "So much 'junk' DNA in our genome.," *Brookhaven Symp. Biol.*, vol. 23, pp. 366–370, 1972.

[28] S. M. Mcnulty and B. A. Sullivan, "Alpha satellite DNA biology: finding function in the recesses of the genome," pp. 115–138, 2018.

[29] J. G. Henikoff, J. Thakur, S. Kasinathan, and S. Henikoff, "A unique chromatin complex occupies young a-satellite arrays of human centromeres," *Sci. Adv.*, vol. 1, no. 1, 2015, doi:

10.1126/sciadv.1400234.

[30] A. S. Komissarov, E. V. Gavrilova, S. J. Demin, A. M. Ishov, and O. I. Podgornaya, “Tandemly repeated DNA families in the mouse genome,” *BMC Genomics*, vol. 12, 2011, doi: 10.1186/1471-2164-12-531.

[31] I. S. Kuznetsova, A. N. Prusov, N. I. Enukashvily, and O. I. Podgornaya, “New types of mouse centromeric satellite DNAs,” *Chromosom. Res.*, vol. 13, no. 1, pp. 9–25, 2005, doi: 10.1007/s10577-005-2346-x.

[32] X. Sun, H. D. Le, J. M. Wahlstrom, and G. H. Karpen, “Sequence analysis of a functional Drosophila centromere,” *Genome Res.*, vol. 13, no. 2, pp. 182–194, 2003, doi: 10.1101/gr.681703.

[33] S. Richards *et al.*, “The genome of the model beetle and pest Tribolium castaneum,” *Nature*, vol. 452, no. 7190, pp. 949–955, 2008, doi: 10.1038/nature06784.

[34] R. Halbach *et al.*, “A satellite repeat-derived piRNA controls embryonic development of *Aedes*,” *Nature*, vol. 580, no. 7802, pp. 274–277, 2020, doi: 10.1038/s41586-020-2159-2.

[35] I. Feliciello, A. Sermek, Ž. Pezer, M. Matulić, and Đ. Ugarković, “Heat stress affects H3K9me3 level at human alpha satellite DNA repeats,” *Genes (Basel).*, vol. 11, no. 6, pp. 1–17, 2020, doi: 10.3390/genes11060663.

[36] A. T. Jonstrup, T. Thomsen, Y. Wang, B. R. Knudsen, J. Koch, and A. H. Andersen, “Hairpin structures formed by alpha satellite DNA of human centromeres are cleaved by human topoisomerase IIα,” *Nucleic Acids Res.*, vol. 36, no. 19, pp. 6165–6174, 2008, doi: 10.1093/nar/gkn640.

[37] J. Brajković, I. Feliciello, B. Bruvo-MadWarić, and D. W. Ugarković, “Satellite DNA-like elements associated with genes within euchromatin of the beetle tribolium castaneum,” *G3 Genes, Genomes, Genet.*, vol. 2, no. 8, pp. 931–941, 2012, doi: 10.1534/g3.112.003467.

[38] D. G. King, M. Soller, and Y. Kashi, “Evolutionary tuning knobs,” *Endeavour*, vol. 21, no. 1, pp. 36–40, 1997, doi: 10.1016/S0160-9327(97)01005-3.

[39] S. Ando, H. Yang, N. Nozaki, T. Okazaki, and K. Yoda, “CENP-A, -B, and -C chromatin complex that contains the I-type alpha-satellite array constitutes the prekinetochore in HeLa cells.,” *Mol. Cell. Biol.*, vol. 22, no. 7, pp. 2229–2241, Apr. 2002, doi: 10.1128/MCB.22.7.2229-2241.2002.

[40] P. Wlodzimierz *et al.*, “Cycles of satellite and transposon evolution in Arabidopsis centromeres,” *Nature*, vol. 618, no. 7965, pp. 557–565, 2023, doi: 10.1038/s41586-023-06062-z.

[41] S. Kasinathan and S. Henikoff, “Non-B-form DNA is enriched at centromeres,” *Mol. Biol. Evol.*, vol. 35, no. 4, pp. 949–962, 2018, doi: 10.1093/molbev/msy010.

[42] Y. Muro, H. Masumoto, K. Yoda, N. Nozaki, M. Ohashi, and T. Okazaki, “Centromere protein B assembles human centromeric α-satellite DNA at the 17-bp sequence, CENP-B box,” *J. Cell Biol.*, vol. 116, no. 3, pp. 585–596, 1992, doi: 10.1083/jcb.116.3.585.

[43] E. Despot-Slade, B. Mravinac, S. Širca, P. Castagnone-Sereno, M. Plohl, and N. Meštrović, “The

Centromere Histone Is Conserved and Associated with Tandem Repeats Sharing a Conserved 19-bp Box in the Holocentromere of Meloidogyne Nematodes," *Mol. Biol. Evol.*, vol. 38, no. 5, pp. 1943–1965, 2021, doi: 10.1093/molbev/msaa336.

[44] S. M. McNulty, L. L. Sullivan, and B. A. Sullivan, "Human Centromeres Produce Chromosome-Specific and Array-Specific Alpha Satellite Transcripts that Are Complexed with CENP-A and CENP-C," *Dev. Cell*, vol. 42, no. 3, pp. 226-240.e6, 2017, doi: 10.1016/j.devcel.2017.07.001.

[45] X. Wei, D. G. Eickbush, I. Speece, and A. M. Larracuente, "Heterochromatin-dependent transcription of satellite DNAs in the Drosophila melanogaster female germline.," *Elife*, vol. 10, Jul. 2021, doi: 10.7554/eLife.62375.

[46] O. M. Palacios-Gimenez, V. B. Bardella, B. Lemos, and D. C. Cabral-De-Mello, "Satellite DNAs are conserved and differentially transcribed among Gryllus cricket species," *DNA Res.*, vol. 25, no. 2, pp. 137–147, 2018, doi: 10.1093/dnares/dsx044.

[47] S. Louzada *et al.*, "Architecture and Plasticity — An Evolutionary and Clinical A ffair," *Genes (Basel).*, 2020.

[48] O. V. Camacho *et al.*, "Major satellite repeat RNA stabilize heterochromatin retention of Suv39h enzymes by RNA-nucleosome association and RNA:DNA hybrid formation," *Elife*, vol. 6, pp. 1–29, 2017, doi: 10.7554/eLife.25293.

[49] R. Valgardsdottir *et al.*, "Transcription of Satellite III non-coding RNAs is a general stress response in human cells," *Nucleic Acids Res.*, vol. 36, no. 2, pp. 423–434, 2008, doi: 10.1093/nar/gkm1056.

[50] G. C. S. Kuhn, "Satellite DNA transcripts have diverse biological roles in Drosophila," *Heredity (Edinb).*, vol. 115, no. 1, pp. 1–2, 2015, doi: 10.1038/hdy.2015.12.

[51] R. Chaves, D. Ferreira, A. Mendes-Da-Silva, S. Meles, and F. Adega, "FA-SAT is an old satellite DNA frozen in several bilateria genomes," *Genome Biol. Evol.*, vol. 9, no. 11, pp. 3073–3087, 2017, doi: 10.1093/gbe/evx212.

[52] E. Despot-Slade, S. Širca, B. Mravinac, P. Castagnone-Sereno, M. Plohl, and N. Meštrović, "Satellitome analyses in nematodes illuminate complex species history and show conserved features in satellite DNAs," *BMC Biol.*, vol. 20, no. 1, pp. 1–19, 2022, doi: 10.1186/s12915-022-01460-7.

[53] G. Bosco, P. Campbell, J. T. Leiva-Neto, and T. A. Markow, "Analysis of Drosophila species genome size and satellite DNA content reveals significant differences among strains as well as between species," *Genetics*, vol. 177, no. 3, pp. 1277–1290, 2007, doi: 10.1534/genetics.107.075069.

[54] G. Dover, "Molecular drive," *Trends Genet.*, vol. 18, no. 11, pp. 587–589, Nov. 2002, doi: 10.1016/S0168-9525(02)02789-0.

[55] N. Meštrović, M. Plohl, B. Mravinac, and D. Ugarković, "Evolution of satellite DNAs from the genus Palorus - Experimental evidence for the 'library' hypothesis," *Mol. Biol. Evol.*, vol. 15, no. 8, pp. 1062–1068, 1998, doi: 10.1093/oxfordjournals.molbev.a026005.

[56] G. C. S. Kuhn, H. Küttler, O. Moreira-Filho, and J. S. Heslop-Harrison, "The 1.688 repetitive DNA of

drosophila: Concerted evolution at different genomic scales and association with genes," *Mol. Biol. Evol.*, vol. 29, no. 1, pp. 7–11, 2012, doi: 10.1093/molbev/msr173.

[57] K. H. C. Wei, J. K. Grenier, D. A. Barbash, and A. G. Clark, "Correlated variation and population differentiation in satellite DNA abundance among lines of Drosophila melanogaster," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 52, pp. 18793–18798, 2014, doi: 10.1073/pnas.1421951112.

[58] M. E. Quesada Del Bosque, R. Navajas-Pérez, J. L. Panero, A. Fernández-González, M. A. Garrido-Ramos, and P. Gustafson, "A satellite DNA evolutionary analysis in the North American endemic dioecious plant Rumex hastatulus (Polygonaceae)," *Genome*, vol. 54, no. 4, pp. 253–260, 2011, doi: 10.1139/g10-115.

[59] F. Sanger *et al.*, "Nucleotide sequence of bacteriophage phi X174 DNA.," *Nature*, vol. 265, no. 5596, pp. 687–695, Feb. 1977, doi: 10.1038/265687a0.

[60] "GenBank and WGS Statistics." https://www.ncbi.nlm.nih.gov/genbank/statistics/ (accessed Oct. 16, 2024).

[61] E. S. Lander *et al.*, "Initial sequencing and analysis of the human genome: International Human Genome Sequencing Consortium (Nature (2001) 409 (860-921))," *Nature*, vol. 412, no. 6846, pp. 565–566, 2001, doi: 10.1038/35087627.

[62] J. G. McEwen and O. M. Gómez, "Genome sequencing using long-read sequencing," *Rev. la Acad. Colomb. Ciencias Exactas, Fis. y Nat.*, vol. 47, no. 183, pp. 439–444, 2023, doi: 10.18257/raccefyn.1937.

[63] Z. Li *et al.*, "Comparison of the two major classes of assembly algorithms: Overlap-layout-consensus and de-bruijn-graph," *Brief. Funct. Genomics*, vol. 11, no. 1, pp. 25–37, 2012, doi: 10.1093/bfgp/elr035.

[64] A. M. Giani, G. R. Gallo, L. Gianfranceschi, and G. Formenti, "Long walk to genomics: History and current approaches to genome sequencing and assembly.," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 9–19, 2020, doi: 10.1016/j.csbj.2019.11.002.

[65] K. D. Christensen, D. Dukhovny, U. Siebert, and R. C. Green, "Assessing the Costs and Cost-Effectiveness of Genomic Sequencing.," *J. Pers. Med.*, vol. 5, no. 4, pp. 470–486, Dec. 2015, doi: 10.3390/jpm5040470.

[66] "DNA Sequencing Costs: Data." https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data (accessed Oct. 16, 2024).

[67] M. D. Adams *et al.*, "The genome sequence of Drosophila melanogaster," *Science (80-.).*, vol. 287, no. 5461, pp. 2185–2195, 2000, doi: 10.1126/science.287.5461.2185.

[68] C. elegans S. Consortium, "Genome sequence of the nematode C. elegans: a platform for investigating biology.," *Science*, vol. 282, no. 5396, pp. 2012–2018, Dec. 1998, doi: 10.1126/science.282.5396.2012.

[69] T. A. G. Initiative, "Analysis of the genome sequence of the flowering plant Arabidopsis thaliana," *Nature*, vol. 408, no. 6814, pp. 796–815, 2000, doi: 10.1038/35048692.

[70] A. T. Chinwalla *et al.*, "Initial sequencing and comparative analysis of the mouse genome," *Nature*, vol. 420, no. 6915, pp. 520–562, 2002, doi: 10.1038/nature01262.

[71] P. S. Schnable *et al.*, "The B73 maize genome: Complexity, diversity, and dynamics," *Science (80-.).*, vol. 326, no. 5956, pp. 1112–1115, 2009, doi: 10.1126/science.1178534.

[72] N. Nagarajan and M. Pop, "Parametric complexity of sequence assembly: Theory and applications to next generation sequencing," *J. Comput. Biol.*, vol. 16, no. 7, pp. 897–908, 2009, doi: 10.1089/cmb.2009.0005.

[73] M. J. P. Chaisson, R. K. Wilson, and E. E. Eichler, "Genetic variation and the de novo assembly of human genomes," *Nat. Rev. Genet.*, vol. 16, no. 11, pp. 627–640, 2015, doi: 10.1038/nrg3933.

[74] "Homo sapiens genome assembly GRCh38.p14 - NCBI - NLM." https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001405.40/ (accessed Oct. 16, 2024).

[75] A. Rhoads and K. F. Au, "PacBio Sequencing and Its Applications.," *Genomics. Proteomics Bioinformatics*, vol. 13, no. 5, pp. 278–289, Oct. 2015, doi: 10.1016/j.gpb.2015.08.002.

[76] Y. Wang, Q. Yang, and Z. Wang, "The evolution of nanopore sequencing," *Front. Genet.*, vol. 5, no. DEC, pp. 1–20, 2014, doi: 10.3389/fgene.2014.00449.

[77] Y. Liu *et al.*, "Comparison of structural variants detected by PacBio-CLR and ONT sequencing in pear," *BMC Genomics*, vol. 23, no. 1, pp. 1–14, 2022, doi: 10.1186/s12864-022-09074-7.

[78] P. Kelleher, J. Murphy, J. Mahony, and D. van Sinderen, "Identification of DNA Base Modifications by Means of Pacific Biosciences RS Sequencing Technology.," *Methods Mol. Biol.*, vol. 1681, pp. 127–137, 2018, doi: 10.1007/978-1-4939-7343-9_10.

[79] C. L. Hall, R. R. Zascavage, F. J. Sedlazeck, and J. V Planz, "Potential applications of nanopore sequencing for forensic analysis.," *Forensic Sci. Rev.*, vol. 32, no. 1, pp. 23–54, Jan. 2020.

[80] M. Pagès-Gallego and J. de Ridder, "Comprehensive benchmark and architectural analysis of deep learning models for nanopore sequencing basecalling," *Genome Biol.*, vol. 24, no. 1, pp. 1–18, 2023, doi: 10.1186/s13059-023-02903-2.

[81] "From kilobases to \"whales\": a short history of ultra-long reads and high-throughput genome sequencing | Oxford Nanopore Technologies." https://nanoporetech.com/blog/news-blog-kilobases-whales-short-history-ultra-long-reads-and-high-throughput-genome (accessed Oct. 17, 2024).

[82] K. H. Miga *et al.*, "Telomere-to-telomere assembly of a complete human X chromosome," *Nature*, vol. 585, no. 7823, pp. 79–84, 2020, doi: 10.1038/s41586-020-2547-7.

[83] "nanoporetech/dorado: Oxford Nanopore's Basecaller." https://github.com/nanoporetech/dorado (accessed Oct. 17, 2024).

[84] "Nanopore DNA Sequencing - StoryMD." https://storymd.com/journal/j43vox8hnm-genetics-glossary-letter-n-o/page/27v45ksybog9-nanopore-dna-sequencing (accessed Oct. 17, 2024).

[85] S. El-Metwally, T. Hamza, M. Zakaria, and M. Helmy, "Next-generation sequence assembly: four

stages of data processing and computational challenges.," *PLoS Comput. Biol.*, vol. 9, no. 12, p. e1003345, 2013, doi: 10.1371/journal.pcbi.1003345.

[86] H. Cheng, G. T. Concepcion, X. Feng, H. Zhang, and H. Li, "Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm," *Nat. Methods 2021 182*, vol. 18, no. 2, pp. 170–175, Feb. 2021, doi: 10.1038/s41592-020-01056-5.

[87] S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman, and A. M. Phillippy, "Canu: Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation," *Genome Res.*, vol. 27, no. 5, pp. 722–736, 2017, doi: 10.1101/gr.215087.116.

[88] S. Nurk *et al.*, "HiCanu: Accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads," *Genome Res.*, vol. 30, no. 9, pp. 1291–1305, 2020, doi: 10.1101/GR.263566.120.

[89] M. Kolmogorov, J. Yuan, Y. Lin, and P. A. Pevzner, "Assembly of long, error-prone reads using repeat graphs," *Nat. Biotechnol.*, vol. 37, no. 5, pp. 540–546, 2019, doi: 10.1038/s41587-019-0072-8.

[90] J. Ruan and H. Li, "Fast and accurate long-read assembly with wtdbg2.," *Nat. Methods*, vol. 17, no. 2, pp. 155–158, Feb. 2020, doi: 10.1038/s41592-019-0669-3.

[91] C. Zhou, S. A. McCarthy, and R. Durbin, "YaHS: yet another Hi-C scaffolding tool," *Bioinformatics*, vol. 39, no. 1, pp. 10–12, 2023, doi: 10.1093/bioinformatics/btac808.

[92] X. Zhang, S. Zhang, Q. Zhao, R. Ming, and H. Tang, "Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data," *Nat. Plants*, vol. 5, no. 8, pp. 833–845, 2019, doi: 10.1038/s41477-019-0487-8.

[93] M. Rautiainen *et al.*, "Telomere-to-telomere assembly of diploid chromosomes with Verkko," *Nat. Biotechnol.*, vol. 41, no. 10, pp. 1474–1482, 2023, doi: 10.1038/s41587-023-01662-6.

[94] M. K. N. Lawniczak, R. Durbin, P. Flicek, K. Lindblad-toh, and X. Wei, "Standards recommendations for the Earth BioGenome Project," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 119, no. 4, pp. 1–8, 2022, doi: 10.1073/pnas.2115639118/-/DCSupplemental.Published.

[95] K.-P. Koepfli, B. Paten, and S. J. O'Brien, "The Genome 10K Project: a way forward.," *Annu. Rev. Anim. Biosci.*, vol. 3, pp. 57–111, 2015, doi: 10.1146/annurev-animal-090414-014900.

[96] M. Blaxter *et al.*, "Sequence locally, think globally: The Darwin Tree of Life Project," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 119, no. 4, pp. 1–7, 2022, doi: 10.1073/pnas.2115642118.

[97] M. Alonge *et al.*, "Automated assembly scaffolding using RagTag elevates a new tomato system for high-throughput genome editing," *Genome Biol.*, vol. 23, no. 1, pp. 1–19, 2022, doi: 10.1186/s13059-022-02823-7.

[98] M. Xu *et al.*, "TGS-GapCloser: A fast and accurate gap closer for large genomes with low coverage of error-prone long reads," *Gigascience*, vol. 9, no. 9, pp. 1–11, 2020, doi: 10.1093/gigascience/giaa094.

[99] A. Shumate and S. L. Salzberg, “Liftoff: accurate mapping of gene annotations,” *Bioinformatics*, vol. 37, no. 12, pp. 1639–1643, Jul. 2021, doi: 10.1093/BIOINFORMATICS/BTAA1016.

[100] G. Benson, “Tandem repeats finder: A program to analyze DNA sequences,” *Nucleic Acids Res.*, vol. 27, no. 2, pp. 573–580, 1999, doi: 10.1093/nar/27.2.573.

[101] D. Olson and T. Wheeler, “ULTRA: A Model Based Tool to Detect Tandem Repeats.,” *ACM-BCB ... ... ACM Conf. Bioinformatics, Comput. Biol. Biomed. ACM Conf. Bioinformatics, Comput. Biol. Biomed.*, vol. 2018, pp. 37–46, 2018, doi: 10.1145/3233547.3233604.

[102] P. Wlodzimierz, M. Hong, and I. R. Henderson, “TRASH: Tandem Repeat Annotation and Structural Hierarchy,” *Bioinformatics*, vol. 39, no. 5, p. btad308, May 2023, doi: 10.1093/bioinformatics/btad308.

[103] P. Novák, L. Ávila Robledillo, A. Koblížková, I. Vrbová, P. Neumann, and J. Macas, “TAREAN: a computational tool for identification and characterization of satellite DNA from unassembled short reads.,” *Nucleic Acids Res.*, vol. 45, no. 12, p. e111, Jul. 2017, doi: 10.1093/nar/gkx257.

[104] R. Ebrahimzadegan, A. Houben, and G. Mirzaghaderi, “Repetitive DNA landscape in essential A and supernumerary B chromosomes of Festuca pratensis Huds,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, 2019, doi: 10.1038/s41598-019-56383-1.

[105] P. Novák, P. Neumann, and J. Macas, “Global analysis of repetitive DNA from unassembled sequence reads using RepeatExplorer2,” *Nat. Protoc.*, vol. 15, no. 11, pp. 3745–3776, 2020, doi: 10.1038/s41596-020-0400-y.

[106] B. S. M. L. Silva, P. Heringer, G. B. Dias, M. Svartman, and G. C. S. Kuhn, “De novo identification of satellite DNAs in the sequenced genomes of Drosophila virilis and D. americana using the RepeatExplorer and TAREAN pipelines,” *PLoS One*, vol. 14, no. 12, pp. 1–15, 2019, doi: 10.1371/journal.pone.0223466.

[107] T. E. Samatadze *et al.*, “Genome Studies in Four Species of Calendula L. (Asteraceae) Using Satellite DNAs as Chromosome Markers,” *Plants*, vol. 12, no. 23, 2023, doi: 10.3390/plants12234056.

[108] J. S. Sproul *et al.*, “Dynamic Evolution of Euchromatic Satellites on the X Chromosome in Drosophila melanogaster and the simulans Clade.,” *Mol. Biol. Evol.*, vol. 37, no. 8, pp. 2241–2256, Aug. 2020, doi: 10.1093/molbev/msaa078.

[109] J. Chen *et al.*, “A complete telomere-to-telomere assembly of the maize genome.,” *Nat. Genet.*, vol. 55, no. 7, pp. 1221–1231, Jul. 2023, doi: 10.1038/s41588-023-01419-6.

[110] A. P. Z. Mota *et al.*, “Unzipped genome assemblies of polyploid root-knot nematodes reveal unusual and clade-specific telomeric repeats.,” *Nat. Commun.*, vol. 15, no. 1, p. 773, Feb. 2024, doi: 10.1038/s41467-024-44914-y.

[111] G. A. Logsdon *et al.*, “The variation and evolution of complete human centromeres,” *Nature*, vol. 629, no. 8010, pp. 136–145, 2024, doi: 10.1038/s41586-024-07278-3.

[112] M. Klingler and G. Bucher, “The red flour beetle T. castaneum: elaborate genetic toolkit and

unbiased large scale RNAi screening to study insect biology and evolution," *Evodevo*, vol. 13, no. 1, pp. 1–11, 2022, doi: 10.1186/s13227-022-00201-9.

[113] S. Richards *et al.*, "The genome of the model beetle and pest *Tribolium castaneum*," *Nature*, vol. 452, no. 7190, pp. 949–955, 2008, doi: 10.1038/nature06784.

[114] N. Herndon *et al.*, "Enhanced genome assembly and a new official gene set for Tribolium castaneum.," *BMC Genomics*, vol. 21, no. 1, p. 47, 2020, doi: 10.1186/s12864-019-6394-6.

[115] S. J. Brown, J. K. Henry, W. C. Black IV, and R. E. Denell, "Molecular genetic manipulation of the red flour beetle: Genome organization and cloning of a ribosomal protein gene," *Insect Biochem.*, vol. 20, no. 2, pp. 185–193, 1990, doi: https://doi.org/10.1016/0020-1790(90)90011-I.

[116] S. Wang, M. D. Lorenzen, R. W. Beeman, and S. J. Brown, "Analysis of repetitive DNA distribution patterns in the Tribolium castaneum genome," *Genome Biol.*, vol. 9, no. 3, pp. 1–14, 2008, doi: 10.1186/gb-2008-9-3-r61.

[117] M. Plohl, V. Lucijanic-Justic, D. Ugarkovic, E. Petitpierre, and C. Juan, "Satellite DNA and heterochromatin of the flour beetle Tribolium confusum," *Genome*, vol. 36, no. 3, pp. 467–475, 1993, doi: 10.1139/g93-064.

[118] T. Gržan, E. Despot-Slade, N. Meštrović, M. Plohl, and B. Mravinac, "CenH3 distribution reveals extended centromeres in the model beetle Tribolium castaneum," *PLOS Genet.*, vol. 16, no. 10, p. e1009115, Oct. 2020, doi: 10.1371/journal.pgen.1009115.

[119] I. Feliciello, G. Chinali, and D. Ugarković, "Structure and population dynamics of the major satellite DNA in the red flour beetle Tribolium castaneum.," *Genetica*, vol. 139, no. 8, pp. 999–1008, Aug. 2011, doi: 10.1007/s10709-011-9601-1.

[120] M. Pavlek, Y. Gelfand, M. Plohl, and N. Meštrović, "Genome-wide analysis of tandem repeats in Tribolium castaneum genome reveals abundant and highly dynamic tandem repeat families with satellite DNA features in euchromatic chromosomal arms," *DNA Res.*, vol. 22, no. 6, pp. 387–401, Dec. 2015, doi: 10.1093/DNARES/DSV021.

[121] T. Gržan *et al.*, "The Low-Copy-Number Satellite DNAs of the Model Beetle Tribolium castaneum," *Genes (Basel).*, vol. 14, no. 5, 2023, doi: 10.3390/genes14050999.

[122] M. Gross-Bellard, P. Oudet, and P. Chambon, "Isolation of High-Molecular-Weight DNA from Mammalian Cells," *Eur. J. Biochem.*, vol. 36, no. 1, pp. 32–38, Jul. 1973, doi: https://doi.org/10.1111/j.1432-1033.1973.tb02881.x.

[123] C. Hussing, M. L. Kampmann, H. S. Mogensen, C. Børsting, and N. Morling, "Comparison of techniques for quantification of next-generation sequencing libraries," *Forensic Sci. Int. Genet. Suppl. Ser.*, vol. 5, pp. e276–e278, 2015, doi: https://doi.org/10.1016/j.fsigss.2015.09.110.

[124] M. Malumbres, R. Mangues, N. Ferrer, S. Lu, and A. Pellicer, "Isolation of High Molecular Weight DNA for Reliable Genotyping of Transgenic Mice," *Biotechniques*, vol. 22, no. 6, pp. 1114–1119, Jun. 1997, doi: 10.2144/97226st03.

[125] S. Chan *et al.*, "Structural Variation Detection and Analysis Using Bionano Optical Mapping BT -

Copy Number Variants: Methods and Protocols," D. M. Bickhart, Ed. New York, NY: Springer New York, 2018, pp. 193–203. doi: 10.1007/978-1-4939-8666-8_16.

[126] H. A. Dahn *et al.*, "Benchmarking ultra-high molecular weight DNA preservation methods for long-read and long-range sequencing," *Gigascience*, vol. 11, p. giac068, Jan. 2022, doi: 10.1093/gigascience/giac068.

[127] J. N. Schultzhaus *et al.*, "Comparison of seven methods for DNA extraction from prosomata of the acorn barnacle, Amphibalanus amphitrite," *Anal. Biochem.*, vol. 586, p. 113441, 2019, doi: https://doi.org/10.1016/j.ab.2019.113441.

[128] B. Oppert, S. Stoss, A. Monk, and T. Smith, "Optimized Extraction of Insect Genomic DNA for Long-Read Sequencing.," *Methods Protoc.*, vol. 2, no. 4, Nov. 2019, doi: 10.3390/mps2040089.

[129] S. J. Brown and M. Coleman, "Isolation of High Molecular Weight DNA from Insects BT - Insect Genomics: Methods and Protocols," S. J. Brown and M. E. Pfrender, Eds. New York, NY: Springer New York, 2019, pp. 27–32. doi: 10.1007/978-1-4939-8775-7_3.

[130] R. Vaser, I. Sović, N. Nagarajan, and M. Šikić, "Fast and accurate de novo genome assembly from long uncorrected reads," *Genome Res.*, vol. 27, no. 5, pp. 737–746, 2017, doi: 10.1101/gr.214270.116.

[131] H. Li, "Minimap2: Pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018, doi: 10.1093/bioinformatics/bty191.

[132] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov, "BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs," *Bioinformatics*, vol. 31, no. 19, pp. 3210–3212, 2015, doi: 10.1093/bioinformatics/btv351.

[133] P. Smit, AFA, Hubley, R & Green, "RepeatMasker Open-4.0." 2015.

[134] M. Benoit and H. G. Drost, "A Predictive Approach to Infer the Activity and Natural Variation of Retrotransposon Families in Plants," *Methods Mol. Biol.*, vol. 2250, pp. 1–14, 2021, doi: 10.1007/978-1-0716-1134-0_1.

[135] K. Katoh and D. M. Standley, "MAFFT multiple sequence alignment software version 7: Improvements in performance and usability," *Mol. Biol. Evol.*, vol. 30, no. 4, pp. 772–780, 2013, doi: 10.1093/molbev/mst010.

[136] E. Paradis and K. Schliep, "Ape 5.0: An environment for modern phylogenetics and evolutionary analyses in R," *Bioinformatics*, vol. 35, no. 3, pp. 526–528, 2019, doi: 10.1093/bioinformatics/bty633.

[137] S. Lê, J. Josse, and F. Husson, "FactoMineR : An R Package for Multivariate Analysis," *J. Stat. Softw.*, vol. 25, no. 1, pp. 253–258, Feb. 2008, doi: 10.18637/jss.v025.i01.

[138] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. [Online]. Available: https://ggplot2.tidyverse.org

[139] S. Cohen and S. Lavi, "Induction of Circles of Heterogeneous Sizes in Carcinogen-Treated Cells:

Two-Dimensional Gel Analysis of Circular DNA Molecules," *Mol. Cell. Biol.*, vol. 16, no. 5, pp. 2002–2014, May 1996, doi: 10.1128/MCB.16.5.2002.

[140] M. Ninova, M. Ronshaugen, and S. Griffiths-Jones, "MicroRNA evolution, expression, and function during short germband development in Tribolium castaneum.," *Genome Res.*, vol. 26, no. 1, pp. 85–96, Jan. 2016, doi: 10.1101/gr.193367.115.

[141] J. Dönitz, L. Gerischer, S. Hahnke, S. Pfeiffer, and G. Bucher, "Expanded and updated data and a query pipeline for iBeetle-Base," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D831–D835, Jan. 2018, doi: 10.1093/nar/gkx984.

[142] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome Biol.*, vol. 10, no. 3, 2009, doi: 10.1186/gb-2009-10-3-r25.

[143] "Babraham Bioinformatics - Trim Galore!" https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/ (accessed Oct. 17, 2024).

[144] P. Danecek *et al.*, "Twelve years of SAMtools and BCFtools.," *Gigascience*, vol. 10, no. 2, Feb. 2021, doi: 10.1093/gigascience/giab008.

[145] M. Morgan, H. Pagès, V. Obenchain, and N. Hayden, "Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import." 2024. [Online]. Available: https://bioconductor.org/packages/Rsamtools

[146] M. Pavlek, Y. Gelfand, M. Plohl, and N. Meštrović, "Genome-wide analysis of tandem repeats in Tribolium castaneum genome reveals abundant and highly dynamic tandem repeat families with satellite DNA features in euchromatic chromosomal arms.," *DNA Res. an Int. J. rapid Publ. reports genes genomes*, vol. 22, no. 6, pp. 387–401, Dec. 2015, doi: 10.1093/dnares/dsv021.

[147] S. J. Brown and M. Coleman, "Isolation of high molecular weight DNA from insects," *Methods Mol. Biol.*, vol. 1858, pp. 27–32, 2019, doi: 10.1007/978-1-4939-8775-7_3.

[148] A. Alvarez-Fuster, C. Juan, and E. Petitpierre, "Genome size in Tribolium flour-beetles: Inter-and intraspecific variation," *Genet. Res.*, vol. 58, no. 1, pp. 1–5, 1991, doi: 10.1017/S0016672300029542.

[149] Đ. Ugarković, M. Podnar, and M. Plohl, "Satellite DNA of the Red Flour Beetle Tribolium castaneum-Comparative Study of Satellites from the Genus Tribolium," 1996.

[150] B. Charlesworth, C. H. Langley, and W. Stephan, "THE EVOLUTION OF RESTRICTED RECOMBINATION AND THE ACCUMULATION OF REPEATED DNA SEQUENCES," *Genetics*, vol. 112, no. 4, pp. 947–962, Apr. 1986, doi: 10.1093/genetics/112.4.947.

[151] S. S. Lower, M. P. McGurk, A. G. Clark, and D. A. Barbash, "Satellite DNA evolution: old ideas, new approaches," *Curr. Opin. Genet. Dev.*, vol. 49, pp. 70–78, 2018, doi: https://doi.org/10.1016/j.gde.2018.03.003.

[152] W. Stephan, "Nonlinear Phenomena in the Evolution of Satellite DNA," no. June 1985, pp. 1029–1034, 1986.

[153] T. Vondrak, L. Ávila Robledillo, P. Novák, A. Koblížková, P. Neumann, and J. Macas, "Characterization of repeat arrays in ultra-long nanopore reads reveals frequent origin of satellite DNA from retrotransposon-derived tandem repeats.," *Plant J.*, vol. 101, no. 2, pp. 484–500, Jan. 2020, doi: 10.1111/tpj.14546.

[154] G. B. Dias, M. Svartman, A. Delprat, A. Ruiz, and G. C. S. Kuhn, "Tetris Is a Foldback Transposon that Provided the Building Blocks for an Emerging Satellite DNA of Drosophila virilis," *Genome Biol. Evol.*, vol. 6, no. 6, pp. 1302–1313, Jun. 2014, doi: 10.1093/gbe/evu108.

[155] Z. Pezer and D. Ugarkovic, "Satellite DNA-associated siRNAs as mediators of heat shock response in insects.," *RNA Biol.*, vol. 9, no. 5, pp. 587–595, May 2012, doi: 10.4161/rna.20019.

[156] A. Girard, R. Sachidanandam, G. J. Hannon, and M. A. Carmell, "A germline-specific class of small RNAs binds mammalian Piwi proteins," *Nature*, vol. 442, no. 7099, pp. 199–202, 2006, doi: 10.1038/nature04917.

[157] J. Brennecke *et al.*, "Discrete small RNA-generating loci as master regulators of transposon activity in Drosophila.," *Cell*, vol. 128, no. 6, pp. 1089–1103, Mar. 2007, doi: 10.1016/j.cell.2007.01.043.

[158] T. Kiuchi *et al.*, "A single female-specific piRNA is the primary determiner of sex in the silkworm," *Nature*, vol. 509, no. 7502, pp. 633–636, 2014, doi: 10.1038/nature13315.

[159] Y.-W. Chen *et al.*, "Systematic study of Drosophila microRNA functions using a collection of targeted knockout mutations.," *Dev. Cell*, vol. 31, no. 6, pp. 784–800, Dec. 2014, doi: 10.1016/j.devcel.2014.11.029.

[160] S. S. Joshi and V. H. Meller, "Satellite Repeats Identify X Chromatin for Dosage Compensation in Drosophila melanogaster Males," *Curr. Biol.*, vol. 27, no. 10, pp. 1393-1402.e2, 2017, doi: 10.1016/j.cub.2017.03.078.

[161] J. M. Fingerhut, J. V Moran, and Y. M. Yamashita, "Satellite DNA-containing gigantic introns in a unique gene expression program during Drosophila spermatogenesis.," *PLoS Genet.*, vol. 15, no. 5, p. e1008028, May 2019, doi: 10.1371/journal.pgen.1008028.

[162] Z. Teng *et al.*, "Topoisomerase I is an evolutionarily conserved key regulator for satellite DNA transcription," *Nat. Commun.*, vol. 15, no. 1, 2024, doi: 10.1038/s41467-024-49567-5.

[163] A. V Probst, I. Okamoto, M. Casanova, F. El Marjou, P. Le Baccon, and G. Almouzni, "A strand-specific burst in transcription of pericentric satellites is required for chromocenter formation and early mouse development.," *Dev. Cell*, vol. 19, no. 4, pp. 625–638, Oct. 2010, doi: 10.1016/j.devcel.2010.09.002.

8. Summary

Satellite DNAs (satDNAs) are tandemly repeated DNA sequence and one of the most abundant repeated sequences. They are the fastest evolving part of the eukaryotic genome. So far, studies have mainly focused on satDNAs in centromeric heterochromatin. Although there is clear evidence that satDNAs have been assigned some roles, especially in centromere structure, the understanding of their organization, their evolutionary dynamics and the molecular mechanisms that drive their spread across the genome, especially in euchromatic regions, is still quite limited. In *Tribolium castaneum*, a species known for its abundance of satDNAs, the existing reference genome assembly, Tcas5.2, is reported to lack more than 25% of the estimated genome size and the repetitive satDNA regions are significantly underrepresenting. To generate a new, the most contiguous genome assembly using Oxford Nanopore (ONT) sequencing, a new protocol for high-molecular-weight (HMW) DNA isolation was developed. The new chromosome-level genome assembly was generated by combining Nanopore long-read sequencing data and a reference-guided assembly approach which was highly efficient in bridging highly repetitive regions in *T. castaneum*. The new TcasONT assembly was generated from 89 Gb of ONT data, spanning 191 Mb of the estimated 204 Mb genome and capturing 47.8 Mb of repetitive elements, including 24.3 Mb of satDNAs—a remarkable 20-fold increase in representation. The enhanced genome assembly provided an exceptional platform for in-depth genome-wide analyses of ten different and the most abundant satDNAs in euchromatin, Cast1-Cast9. Our genomic analyzes revealed that contrary to common assumptions, satDNAs are abundant in gene-rich regions, including long arrays and rarely overlap with transposons.. Based on the integration of the results of principal component analysis of monomer variation and sequence relationships between arrays, we proposed the most plausible scenario of genome dynamics of euchromatic Cast satDNAs in the *T. castaneum* genome. These scenarios involve alternating cycles of dramatic expansion from one or more centers involving intra- and interchromosomal spread, followed by a cycle characterized by process sequence divergence and elongation of satDNA arrays. Comparative analyses of satDNA arrays, surrounding regions and their junctions reveal efficient self-propagation mechanism that operates at the inter- and intra-chromosome level. Analyses of arrays' neighboring regions showed a tendency of one Cast satDNA to be associated with transposable-like elements. In addition, the experimental evidence suggests also role of extrachromosomal circular DNA (ecc DNA) in this extensive satDNA exchange. It can be proposed that satDNA spread occurs via transposition by

transposable elements and eccDNA-mediated insertion. Considering the effects of recombination on the spread of euchromatic satDNA, the results show that suppressed recombination has less impact on the dynamics of satDNA array exchange, but has effects on the length of satDNA arrays stimulating the longer arrays. We proposed that the demonstrated extensive genome dynamics of satDNAs in gene-rich regions implies their potential effects on gene expression and regulation. The expression of these euchromatic satDNAs during embryogenesis and brain development was also analyzed. The results show that of the 10 euchromatic satDNAs analyzed, three were transcribed and processed into small RNAs. Two of them showed differential transcription with peaks in the early blastoderm during embryogenesis and in the female pupal brain. In embryogenesis, transcripts are processed into both miRNAs and piRNAs, whereas transcripts in the brain were exclusively processed into miRNAs. The absence of other genomic Cast-specific small RNAs targets suggests that the processed RNAs probably play a role exclusively in a self-regulatory mechanism. Additionally, variations in the transcription locations of RNAs from satDNA monomers coupled with different lengths of small RNA fragments pointing to the structural and functional roles of satDNAs, highlighting their significant yet underappreciated influence on genome function and evolution.

9. Sažetak

Satelitske DNA (satDNA) su tandemski ponovljene sekvence DNA i jedna od najzastupljenijih ponovljenih sekvenci. Oni su dio eukariotskog genoma koji se najbrže mijenja. Do sada su se studije uglavnom fokusirale na satDNA u centromernom heterokromatinu. Iako postoje jasni dokazi da satDNA imaju neke uloge u genomu, posebno u strukturi centromera, razumijevanje njihove organizacije, evolucijske dinamike i molekularnih mehanizama koji pokreću njihovo širenje po genomu, posebno u eukromatskim regijama, još uvijek je prilično ograničeno. Kod vrste *Tribolium castaneum*, koja je poznata po značajnom udjelu satDNA, pokazalo se da postojeći referentni genomski sklopu, Tcas5.2, ima nedostatak više od 25% procijenjene veličine genoma, a među ostalima satDNA regije su značajno podzastupljene. Kako bi se kreirao novi, najkontinuiraniji genomski sklop korištenjem Oxford Nanopore (ONT) tehnologije sekvenciranja, razvijen je novi protokol za izolaciju DNA visoke molekularne težine (HMW). Novi genomski sklop na razini kromosoma složen je kombinacijom Nanopore dugih očitanja i korištenja referentnog Tcas5.2 genoma, što se pokazalo iznimno učinkovitom metodom u premošćivanju visoko repetitivnih genomskih regija kod *T. castaneum*. Novi TcasONT sklop sastavljen je od 89 Gb ONT podataka, obuhvaćajući 191 Mb od procijenjenih 204 Mb genoma i 47.8 Mb repetitivnih elemenata, uključujući 24.3 Mb satDNA—što je značajno povećanje u zastupljenosti satDNA, 20 puta veće. Poboljšani genomski sklop, TcasONT pružio je platformu za detaljne analize deset različitih i najzastupljenijih satDNA u eukromatinu, Cast1-Cast9. Genomske analize otkrile su da, suprotno važećim hipotezama, satDNA su zastupljene i u regijama bogatim genima, uključujući i jako duge nizove, te se regije satDNA rijetko preklapaju s transpozonima. Na temelju integracije rezultata analize varijacije monomera i srodnosti sekvenci između nizova, predložili smo najvjerojatniji scenarij genomske dinamike eukromatskih Cast satDNA u genomu *T. castaneum*. Ovaj scenariji uključuje izmjenične cikluse dramatične ekspanzije satDNA iz jednog ili više centara koji uključuju unutar-kromosomsko i među-kromosomsko širenje, nakon čega slijedi ciklus karakteriziran procesom divergencije satDNA sekvence i produljenja nizova. Komparativne analize nizova satDNA, okolnih regija i njihovih insercijskih mjesta otkrivaju učinkovit mehanizam samoširenja koji djeluje na inter- i intra-kromosomskoj razini. Analize susjednih regija nizova pokazale su tendenciju da se jedna Cast satDNA povezuje s transpozonskim elementima. Osim toga, eksperimentalni dokazi također sugeriraju ulogu ekstrakromosomalne kružne DNA (eccDNA) u opsežnoj propagaciji satDNA. Stoga, je moguće predložiti da se širenje satDNA događa putem transpozicije i eccDNA posredovanog umetanja. S

obzirom na učinke rekombinacije na širenje eukromatskih satDNA, rezultati pokazuju da smanjena rekombinacija ima manji utjecaj na dinamiku širenja nizova satDNA, ali pozitivno utječe na povećanje duljine nizova satDNA. Predlažemo da značajna dinamika satDNA u regijama bogatim genima implicira njihove potencijalne učinke na ekspresiju i regulaciju gena. Također je analizirana ekspresija eukromatskih satDNA tijekom embrionalnog razvoja i razvoja mozga. Rezultati pokazuju da su od deset analiziranih Cast eukromatskih satDNA, tri bile transkribirane i obrađena u male RNA. Dvije od njih pokazale su diferencijalnu transkripciju s povećanjima u ranoj blastodermi tijekom embrionalnog razvoja i u mozgu ženskih pupa. Tijekom embrionalnog razvoja, transkripti se procesiraju u miRNA i piRNA, dok su transkripti u mozgu ekskluzivno procesirani u miRNA. Izostanak sekvenci sličnih Cast-specifičnim malim RNA u genomu sugerira da procesirane Cast derivirane RNA vjerojatno igraju isključivu ulogu u mehanizmu samoregulacije. Uz to, varijacije u transkripcijskom profilu monomera satDNA s obzirom na RNA, zajedno s različitim duljinama fragmenata male RNA, ukazuju na strukturalne i funkcionalne uloge satDNA, ističući njihov značajan, ali nedovoljno istražen utjecaj na funkciju i evoluciju genoma.

10. Curriculum vitae

Marin Volarić

Researcher Identification Number (MBZ): 393936

Contact:

Email	mvolaric@irb.hr
Telephone	(385)098/949-0394
Adress	Slave Raškaj 22, Karlovac, Croatia

Education

2020-	**Josip Juraj Strossmayer University of Osijek**, Osijek, Croatia. University Postgraduate Interdisciplinary Doctoral Study, Molecular Biosciences Mentor: Nevenka Meštrović, PhD
2018-2020	**University of Zagreb**, Faculty of Science, Croatia Master studies in Molecular Biology, diploma cum laude (4.574) Master thesis: "Selection of the most informative genomic regions for the determination of melanoma cell-of-origin using machine learning methods" Mentor: Rosa Karlić, PhD
2015-2018	**University of Zagreb**, Faculty of Science, Croatia. Bachelor studies in Molecular Biology, diploma cum laude (4.559)
2012-2015	**Natural Science and Mathematics High School**, Karlovac, Croatia

Professional Experience

2020-	**Research Assistant** Ruđer Bošković Institute, Zagreb, Croatia.
2024	**Laboratory skill training** University of Gottingen Department of Evolutionary Developmental Genetics
2018	**Laboratory skill training** University of Zagreb, Faculty of Science Department of Molecular biology, Zagreb, Croatia
2022-	**CEO & Founder** Solved Game d.o.o. A football data analysis company.

Publications

1. **Volarić, Marin**; Despot-Slade, Evelin; Veseljak, Damira; Mravinac, Brankica; Meštrović, Nevenka Long-read genome assembly of the insect model organism *Tribolium castaneum* reveals spread of satellite DNA in gene-rich regions by recurrent burst events // Genome Research **(IN PRESS)**

2. **Volarić, Marin**; Despot-Slade, Evelin; Veseljak, Damira; Pavlek, Martina; Vojvoda Zeljko, Tanja; Mravinac, Brankica; Meštrović, Nevenka
The Genome Organization of 5S rRNA Genes in the Model Organism *Tribolium castaneum* and Its Sibling Species Tribolium freemani // Genes, 15 (2024), 6; 776, 11. doi: 10.3390/genes15060776

3. Gržan, Tena ; Dombi, Mira ; Despot-Slade, Evelin ; Veseljak, Damira ; **Volarić, Marin** ; Meštrović, Nevenka ; Plohl, Miroslav ; Mravinac, Brankica
The Low-Copy-Number Satellite DNAs of the Model Beetle *Tribolium castaneum* // Genes, 14 (2023), 5; 999, 21. doi: 10.3390/genes14050999

4. Cervena, Klara; Siskova, Anna; Jungwirth, Jiri; Volarić, Marin; Kral, Jan; Kohout, Pavel; Levy, Miroslav; Vymetalkova, Veronika
MALAT1 in Liquid Biopsy: The Diagnostic and Prognostic Promise for Colorectal Cancer and Adenomas? // International journal of general medicine, 16 (2023), 3517-3531. doi: 10.2147/IJGM.S420127

5. Volarić, Marin ; Despot-Slade, Evelin ; Veseljak, Damira ; Meštrović, Nevenka ; Mravinac, Brankica
Reference-Guided De Novo Genome Assembly of the Flour Beetle *Tribolium freemani* // International journal of molecular sciences, 23 (2022), 11; 5869, 18. doi: 10.3390/ijms23115869

6. Volarić, Marin ; Veseljak, Damira ; Mravinac, Brankica ; Meštrović, Nevenka ; Despot-Slade, Evelin
Isolation of High Molecular Weight DNA from the Model Beetle Tribolium for Nanopore Sequencing // Genes, 12 (2021), 8; 1114, 12. doi: 10.3390/genes12081114

Conference abstracts

1. Despot-Slade, Evelin; Volarić, Marin; Mravinac, Brankica; Meštrović, Nevenka
Epigenetic regulation of repetitive DNA in insect *Tribolium castaneum* // International congress on transposable elements 2024 : Abstract book.
Saint Malo: ICTE, 2024. str. 76-76

2. Veseljak, Damira; Volarić, Marin; Despot-Slade, Evelin; Meštrović, Nevenka; Mravinac, Brankica
The genomes of Tribolium sibling species framed by the evolution of satellite DNAs // Program and Abstracts, Arthropod Satellite Meeting, Helsinki 2024 / Chipman, Ariel; El-Sherif, Ezzat; van der Zee, Maurijn et al. (ur.).
Helsinki: EED organizin committees, 2024. str. 31-31

3. Veseljak, Damira; Despot-Slade, Evelin; Volarić, Marin; Meštrović, Nevenka; Mravinac, Brankica
Satellitomes of flour beetles from the genus *Tribolium*: an evolutionary perspective // Euro EvoDevo 2004 Programme Book / Kratochwil, Claudius (ur.).
Helsinki: EED organizing committees, 2024. str. 567-567

4. Volarić, Marin; Despot-Slade, Evelin; Meštrović, Nevenka; Mravinac, Brankica; Veseljak, Damira
Oxford Nanopore Sequencing reveals complex mechanisms of repetitive DNA propagation in Tribolum castaneum // International congress on transposable elements 2024 : Abstract book.
Saint Malo: ICTE, 2024. str. 154-154

5. Veseljak, Damira; Despot-Slade, Evelin; Volarić, Marin; Meštrović, Nevenka; Mravinac, Brankica
Dynamic evolution of satellite DNAs drastically alters genomes of Tribolium sibling species // Abstract Book: the Evolution of Animal Genomes.

European Molecular Biology Organization (EMBO), 2023. str. 148-148

6. Despot-Slade, Evelin ; **Volarić, Marin** ; Meštrović, Nevenka
Transcriptomics of euchromatic satellite DNAs in embryogenesis and development // Epigenome inheritance and reprogramming in health and disease : Abstract book.
2022. str. 17-17

7. **Volarić, Marin** ; Veseljak, Damira ; Mravinac, Brankica ; Meštrović, Nevenka ; Despot-Slade, Evelin
Nanopore sekvenciranje kukaca roda tribolium s tvrdim egzoskeletom // 6. simpozij studenata doktorskih studija PMF-a : knjiga sažetaka = 6th Faculty of Science PhD student symposium : book of abstracts.
Zagreb: Prirodoslovno-matematički fakultet Sveučilišta u Zagrebu, 2022. str. 238-239

8. **Volarić, Marin** ; Despot-Slade, Evelin ; Meštrović, Nevenka
Nanopore based analyses of genome-wide DNA methylation profiles through Tribolium castaneum development // Epigenome inheritance and reprogramming in health and disease : Abstract book.
2022. str. 22-22

9. **Volarić, Marin**; Despot-Slade, Evelin; Meštrović, Nevenka
Preliminary analyses of genome-wide DNA methylation profiles through the Tribolium castaneum development using nanopore long reads // Chromatin Structure and Function - GRC Poster List.
2022. str. 11-11

10. **Volarić, Marin** ; Despot-Slade, Evelin ; Meštrović, Nevenka
Long-range organisation of holocentromeres // Simpozij studenata doktorskih studija PMF-a : knjiga sažetaka = PhD student symposium 2021 : book of abstracts.
Zagreb: Prirodoslovno-matematički fakultet Sveučilišta u Zagrebu, 2021. str. 267-268

Other Competencies

Programming Languages	R	High level of competency for data analysis and visualization, including most vital data processing libraries such as **data.table**, the entire **tidyverse**, **ggplot2**
		High proficiency in biology specific libraries such as **Biostrings** and **GenomicRanges**

	Python	High level of competency with standard data processing pipelines, including **pandas**, **matplotlib** and **polars**.
		Additionally proficient and have built applications in web development frameworks such as **Flask** and **FastAPI,** also experience with common database management frameworks such as **SQLAlchemy**
	Rust	Proficient in building standalone CLI apps, as well as python integration using **maturin** and web assembly using **wasmpack**
	Shell/CLI	High level of proficiency in developing and using shell scripts for automated pipelines, additionally proficient in developing and using containers like **Docker** and **Singularity** (**Apptainer**)
High Performance computing	Proficient in writing, automating and submitting jobs to **SGE** and **PBS pro** HPC cluster arrays as both standalone and containerized applications	
Cloud computing	Amazon web services development and deployment, high proficiency in Amazon **S3**, **EC2**, **RDS** and **Route 53** services with production experience. Successfully deployed 2 standalone applications.	
Laboratory expertise	DNA isolation and gel electrophoresis	
	RNA isolation	
	Protein isolation and western blotting	
	DNA and RNA sequencing and read analysis	
	ChIP sequening and read analysis	
	Oxford Nanopore sequencing and read analysis	
	Oxford Nanopore methylation analysis	
	PCR	
	Bacterial plateing and cloning	

Open source projects

Project	Description
https://github.com/mvolar/SatXplor/	A satDNA analysis pipeline.
https://github.com/mvolar/melanoma_random_forest	Repository containing code and graphs for PCA analysis coupled with Random forest predictions of melanoma mutations and cell of origin.
https://github.com/mvolar/tcasont_assembly	This repository contains the necessary scripts to recreate the visualizations presented in the research paper *Long-read genome assembly of the insect model organism Tribolium castaneum reveals spread of satellite DNA in gene-rich regions by recurrent burst events.*
https://github.com/mvolar/R-binance-trading-bot	A basic 100/50 SMA MACD Binance trading bot with R-Binance API
https://github.com/mvolar/latex_to_clipboard	A rust program which takes the clipboard last input and puts it into a wolfprham alpha API and returns the wolfram alpha result cell and decimal approximations for simple queries of different latex formulas in the clipboard.

Workshops

2022	EMBO workshop: Epigenome inheritance and reprogramming in health and disease Split, Croatia
2022	Fundamentals of Accelerated Computing with CUDA C/C++ Online
2021	Usage of Isabella high-performance cluster Online
2021	MedILS Bioinformatics School in Transcriptomics Online
2020	Winter School of Research Commercialization Online

Awards and participations

2022	STEM games, Rovinj, Croatia Mentorship in the Science Arena
2018	STEM games, Poreč Croatia 1st Place
2018	University of Zagreb, Croatia Participation in the 2018 Biology Night
2013	National competition in Biology, Šibenik, Croatia 1st Place

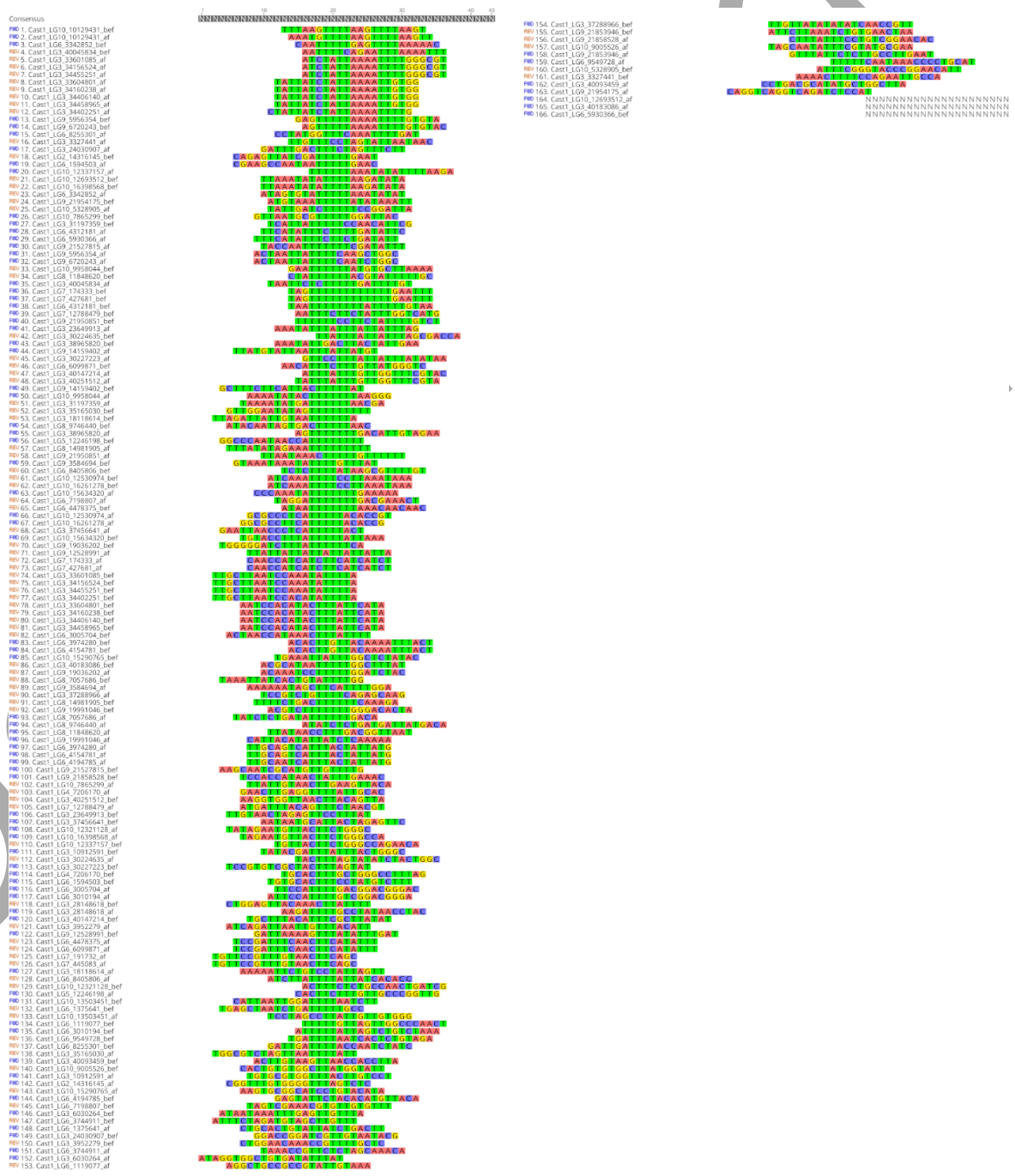
Languages

Croatian	Native Speaker
English	Speak and read/write fluently
German	Basic familiarity

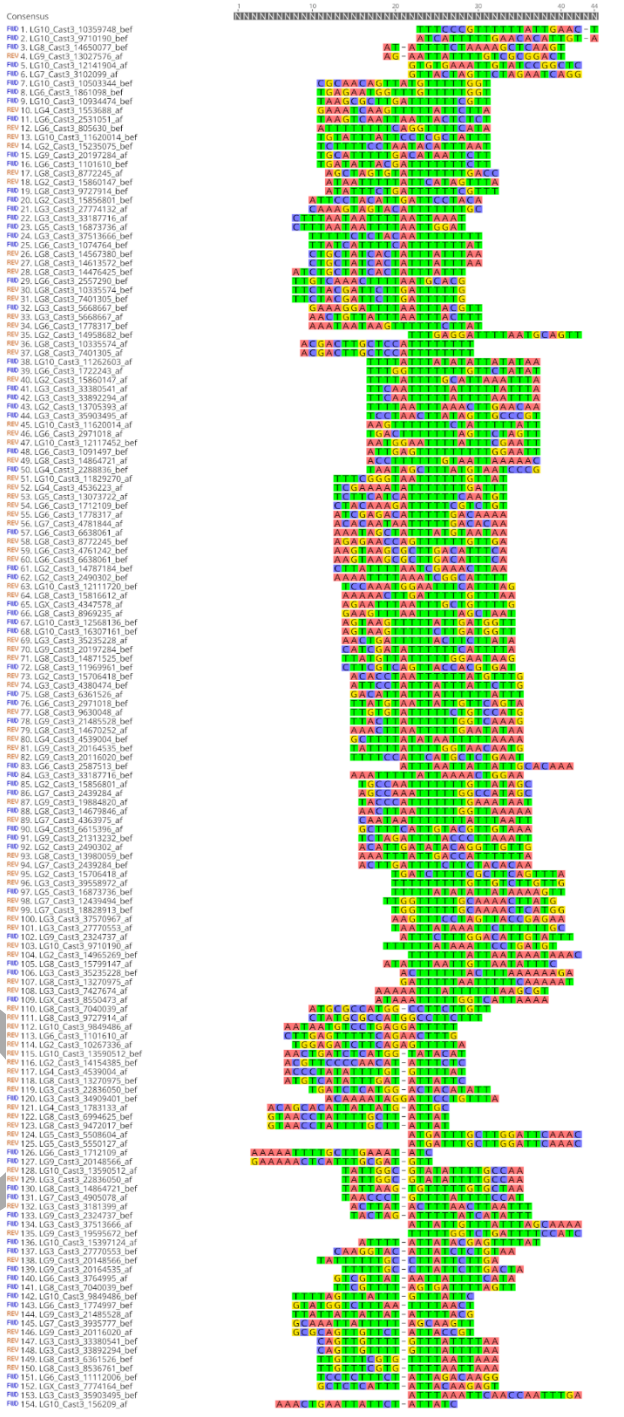
11. Supplementary material

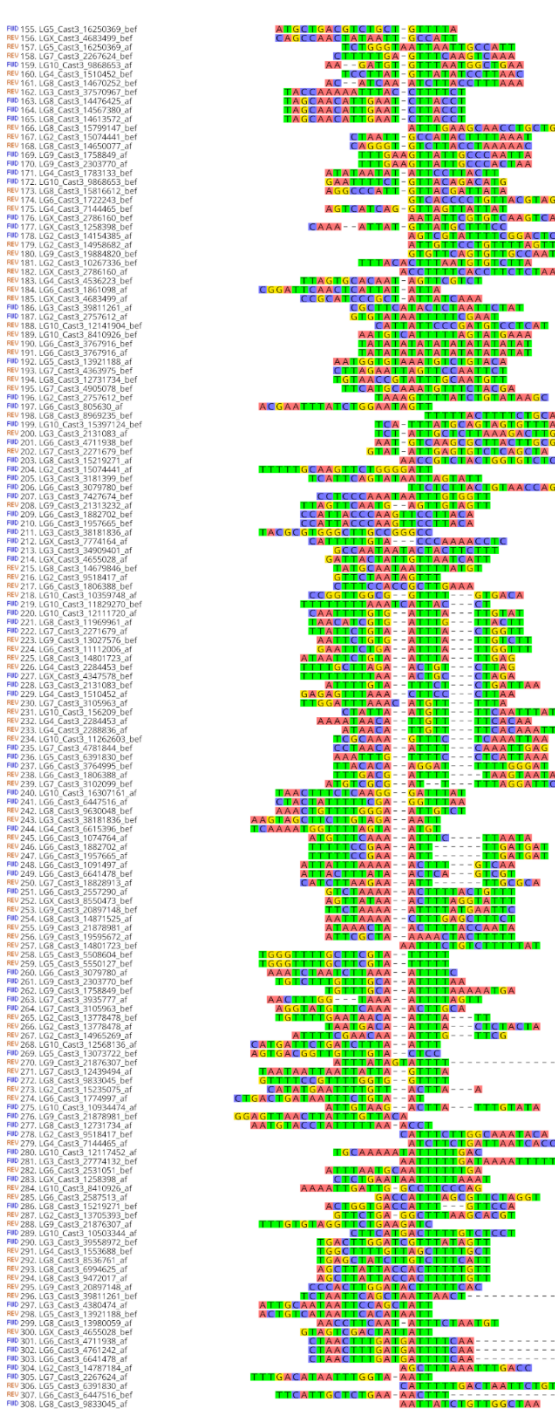
Supplementary Figures

A

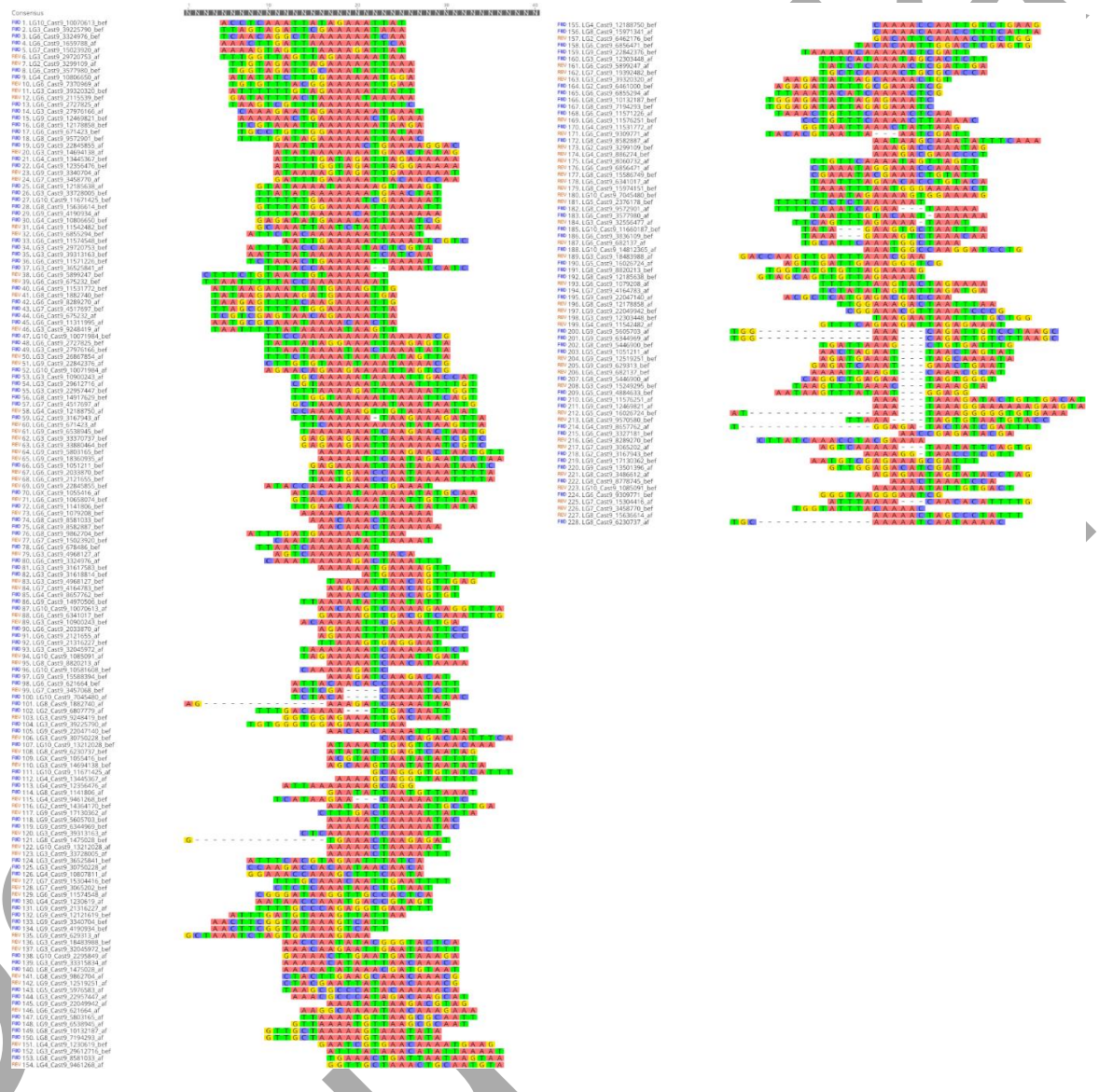


B

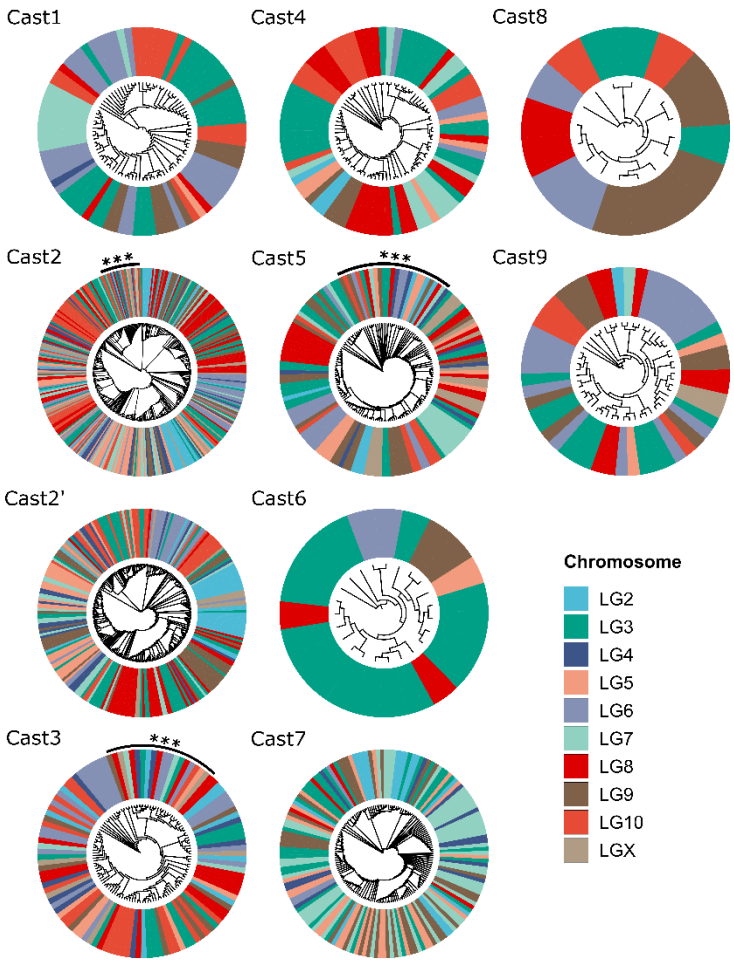




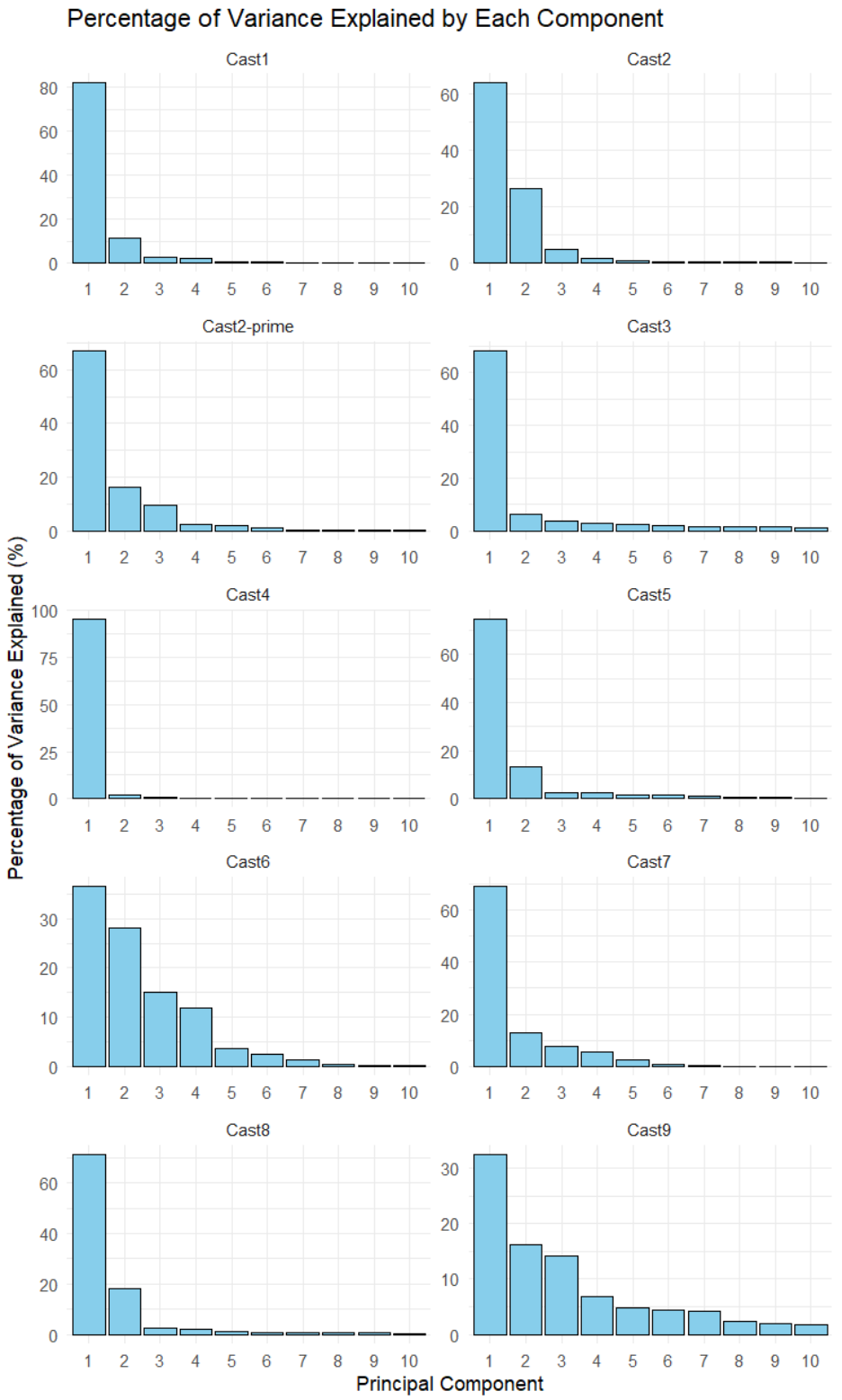
C



Supplementary Figure 1. Microhomology (20bp) alignment for **A** Cast1 **B** Cast3 and **C** Cast9 with visible enrichment of polyT and polyA stretches, with their appropriate sequence motifs shown in **Figure 17B**. Precise edges were determined using the outlined k-mer counting algorithm and then sequences consisting of 20 bp before and after each precisely defined array edges were extracted and aligned using MAFFT algorithm.

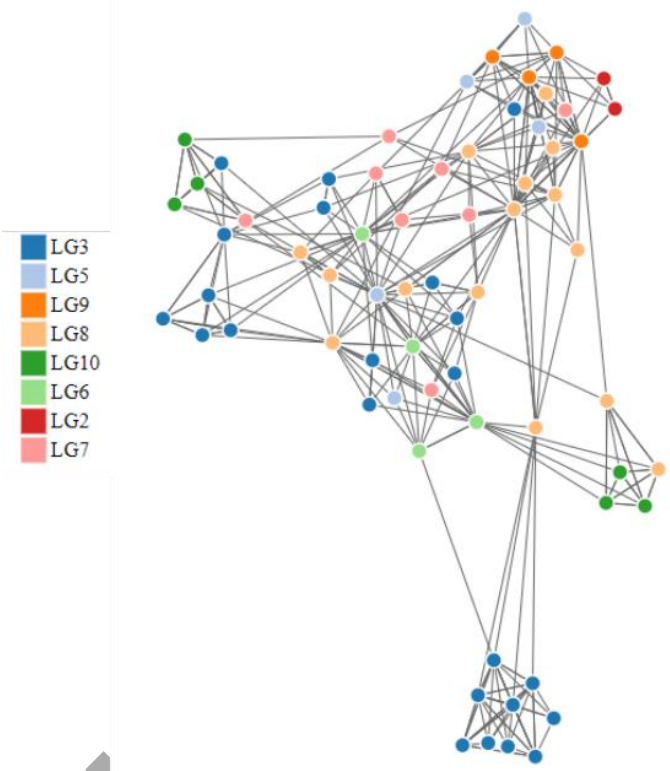


Supplementary Figure 2. Trees of array consensus monomers generated using IQ-TREE software for each satDNA family and visualized using ggtree. Since the procedure of generating array consensuses and subsequent tree creation proved inaccurate due to low bootstrap values and long execution time, a dimensionality based approach was used to gauge evolutionary relationship between monomers of each satDNA family.

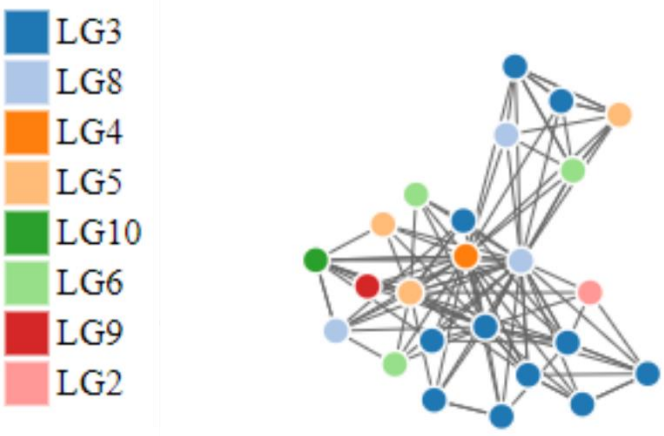


Supplementary Figure 3. Percentage of variance explained by individual principal components using FactoMineR PCA package. All Cast1-Cast9 satDNAs have > 50% of their genomic variance explained by the first 2 principal components and some like Cast1 and Cat4 even more than >90%.

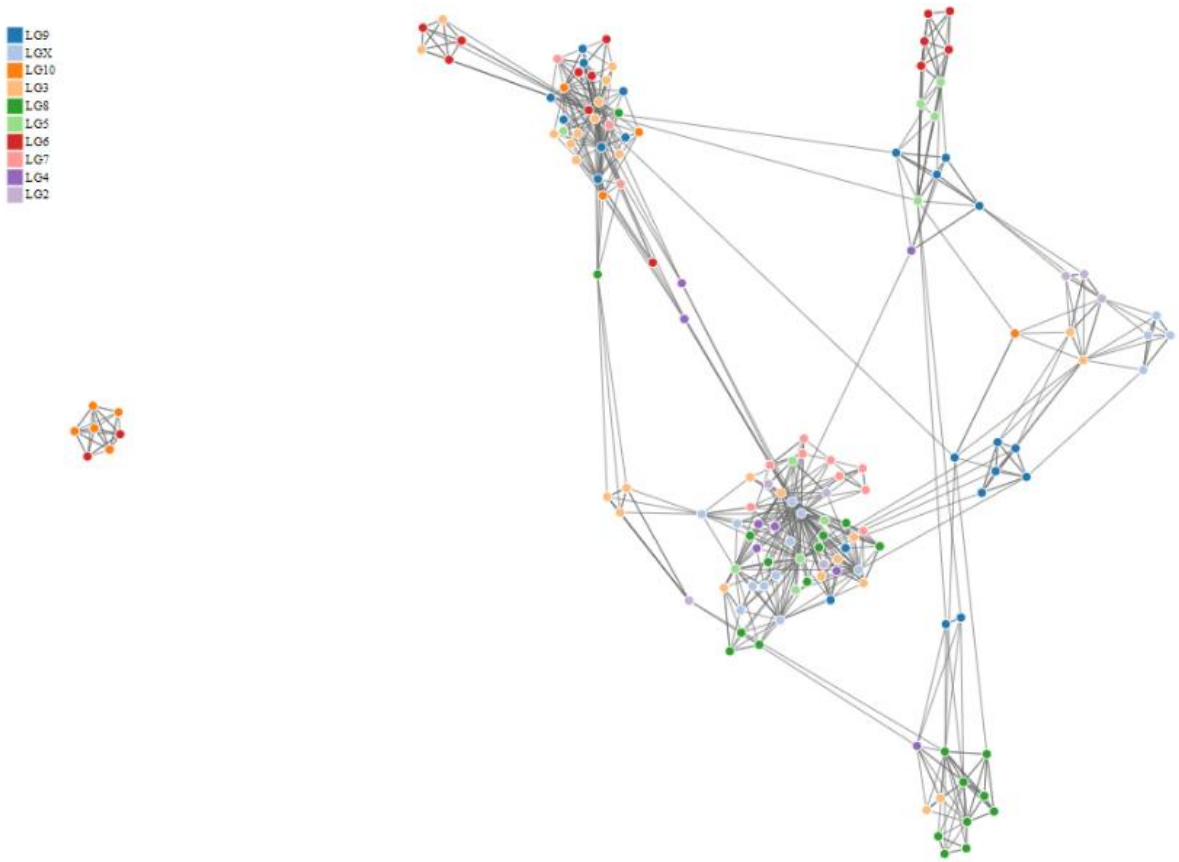
A Cast4



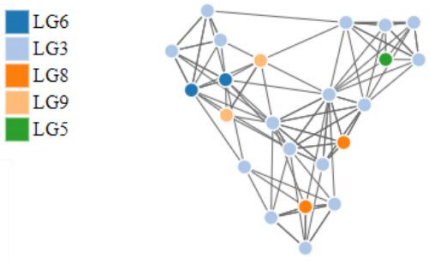
B Cast 7



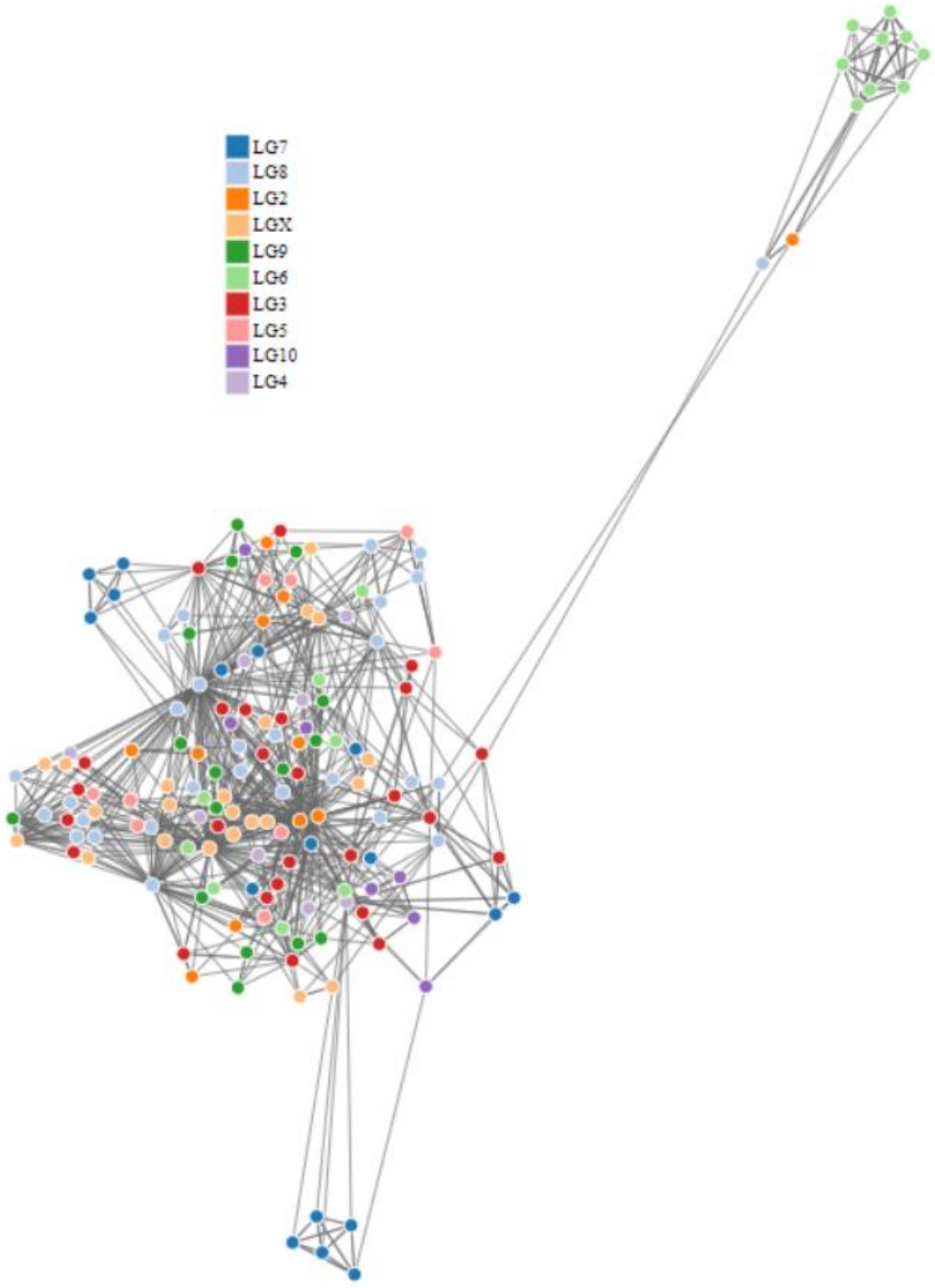
C Cast5



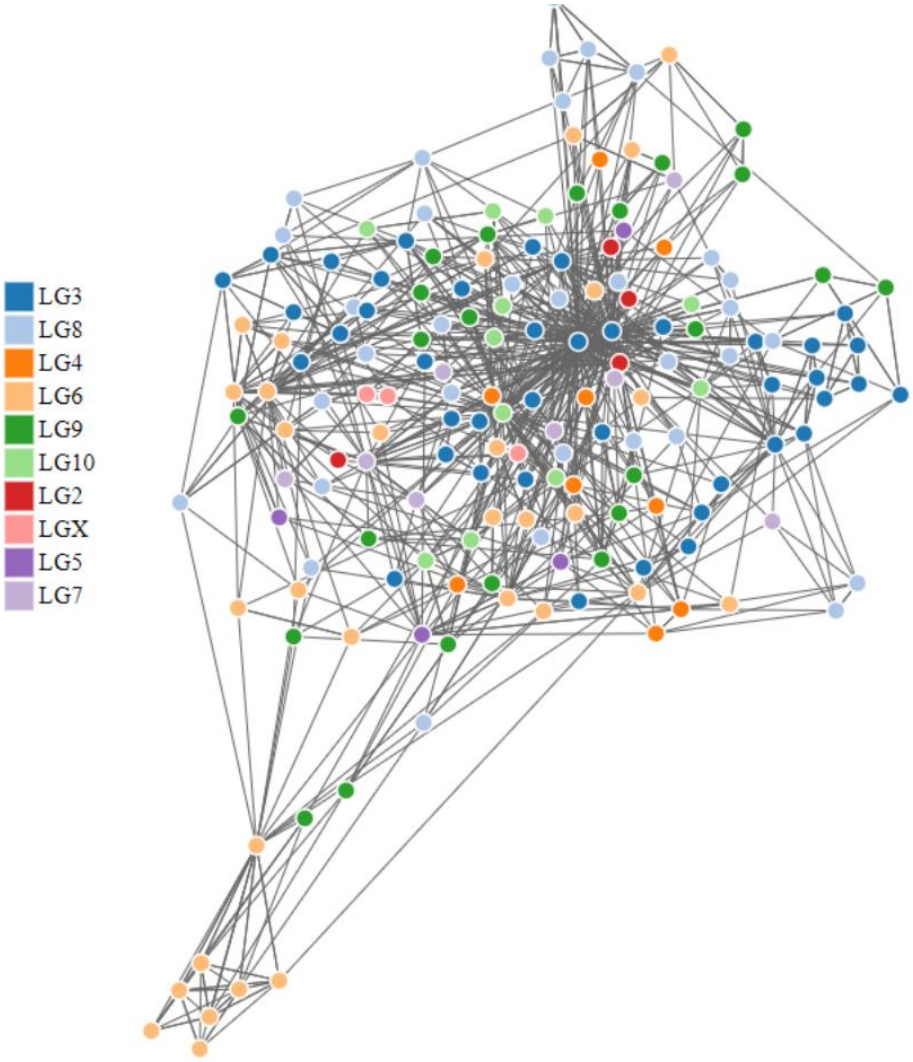
D Cast6



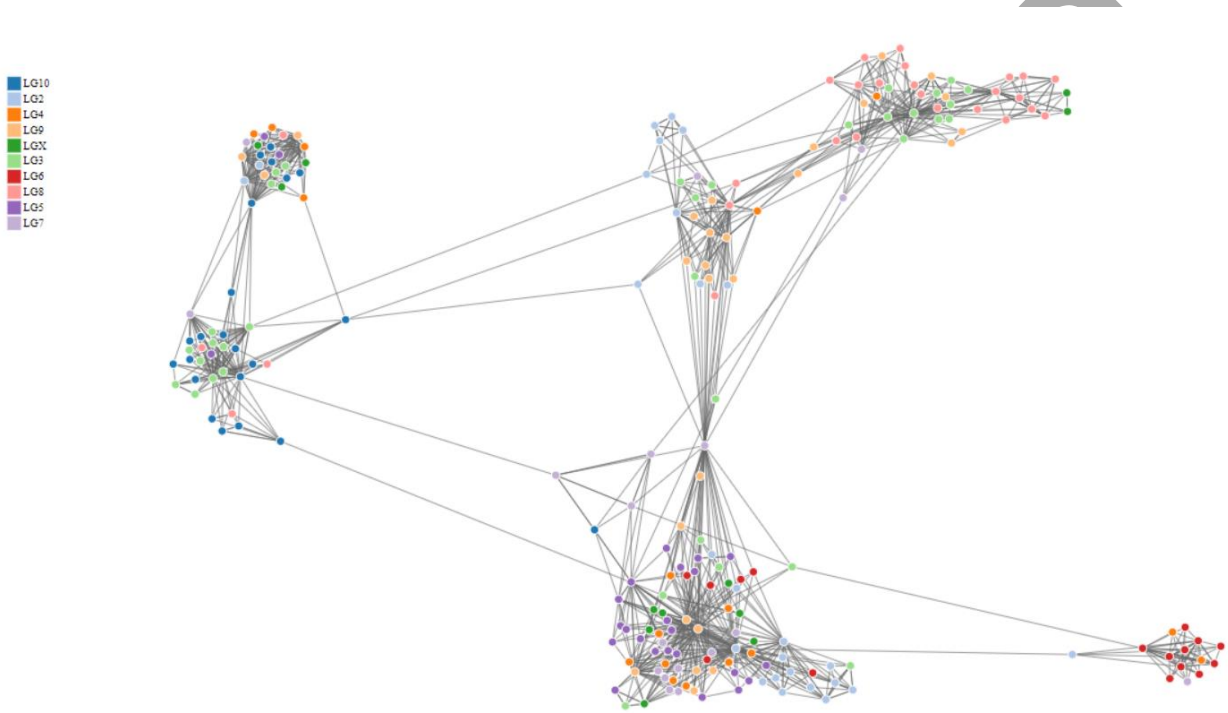
E Cast 8



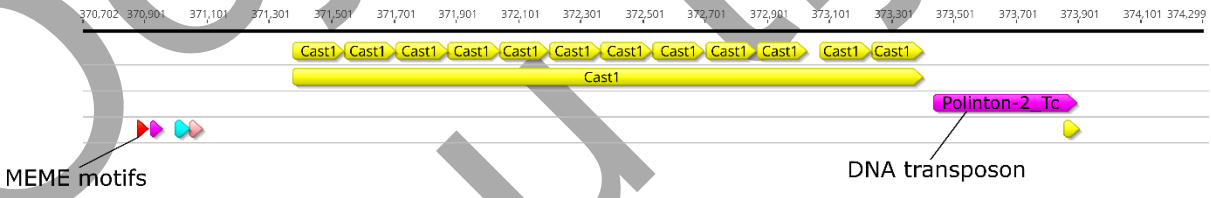
F Cast9



G Cast2’



Supplementary Figure 4. A-G Graph network visualization of the remaining Cast1-Cast9 families belonging to the three patterns.



Supplementary Figure 5. Characteristic presence of Polinton-2 sequence in the vicinity of Cast1 arrays present on LG7 chromosome.

Supplementary Figure 6. A Scaled number of arrays per satDNA family and chromosome. LGX is colored blue while the autosomes are colored red. **B** Mean width of satDNA arrays per chromosome. LGX is colored blue while the rest are colored red.

Supplementary Tables

Supplementary Table 1. *T. castaneum* genome size estimation performed by findGSE (Sun et al., 2018) and CovEST (Hozza et al., 2015). The estimations were performed on corrected reads with a k-mer size of 31.

Program	Estimated genome size (bp)
findGSE	203,772,508
CovEST RE	208,366,566
Repeat ratio	
findGSE	27%

Supplementary Table 2. Distribution of satellites and genes within the 471 contigs used for RagTag orientation as well as status of their inclusion in the final TcasONT assembly.

Contig name	*Contig length (bp)*	*Satellite occupancy (%)*	*Gene occupancy (%)*	*Contig status in final ONT assembly*
tig00001249	202992	53.31638685	0.66	Final assembly
tig00001150	64053	40.43526455	5.23	Final assembly
tig00001200	715149	38.5754577	0.29	Final assembly
tig00000106	105261	28.96134371	29.94	Final assembly
tig00000479	78317	27.68875212	8.94	Final assembly
tig00001285	844215	20.15339694	28.18	Final assembly
tig00001321	4064624	17.49544361	25.89	Final assembly
tig00001230	69590	15.53527806	34.37	Final assembly
tig00000141	163001	15.20113374	43.49	Final assembly
tig00000341	80874	15.05922793	9.13	Final assembly
tig00001247	2191239	14.38359759	53.59	Final assembly
tig00000393	181067	10.26470864	31.72	Final assembly
tig00001095	425567	8.594886352	45.13	Final assembly
tig00000368	162640	7.58669454	11.48	Final assembly
tig00000205	86465	7.549875672	6.09	Final assembly
tig00001256	16371545	6.089895609	67.54	Final assembly
tig00001106	5280288	5.599561994	67.62	Final assembly
tig00000123	76509	5.592806075	9.36	Final assembly
tig00001078	184875	5.308451657	6.01	Final assembly
tig00000080	103974	5.22342124	30.50	Final assembly
tig00000104	122215	5.058298899	74.39	Final assembly
tig00000189	133676	4.869984141	51.70	Final assembly
tig00000380	161644	4.385563337	15.14	Final assembly

tig00000183	99041	3.757029917	38.83	Final assembly
tig00000389	112242	3.496908466	14.76	Final assembly
tig00001843	11279182	3.306241534	66.39	Final assembly
tig00001534	1266074	3.197759373	25.67	Final assembly
tig00000053	72083	3.047875366	41.38	Final assembly
tig00000097	108760	2.853990438	6.48	Final assembly
tig00001171	13335735	2.757440816	72.62	Final assembly
tig00001098	6643062	2.688248281	69.19	Final assembly
tig00001229	2431020	2.432188958	67.60	Final assembly
tig00001148	1963873	2.412426873	80.78	Final assembly
tig00000435	146430	1.981834324	3.75	Final assembly
tig00001108	5621849	1.779041024	63.68	Final assembly
tig00000391	4986065	1.682428929	62.48	Final assembly
tig00001145	6796095	1.674976586	58.64	Final assembly
tig00001172	61547	1.511040343	22.90	Final assembly
tig00000115	103718	1.496365144	11.02	Final assembly
tig00000263	109438	1.419068331	7.44	Final assembly
tig00000096	112155	1.391823815	31.32	Final assembly
tig00001107	3660290	1.36565682	76.46	Final assembly
tig00001245	2242019	1.356143726	55.88	Final assembly
tig00000352	7546636	1.344426841	75.57	Final assembly
tig00000385	117287	1.321544587	47.30	Final assembly
tig00001119	4920069	1.256994567	71.46	Final assembly
tig00000369	209279	1.226592252	34.72	Final assembly
tig00001118	1742344	1.143517009	63.20	Final assembly
tig00001154	6364958	0.845944309	72.66	Final assembly
tig00000078	127584	0.75166165	6.40	Final assembly
tig00001130	5488169	0.720422421	64.89	Final assembly
tig00001244	12783081	0.66976811	75.97	Final assembly
tig00001081	298804	0.63151765	59.27	Final assembly
tig00001159	1577151	0.59734293	69.28	Final assembly
tig00001083	2145271	0.527858718	61.27	Final assembly
tig00001246	2296785	0.493428858	69.08	Final assembly
tig00001082	3520765	0.304621297	60.21	Final assembly
tig00001092	922934	0.281276884	63.65	Final assembly
tig00001109	906841	0.271933007	66.85	Final assembly
tig00000422	160869	0.224406194	5.96	Final assembly
tig00001117	835545	0.172941015	58.78	Final assembly
tig00000221	218032	0.149060688	2.16	Final assembly
tig00001228	946280	0.113497062	34.33	Final assembly
tig00001079	2223311	0.064048619	46.75	Final assembly
tig00001076	1736555	0.063689316	73.37	Final assembly
tig00000004	407910	0	0.64	Final assembly
tig00000012	88187	0	2.05	Final assembly
tig00000036	82102	0	20.90	Final assembly

tig00000040	119118	0	1.44	Final assembly
tig00000041	89542	0	85.43	Final assembly
tig00000047	124412	0	3.58	Final assembly
tig00000055	181673	0	63.27	Final assembly
tig00000060	116916	0	23.46	Final assembly
tig00000063	146662	0	31.16	Final assembly
tig00000068	97769	0	74.41	Final assembly
tig00000069	117305	0	17.31	Final assembly
tig00000071	83113	0	2.02	Final assembly
tig00000072	176914	0	67.02	Final assembly
tig00000077	72879	0	2.10	Final assembly
tig00000087	85951	0	13.08	Final assembly
tig00000103	98551	0	14.84	Final assembly
tig00000108	83472	0	20.82	Final assembly
tig00000111	103724	0	38.01	Final assembly
tig00000119	92506	0	6.42	Final assembly
tig00000120	82069	0	16.76	Final assembly
tig00000121	82378	0	52.44	Final assembly
tig00000122	69710	0	10.81	Final assembly
tig00000124	84038	0	22.77	Final assembly
tig00000125	84374	0	37.01	Final assembly
tig00000135	101214	0	33.15	Final assembly
tig00000138	104205	0	9.44	Final assembly
tig00000140	74231	0	70.30	Final assembly
tig00000142	104908	0	2.83	Final assembly
tig00000149	142973	0	2.74	Final assembly
tig00000154	248946	0	1.20	Final assembly
tig00000160	123490	0	27.57	Final assembly
tig00000163	100238	0	17.10	Final assembly
tig00000164	96690	0	27.23	Final assembly
tig00000166	93915	0	45.95	Final assembly
tig00000171	109996	0	14.39	Final assembly
tig00000172	73824	0	5.97	Final assembly
tig00000178	67454	0	24.07	Final assembly
tig00000192	69844	0	6.51	Final assembly
tig00000193	81321	0	4.69	Final assembly
tig00000194	82855	0	20.60	Final assembly
tig00000202	167080	0	26.87	Final assembly
tig00000209	94285	0	34.04	Final assembly
tig00000210	185399	0	0.78	Final assembly
tig00000223	113944	0	2.74	Final assembly
tig00000225	61969	0	16.21	Final assembly
tig00000228	206185	0	1.32	Final assembly
tig00000230	1112037	0	70.01	Final assembly
tig00000231	78963	0	91.30	Final assembly

tig00000236	115144	0	43.51	Final assembly
tig00000242	93674	0	17.81	Final assembly
tig00000246	86573	0	40.71	Final assembly
tig00000255	79030	0	17.35	Final assembly
tig00000256	89639	0	10.41	Final assembly
tig00000269	81152	0	4.47	Final assembly
tig00000273	93872	0	15.53	Final assembly
tig00000274	130688	0	12.13	Final assembly
tig00000275	87741	0	41.59	Final assembly
tig00000282	104692	0	41.49	Final assembly
tig00000284	295212	0	1.61	Final assembly
tig00000290	117884	0	32.68	Final assembly
tig00000293	78469	0	22.61	Final assembly
tig00000294	166519	0	23.23	Final assembly
tig00000296	83120	0	45.73	Final assembly
tig00000298	130337	0	7.54	Final assembly
tig00000300	99488	0	57.72	Final assembly
tig00000301	88416	0	13.89	Final assembly
tig00000302	187388	0	33.04	Final assembly
tig00000306	129249	0	5.14	Final assembly
tig00000311	198376	0	16.70	Final assembly
tig00000312	142779	0	32.69	Final assembly
tig00000313	77417	0	55.93	Final assembly
tig00000317	593677	0	85.18	Final assembly
tig00000318	111523	0	49.87	Final assembly
tig00000324	63667	0	51.71	Final assembly
tig00000326	80794	0	6.05	Final assembly
tig00000328	87578	0	8.15	Final assembly
tig00000340	74499	0	48.63	Final assembly
tig00000344	113023	0	1.78	Final assembly
tig00000349	77849	0	5.62	Final assembly
tig00000350	143005	0	38.17	Final assembly
tig00000351	85428	0	49.03	Final assembly
tig00000354	73098	0	61.72	Final assembly
tig00000355	88252	0	4.10	Final assembly
tig00000356	79423	0	19.78	Final assembly
tig00000364	95487	0	28.95	Final assembly
tig00000366	93738	0	4.09	Final assembly
tig00000370	82474	0	20.97	Final assembly
tig00000372	81613	0	1.94	Final assembly
tig00000374	78223	0	18.30	Final assembly
tig00000375	75926	0	4.82	Final assembly
tig00000377	98562	0	19.51	Final assembly
tig00000383	78729	0	3.09	Final assembly
tig00000386	106469	0	20.17	Final assembly

tig00000387	112678	0	47.14	Final assembly
tig00000390	127157	0	13.29	Final assembly
tig00000394	104747	0	39.03	Final assembly
tig00000396	115591	0	7.52	Final assembly
tig00000397	73821	0	9.65	Final assembly
tig00000399	98043	0	7.83	Final assembly
tig00000402	99939	0	15.88	Final assembly
tig00000403	138167	0	1.06	Final assembly
tig00000410	84497	0	12.43	Final assembly
tig00000411	76132	0	21.17	Final assembly
tig00000412	76762	0	12.24	Final assembly
tig00000416	87565	0	4.96	Final assembly
tig00000418	88189	0	4.75	Final assembly
tig00000460	801972	0	8.13	Final assembly
tig00000461	162033	0	33.10	Final assembly
tig00000464	123410	0	59.97	Final assembly
tig00000473	114221	0	4.07	Final assembly
tig00000474	98592	0	17.08	Final assembly
tig00000477	67705	0	29.57	Final assembly
tig00000478	77005	0	1.53	Final assembly
tig00000481	163715	0	20.13	Final assembly
tig00000483	104527	0	20.03	Final assembly
tig00000488	109881	0	44.31	Final assembly
tig00000489	94010	0	35.54	Final assembly
tig00000495	56730	0	8.24	Final assembly
tig00000499	104008	0	1.00	Final assembly
tig00000502	86677	0	12.73	Final assembly
tig00000503	122566	0	31.69	Final assembly
tig00000513	387201	0	41.11	Final assembly
tig00000514	353143	0	36.94	Final assembly
tig00000519	540479	0	52.38	Final assembly
tig00000545	89242	0	4.48	Final assembly
tig00000630	103648	0	3.17	Final assembly
tig00000663	73154	0	3.00	Final assembly
tig00000668	109054	0	16.40	Final assembly
tig00000670	385409	0	1.72	Final assembly
tig00000672	190074	0	1.73	Final assembly
tig00001004	99481	0	2.22	Final assembly
tig00001011	69011	0	12.64	Final assembly
tig00001031	162065	0	2.54	Final assembly
tig00001074	53946	0	17.26	Final assembly
tig00001084	54700	0	14.50	Final assembly
tig00001085	1125725	0	67.99	Final assembly
tig00001086	1215532	0	54.10	Final assembly
tig00001087	774633	0	32.78	Final assembly

tig00001089	139911	0	38.58	Final assembly
tig00001110	1662152	0	60.54	Final assembly
tig00001113	871886	0	44.60	Final assembly
tig00001114	134681	0	2.82	Final assembly
tig00001115	753872	0	4.16	Final assembly
tig00001116	1149083	0	63.59	Final assembly
tig00001121	703386	0	53.84	Final assembly
tig00001137	1681977	0	23.96	Final assembly
tig00001146	104362	0	43.42	Final assembly
tig00001155	1716042	0	72.83	Final assembly
tig00001160	48355	0	14.58	Final assembly
tig00001162	45736	0	27.64	Final assembly
tig00001163	100722	0	6.49	Final assembly
tig00001164	52285	0	6.69	Final assembly
tig00001165	115092	0	7.61	Final assembly
tig00001166	85739	0	20.49	Final assembly
tig00001175	134583	0	12.24	Final assembly
tig00001241	916700	0	78.38	Final assembly
tig00001242	81485	0	5.75	Final assembly
tig00001243	211169	0	59.36	Final assembly
tig00001254	93472	0	7.84	Final assembly
tig00001262	81914	0	13.43	Final assembly
tig00001263	124962	0	21.01	Final assembly
tig00001264	93120	0	10.34	Final assembly
tig00001265	48555	0	7.77	Final assembly
tig00001272	660651	0	73.24	Final assembly
tig00001273	291379	0	48.83	Final assembly
tig00001279	905038	0	59.23	Final assembly
tig00001284	1495325	0	31.06	Final assembly
tig00001292	82812	0	6.65	Final assembly
tig00001293	87011	0	16.60	Final assembly
tig00001297	211024	0	38.30	Final assembly
tig00001306	264753	0	52.64	Final assembly
tig00001311	100434	0	1.46	Final assembly
tig00001312	75784	0	11.57	Final assembly
tig00001338	64206	0	8.52	Final assembly
tig00001440	1567878	0	67.94	Final assembly
tig00001464	336888	0	0.68	Final assembly
tig00001546	627080	0	12.24	Final assembly
tig00001806	78956	0	2.11	Final assembly
tig00001820	70295	0	2.88	Final assembly
tig00001168	107370	81.59541771	5.13	Unplaced assembly
tig00001661	116514	78.1502652	1.61	Unplaced assembly
tig00001210	176808	77.24989819	0.61	Unplaced assembly
tig00001607	237164	73.66927527	0.79	Unplaced assembly

tig00001401	319825	72.98460095	2.28	Unplaced assembly
tig00001384	158344	71.68001314	2.48	Unplaced assembly
tig00000804	640160	71.15705449	0.66	Unplaced assembly
tig00001211	67018	70.5840222	7.54	Unplaced assembly
tig00000543	75297	70.25113882	1.43	Unplaced assembly
tig00001217	517256	70.11769801	0.26	Unplaced assembly
tig00001019	170539	68.08882426	3.67	Unplaced assembly
tig00000803	141643	68.02736457	1.33	Unplaced assembly
tig00000985	245310	67.88023317	0.77	Unplaced assembly
tig00000749	146502	67.77177103	1.29	Unplaced assembly
tig00000585	651258	67.71003197	0.23	Unplaced assembly
tig00001123	375950	67.50897726	1.52	Unplaced assembly
tig00001239	612206	66.52679	0.51	Unplaced assembly
tig00001547	279211	66.29502419	0.90	Unplaced assembly
tig00001124	156082	66.19853667	0.82	Unplaced assembly
tig00001552	217326	65.10081629	3.30	Unplaced assembly
tig00001402	253796	64.6846286	1.23	Unplaced assembly
tig00001612	282124	64.59783641	1.28	Unplaced assembly
tig00001738	130346	64.06180474	1.86	Unplaced assembly
tig00001184	719039	63.81406294	0.31	Unplaced assembly
tig00001627	714629	63.74426451	0.21	Unplaced assembly
tig00001574	101642	63.35668326	2.39	Unplaced assembly
tig00000724	432183	62.77363987	0.38	Unplaced assembly
tig00001556	693092	62.67926913	0.21	Unplaced assembly
tig00000894	137426	62.03047458	2.52	Unplaced assembly
tig00001377	388594	61.88078045	0.80	Unplaced assembly
tig00000832	139487	61.59857191	6.59	Unplaced assembly
tig00000656	468470	61.37938395	0.67	Unplaced assembly
tig00001169	349173	61.12070521	0.71	Unplaced assembly
tig00001259	247931	61.03916009	0.85	Unplaced assembly
tig00001335	239086	60.819956	1.23	Unplaced assembly
tig00001180	178659	60.19903839	0.83	Unplaced assembly
tig00001361	84860	60.020033	3.67	Unplaced assembly
tig00001672	94224	59.32777212	1.53	Unplaced assembly
tig00001185	361726	58.20151164	0.76	Unplaced assembly
tig00001225	113781	56.76255262	4.08	Unplaced assembly
tig00001182	499300	55.82054877	0.66	Unplaced assembly
tig00000659	264988	55.25306806	0.71	Unplaced assembly
tig00001671	153612	54.89219592	0.88	Unplaced assembly
tig00001512	858853	53.91912236	0.49	Unplaced assembly
tig00001379	86819	53.83038275	6.96	Unplaced assembly
tig00001673	103220	53.07498547	2.18	Unplaced assembly
tig00000947	77800	52.17352185	1.59	Unplaced assembly
tig00001793	117736	51.95776993	1.36	Unplaced assembly
tig00001412	499171	50.73211384	0.28	Unplaced assembly

tig00001721	185949	49.49475394	1.29	Unplaced assembly
tig00001295	165197	49.00270586	0.65	Unplaced assembly
tig00001602	85658	48.50335053	2.84	Unplaced assembly
tig00000901	165256	47.96558067	1.27	Unplaced assembly
tig00000783	454939	47.55274883	0.73	Unplaced assembly
tig00001692	121422	45.52305184	1.73	Unplaced assembly
tig00000887	103213	37.64254503	2.35	Unplaced assembly
tig00001144	277781	36.7156141	0.53	Unplaced assembly
tig00000800	30779	35.61844114	7.89	Unplaced assembly
tig00001601	50791	34.74040676	4.75	Unplaced assembly
tig00000027	88117	25.81000261	6.04	Unplaced assembly
tig00000434	102941	18.63591766	7.44	Unplaced assembly
tig00000148	103051	17.44864193	22.02	Unplaced assembly
tig00000425	147682	16.25587411	41.34	Unplaced assembly
tig00000250	73822	16.10360055	38.75	Unplaced assembly
tig00001641	90051	13.69446203	3.23	Unplaced assembly
tig00000338	105940	8.27921465	19.62	Unplaced assembly
tig00000245	104107	6.850644049	12.07	Unplaced assembly
tig00001608	330533	5.676286483	2.17	Unplaced assembly
tig00000110	69827	5.347501683	28.60	Unplaced assembly
tig00000214	93051	5.208971424	20.35	Unplaced assembly
tig00000254	75369	4.943677109	13.07	Unplaced assembly
tig00000243	108049	4.600690427	21.13	Unplaced assembly
tig00000421	81067	3.561251804	54.91	Unplaced assembly
tig00001493	708200	1.692742163	28.60	Unplaced assembly
tig00000414	87668	1.259296437	29.86	Unplaced assembly
tig00000098	78941	0.936142182	51.45	Unplaced assembly
tig00001252	68013	0.473438901	12.30	Unplaced assembly
tig00000099	94483	0.381020924	13.94	Unplaced assembly
tig00000162	85304	0.376301229	113.15	Unplaced assembly
tig00000005	106363	0	3.82	Unplaced assembly
tig00000010	70878	0	9.06	Unplaced assembly
tig00000015	108233	0	1.46	Unplaced assembly
tig00000017	74873	0	9.58	Unplaced assembly
tig00000020	91945	0	67.28	Unplaced assembly
tig00000021	112451	0	1.02	Unplaced assembly
tig00000025	85816	0	26.78	Unplaced assembly
tig00000030	83620	0	6.75	Unplaced assembly
tig00000034	62140	0	22.84	Unplaced assembly
tig00000039	247330	0	31.63	Unplaced assembly
tig00000044	122344	0	0.95	Unplaced assembly
tig00000045	97006	0	2.59	Unplaced assembly
tig00000046	234079	0	0.62	Unplaced assembly
tig00000083	63240	0	12.58	Unplaced assembly
tig00000084	57236	0	31.02	Unplaced assembly

tig00000085	72879	0	6.45	Unplaced assembly
tig00000092	152602	0	23.84	Unplaced assembly
tig00000093	87327	0	39.91	Unplaced assembly
tig00000095	67560	0	12.28	Unplaced assembly
tig00000127	64107	0	44.91	Unplaced assembly
tig00000139	104087	0	28.39	Unplaced assembly
tig00000145	144136	0	5.94	Unplaced assembly
tig00000147	87190	0	17.74	Unplaced assembly
tig00000176	89550	0	26.50	Unplaced assembly
tig00000180	84392	0	36.42	Unplaced assembly
tig00000187	84642	0	31.56	Unplaced assembly
tig00000201	76926	0	15.33	Unplaced assembly
tig00000213	77758	0	3.35	Unplaced assembly
tig00000217	345276	0	67.03	Unplaced assembly
tig00000218	77574	0	30.77	Unplaced assembly
tig00000219	88124	0	7.90	Unplaced assembly
tig00000226	86344	0	8.63	Unplaced assembly
tig00000234	88869	0	4.80	Unplaced assembly
tig00000239	89660	0	2.42	Unplaced assembly
tig00000244	92526	0	4.88	Unplaced assembly
tig00000248	96904	0	11.37	Unplaced assembly
tig00000249	118321	0	7.33	Unplaced assembly
tig00000252	97502	0	78.66	Unplaced assembly
tig00000259	108031	0	15.80	Unplaced assembly
tig00000265	94516	0	62.51	Unplaced assembly
tig00000267	88774	0	87.62	Unplaced assembly
tig00000271	168029	0	5.54	Unplaced assembly
tig00000283	109338	0	29.52	Unplaced assembly
tig00000288	80612	0	48.72	Unplaced assembly
tig00000289	58975	0	35.11	Unplaced assembly
tig00000295	144071	0	68.18	Unplaced assembly
tig00000304	102115	0	43.99	Unplaced assembly
tig00000308	88424	0	1.71	Unplaced assembly
tig00000322	147283	0	36.63	Unplaced assembly
tig00000325	116164	0	44.91	Unplaced assembly
tig00000334	102418	0	47.59	Unplaced assembly
tig00000335	90023	0	2.21	Unplaced assembly
tig00000358	78569	0	36.83	Unplaced assembly
tig00000362	87218	0	9.65	Unplaced assembly
tig00000363	121205	0	5.90	Unplaced assembly
tig00000367	78672	0	9.92	Unplaced assembly
tig00000381	75061	0	5.71	Unplaced assembly
tig00000382	105076	0	17.67	Unplaced assembly
tig00000384	121125	0	80.02	Unplaced assembly
tig00000415	115676	0	31.67	Unplaced assembly

tig00000419	131252	0	81.19	Unplaced assembly
tig00000424	84265	0	1.34	Unplaced assembly
tig00000426	86369	0	17.75	Unplaced assembly
tig00000432	100636	0	3.03	Unplaced assembly
tig00000433	141185	0	10.38	Unplaced assembly
tig00000436	85282	0	13.62	Unplaced assembly
tig00000441	125131	0	79.31	Unplaced assembly
tig00000451	125740	0	1.88	Unplaced assembly
tig00000466	167315	0	13.17	Unplaced assembly
tig00000476	82359	0	7.58	Unplaced assembly
tig00000493	119079	0	12.75	Unplaced assembly
tig00000505	131714	0	68.25	Unplaced assembly
tig00000506	117803	0	37.46	Unplaced assembly
tig00000507	109720	0	30.49	Unplaced assembly
tig00000552	55776	0	5.76	Unplaced assembly
tig00000609	103406	0	6.83	Unplaced assembly
tig00000625	84229	0	16.95	Unplaced assembly
tig00000664	82292	0	19.15	Unplaced assembly
tig00000669	78435	0	18.94	Unplaced assembly
tig00000674	230674	0	1.66	Unplaced assembly
tig00000694	151995	0	0.89	Unplaced assembly
tig00000697	51810	0	2.52	Unplaced assembly
tig00000712	205885	0	5.74	Unplaced assembly
tig00000867	243329	0	18.21	Unplaced assembly
tig00000983	82849	0	4.06	Unplaced assembly
tig00000992	106212	0	6.94	Unplaced assembly
tig00000996	62007	0	5.08	Unplaced assembly
tig00001010	70174	0	6.61	Unplaced assembly
tig00001025	73711	0	1.42	Unplaced assembly
tig00001029	98258	0	10.29	Unplaced assembly
tig00001032	80128	0	3.94	Unplaced assembly
tig00001034	79315	0	3.98	Unplaced assembly
tig00001041	67046	0	3.85	Unplaced assembly
tig00001047	314847	0	3.89	Unplaced assembly
tig00001056	106046	0	2.71	Unplaced assembly
tig00001062	60376	0	3.64	Unplaced assembly
tig00001063	60215	0	3.95	Unplaced assembly
tig00001069	65261	0	4.13	Unplaced assembly
tig00001072	81797	0	12.28	Unplaced assembly
tig00001090	76556	0	1.34	Unplaced assembly
tig00001120	108044	0	6.27	Unplaced assembly
tig00001156	82809	0	2.54	Unplaced assembly
tig00001161	72472	0	19.92	Unplaced assembly
tig00001173	44331	0	35.77	Unplaced assembly
tig00001174	61622	0	72.30	Unplaced assembly

tig00001202	146021	0	0.75	Unplaced assembly
tig00001205	98311	0	7.80	Unplaced assembly
tig00001226	99338	0	3.75	Unplaced assembly
tig00001227	51116	0	5.10	Unplaced assembly
tig00001278	68816	0	53.39	Unplaced assembly
tig00001302	62497	0	13.27	Unplaced assembly
tig00001307	137048	0	28.74	Unplaced assembly
tig00001351	376256	0	0.62	Unplaced assembly
tig00001460	183334	0	3.94	Unplaced assembly
tig00001461	433434	0	8.82	Unplaced assembly
tig00001465	134406	0	1.42	Unplaced assembly
tig00001475	102695	0	2.03	Unplaced assembly
tig00001505	70042	0	3.99	Unplaced assembly
tig00001541	174222	0	6.72	Unplaced assembly
tig00001542	80002	0	4.87	Unplaced assembly
tig00001543	172982	0	2.15	Unplaced assembly
tig00001544	133266	0	3.72	Unplaced assembly
tig00001549	238935	0	6.17	Unplaced assembly
tig00001550	59205	0	9.70	Unplaced assembly
tig00001609	113820	0	5.54	Unplaced assembly
tig00001642	44079	0	2.32	Unplaced assembly
tig00001643	83791	0	4.54	Unplaced assembly
tig00001644	94286	0	1.70	Unplaced assembly
tig00001645	55446	0	3.93	Unplaced assembly
tig00001646	66920	0	6.46	Unplaced assembly
tig00001647	73428	0	2.97	Unplaced assembly
tig00001648	55190	0	4.12	Unplaced assembly
tig00001649	116706	0	5.20	Unplaced assembly
tig00001781	72807	0	2.82	Unplaced assembly
tig00001782	82912	0	1.26	Unplaced assembly
tig00001783	136303	0	5.32	Unplaced assembly
tig00001807	63592	0	2.06	Unplaced assembly
tig00001808	87237	0	2.50	Unplaced assembly
tig00001813	119094	0	3.39	Unplaced assembly
tig00001814	55195	0	7.80	Unplaced assembly
tig00001821	71757	0	5.02	Unplaced assembly
tig00001822	80122	0	4.22	Unplaced assembly
tig00001823	46940	0	4.64	Unplaced assembly
tig00001831	56060	0	2.57	Unplaced assembly
tig00001833	79687	0	4.79	Unplaced assembly
tig00001835	76429	0	5.63	Unplaced assembly
tig00001837	52649	0	2.03	Unplaced assembly
tig00001838	57977	0	7.68	Unplaced assembly

Supplementary Table 3. Comparison of assembly gaps in the reference Tcas5.2 assembly and TcasONT

	Original Tcas5.2 assembly	*Gap_filled Tcas5.2 assembly*	*Gap difference*	*Gap difference (%)*
Gap number (N>10)	3669	62	**3607**	**98.31016626**
Total N size	11,495,702	991,852	**10,503,850**	
1st quantile	81	41		
Median gap size	381	91		
3rd quantile	1081	98		
Maximal gap size	1,200,301	248,621		
Mean gap size	3125	15997		

Supplementary Table 4. Gene retention from official Tcas5.2 gene set to TcasONT assembly

Feature	TcasONT	Tcas5.2	Retained
GENE	14337	14467	99.10140319
CDS	150962	153698	98.21988575
EXON	167786	171320	97.93719356
MRNA	22267	22598	98.53526861
LNC_RNA	1406	1364	103.0791789*
TRANSCRIPT	308	317	97.16088328
PRIMARY_TRANSCRIPT	226	220	102.7272727*
TRNA	236	247	95.5465587

Supplementary Table 5. Comparision of number and cumulative length of repetitive elements in T. castaneum TcastONT and Tcas5.2 assembly. The blue shade indicates transposable elements.

Number of elements

	TcasONT	Tcas5.2	difference	difference (%)
DNA	45267	33970	11297	24.96
LINE	32237	4684	27553	85.47
LTR	14861	2593	12268	82.55
RC	2746	1997	749	27.28
RRNA	998	346	652	65.33
SINE	250	190	60	24.00

TRNA	29	26	3	10.34
SIMPLE_REPEAT	73293	36673	36620	49.96
LOW_COMPLEXITY	16286	10168	6118	37.57
UNKNOWN	789	597	192	24.33
TOTAL	186756	91244	95512	
TOTAL TES	92615	41437	51178	

Length of elements (bp)

	TcastONT	Tcas5.2	difference	difference (%)
DNA	13499636	8279569	5220067	38.67
LINE	16084939	1572720	14512219	90.22
LTR	2542411	766028	1776383	69.87
RC	353694	258101	95593	27.03
RRNA	354597	50383	304214	85.79
SINE	31176	24455	6721	21.56
TRNA	2202	1978	224	10.17
SIMPLE_REPEAT	4030246	1662985	2367261	58.74
LOW_COMPLEXITY	752292	487704	264588	35.17
UNKNOWN	117266	59604	57662	49.17
TOTAL	37768459	13163527	24604932	
TOTAL TES	32158162	10642772	21515390	

Supplementary Table 6. Comparision of tandem repeat cumulative length in *T. castaneum* TcastONT and Tcas5.2 assembly.

Total sum of tandem repeats (TR) length (bp)

PERIOD SIZE	Tcas5.2	TcasONT	Difference
<50	1825166	3639659	1814493
50-500	4704595	16769526	12064931
>500	2618407	14895753	12277346
TOTAL	9148168	35304938	26156770
TOTAL LARGE (>50)	7323002	31665279	24342277
TOTAL ENRICHMENT OF REPETITIVE	50761702		

Supplementary Table 7. Statistical analysis of Cast arrays flanking regions and gene presence distribution . The results of the one-sided Kolmogorov-Smirnov test for Cast satDNAs flanking regions having significantly (p<0.01) fewer (red) or more (green) genes than the average same-sized *T. castaneum* genome sequence.

satDNA family	Significantly less genes	Significantly more genes
Cast1	0.001494	0.482069
Cast2'	0.549259	0
Cast2	0.678151	0.085932
Cast3	0.863316	0.000014
Cast4	0.812581	0.000307
Cast5	0.986053	0
Cast6	0.043741	0.682585
Cast7	0	0.855513
Cast8	0.9896	0.0022
Cast9	0.290457	0.014985

Supplementary Table 8. Number of Cast1-Cast9 arrays per MB of chromosome length

Chromosome	Length (in mb)	Number of Cast1-Cast9 arrays	N/mb
LG10	16.52	419	25.36
LG2	18.60	164	8.81
LG3	40.53	444	10.95
LG4	13.99	131	9.36
LG5	17.65	209	11.84
LG6	12.97	274	21.12
LG7	21.23	299	14.09
LG8	16.31	358	21.95
LG9	23.52	269	11.44
LGX	10.26	98	9.55

Supplementary Code

Supplementary Code 1. BLAST functions used in defining the satDNA monomers in both TcasONT and Tcas5.2 assemblies.

```
blast_to_gff <- function(s_name,q_name,name,work_dir)
{

setwd(work_dir)
blasts <- readDNAStringSet(s_name)
#try(blasts <- blasts[1:10000])
blastq <- readDNAStringSet(q_name)

writeXStringSet(blasts,"C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blasts.fa",format="fasta")
writeXStringSet(blastq,"C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blastq.fa",format="fasta")

blast_dt <- blast_nucleotide_to_nucleotide(
            query   = 'C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blastq.fa',
            subject = 'C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blasts.fa',
            output.path = tempdir(),
            db.import  = FALSE,
            evalue = 0.001,
            cores= 16) %>% as.data.table(.)

q_tmp_dt <- data.table(query_id=names(blastq),widt=width(blastq))
casts_in_un <- blast_dt
gff_temp <- casts_in_un[qcovhsp>70 & perc_identity>70,c("subject_id","query_id","s_start","s_end","bit_score")]

setnames(gff_temp,c("subject_id","query_id","s_start","s_end","bit_score"),c("seqnames","feature","start","end","scor
e"))
gff_temp[,source:="Rblast"]
gff_temp[,strand:="+"]
gff_temp[,frame:="."]
gff_temp[,group:=name]
gff_temp[start>end, c("end", "start") := .(start, end)]
setcolorder(gff_temp,c("seqnames","source","feature","start","end","score","strand","frame","group"))
file = paste0(getwd(),"/",name,".gff")
fwrite(gff_temp, file = file, row.names=FALSE, sep="\t",quote=FALSE,col.names = FALSE)
return(gff_temp)
}

blast_to_raw<- function(s_name,q_name,name,work_dir)
{
setwd(work_dir)
blasts <- readDNAStringSet(s_name)
```

```
blastq <- readDNAStringSet(q_name)
writeXStringSet(blasts,"C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blasts.fa",format="fasta")
writeXStringSet(blastq,"C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blastq.fa",format="fasta")
blast_dt <- blast_nucleotide_to_nucleotide(
            query   = 'C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blastq.fa',
            subject = 'C:/Users/User/Documents/R/win-library/4.0/metablastr/seqs/blasts.fa',
            output.path = tempdir(),
            db.import  = FALSE,
            evalue = 0.001,
            cores= 16) %>% as.data.table(.)
q_tmp_dt <- data.table(query_id=names(blastq),widt=width(blastq))
casts_in_un <- blast_dt
blast_dt <- casts_in_un#[qcovhsp>70 &
perc_identity>70]#,c("subject_id","query_id","s_start","s_end","bit_score","strand")]
#blast_dt[start>end, c("end", "start") := .(start, end)]
return(blast_dt)
}
rpt_fix <- function(dt,katalog)
{
dt <- copy(dt)
dt[,V10:=str_remove(V10,"Motif:")]
dt[,V3:=V10]
#imena kroz katalog repeatova da bi se dobile klase repeatova
crossref <- copy(katalog)
 setnames(crossref,c("pos in repeat: begin","repeat","class/family"),c("status","type","class"))
   crossref[grep("^\\D+",status),type:=class]
    crossref[grep("^\\D+",status),class:=status]
crossref <- crossref[,.(class,type)] %>% unique(.)
setnames(dt,"V10","type")


setkey(crossref,type)
setkey(dt,type)
print(dt)
dt <- dt
crossref<-crossref
dt2 <-  merge(crossref,dt,by="type",allow.cartesian=TRUE)
dt2[type==V3]
dt2[,V3:=class]
dt2[,class:=NULL]
dt2[,V9:=type]
dt2[,type:=NULL]
print(dt2)
return(dt2)
}
```

Supplementary Code 2. Filtering contigs. This scripts defines the steps for contig filtering in the assembly.

```{r}
library(data.table)
library(Biostrings)
library(stringr)
source("blast_functions.R")

```
#filtering contigs, based on 1000bp gene content in them, everything else gets discarded
```{r}

genes_on_tigs <- fread("./scaffolding_results/genes_on_contigs.gff3",fill=TRUE,header=FALSE,skip=3,sep="\t")

genes_on_tigs[,.N,by=V1][N>10]

genes_on_tigs <- genes_on_tigs[V3=="gene",sum(V5-V4),by=V1]

colnames(genes_on_tigs) <- c("names","length")


tigs <- readDNAStringSet("scaffolding_results/ragtag/t_cast_contigs.fasta")

tigs[str_remove(names(tigs)," .*")%in%genes_on_tigs[length>1000,names]] %>%
writeXStringSet("./scaffolding_results/filtered_contigs.fasta")
```
analyis of mapped contigs

```{r}
dt <- fread("./scaffolding_results/scafolding_output_1/ragtag.scaffold.confidence.txt")
names <- dt[,query]
contigs <- readDNAStringSet("./scaffolding_results/filtered_contigs.fasta")
contigs[str_remove(names(contigs)," .+")%in%dt[,query]] %>%
writeXStringSet("./scaffolding_results/ragtag/included_contigs.fasta")
all_contigs <- names(contigs)

```
#blasting sattelites on contigs finding sat content on contigs

```{r}
 dt <- blast_to_raw(q_name = "casts_19.fasta",s_name = "TcasONT.fasta",work_dir =
"E:/t_cast_assembly/assembly_analysis/",name="blast_sats_assembly")
dt <- dtl
# dt <- blast_to_gff(q_name = "main_sat.fasta",s_name = "t_cast_contigs.fasta",work_dir =
"E:/t_cast_assembly/assembly_analysis/",name="blast_main_sat_assembly")
```

```
dt <- fread("E:/t_cast_assembly/assembly_analysis/blast_main_sat_assembly.gff")
dt[V4>V5, c("V5", "V4") := .(V4, V5)]
tig_lengths <- readDNAStringSet("E:/t_cast_assembly/contig_analysis/t_cast_contigs.fasta")

tig_lengths <- data.table(seqnames=str_remove(names(tig_lengths)," .+"),seq_width=width(tig_lengths))
dt <- merge(dt,tig_lengths,by.x="V1",by.y="seqnames")
dt <- dt[V6>400] %>%makeGRangesFromDataFrame(seqnames.field = "V1",start.field = "V4",end.field =
"V5") %>%  reduce() %>% as.data.table()
dt <- merge(dt,tig_lengths,all=TRUE)
dt <- dt[,sum(width),by=c("seqnames","seq_width")]
dt[,sat_percentage:=round(V1*100/seq_width,5)]
gene_perc <- fread("./scaffolding_results/ragtag/genes_on_contigs.gff3",skip=3) %>% .[V3=="gene"]
gene_perc[,width:=abs(V5-V4)]
gene_perc <- gene_perc %>% .[,sum(width),by=V1]
colnames(gene_perc) <- c("seqnames","gene_length")
dt <- merge(dt,gene_perc,all=TRUE)
dt[,gene_perc:=gene_length*100/seq_width]
dt_2 <- dt
dt_2[,in_assembly:="Not in assembly"]
dt_2[seqnames%in%str_remove(all_contigs," .*"),in_assembly:="Unplaced assembly"]
dt_2[seqnames%in%names,in_assembly:="Final assembly"]
dt_2[is.na(sat_percentage),sat_percentage:=0]
dt_2[is.na(gene_perc),gene_perc:=0]
dt_2[in_assembly !="Not in assembly" ,c(1,2,4,6,7)] %>%
fwrite("./scaffolding_results/ragtag/contig_gene_sat_content_za_evelin.csv")
sv_1 <- alignments[,unique(cum_ref),by=refID][order(V1)][,unique(V1)]
namess <- alignments[,unique(cum_ref),by=refID][,unique(refID)]
sv_2 <- alignments[,unique(cum_query),by=queryID][,V1]
namesq <- alignments[,unique(cum_query),by=queryID][,queryID]
alignments[,unique(cum_ref),by=refID]
dt[order(-sat_percentage)]
```

Supplementary Code 3. Finding and creating arrays in TcasONT and Tcas5.2 assemblies

```{r}
source("blast_functions.R")
library(data.table)
library(Biostrings)
library(stringr)
library(tidyverse)
library(ggplot2)
```

satelite content, 2d density plot analyis for filtering parameters

```{r}

sat_cont_dt <- blast_to_raw(q_name = "casts_19.fasta",s_name = "TcasONT.fasta",work_dir =
"./data/",name="blast_sat_assembly")

tmp <-
sat_cont_dt[grep("Cast",query_id),.N,by=c("query_id","qcovhsp","perc_identity")] %>% .[,perc_identity:=round(perc_i
dentity)]

p <- ggplot(tmp[-grep("Cast2-prime",query_id)], aes(qcovhsp, perc_identity)) +
 geom_density_2d_filled(contour_var = "ndensity",bins=50) +
 facet_wrap(vars(query_id))+
 theme_bw() +
 scale_fill_discrete_divergingx()+
 ylab("Percentage identity (%)") +
 xlab("Query coverage (%)") +
 theme(legend.position = "none")

```

#finding the cast2 array size

```{r}
ext_table <- data.table(name=unique(sat_cont_dt[grep("Cast",feature),feature])) %>% dcast(...~name)
vec <- seq(from=100,to=2000,by=10)
ext_table <- rbind(ext_table,vec,fill=T) %>% .[-1]
ext_table[,.:=NULL]

ext_table <- ext_table %>% melt(id.vars="x")


fun <- function(ext_factor,array="Cast1")
{
 sat_copy <- copy(sat_cont_dt[feature==array])
 sat_copy[,enE:=end+ext_factor]
```

```
 result <- sat_copy %>% makeGRangesFromDataFrame() %>% reduce() %>% as.data.table() %>% .[,mean(width-
ext_factor)]
 return(result)
}
for(i in unique(ext_table[,variable]))
{
ext_table[variable==i,value:=sapply(ext_table[variable==i,x],FUN = fun,array=i)]
}

ext_table[,value:=as.double(value)]
ext_table[,value_scaleE:=value/max(value),by=variable]
ext_table[,cumsum_value:=cumsum(value_scaled),by=variable]
sat_cont_dt

p <- ext_table %>% ggplot() +
 geom_line(aes(x=x,y=value_scaled,color=variable)) +
 scale_color_npg() + facet_wrap(~variable) +
 theme_bw()

```
#creating all arrays file
```{r}
#setnames(sat_cont_dt,"query_id","feature")
names <- unique(ext_table[,variable])
ext_fact <- c(250,250,250,250,250,1000,250,500,250,250)
ext_fac_dt <- data.table(names,ext_fact)


for (i in ext_fac_dt[,names])
{
 print(i)
 ext_factor <- ext_fac_dt[names==i,ext_fact]
 sat_copy <- copy(sat_cont_dt[feature==i])
 sat_copy[,enE:=end+ext_factor]
 result <- sat_copy %>% makeGRangesFromDataFrame() %>% reduce() %>% as.data.table()
 result[,array:=i]
 #turn off for stats
 result[,width:=width-ext_factor]
 result[,enE:=end-ext_factor]
 if (i =="Cast1")
 {
  array_dt <- result
 }

 if (i !="Cast1")
```

```
 {
  array_dt <- rbind(array_dt,result)
 }

}

#if cast2 array has a cast2-prime within 170 bp beofre or after it is a Cast2-prime array
cast2_array_dt <- array_dt[array=="Cast2"]
cast2_array_dt[,ar_id:=paste(seqnames,start,sep="_")]
cast2_array_dt[,c("start", "end") := .(start-170, end+170)]
cast_prime_array_dt <- array_dt[array=="Cast2-prime"]
try(cast_prime_array_dt[,ar_id:=NULL])
setkey(cast2_array_dt,seqnames,start,end)
setkey(cast_prime_array_dt,seqnames,start,end)
ar_ids_with_cast_prime <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit() %>% .[,ar_id]
new_arrays <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit()

final_array_number <- new_arrays[,c("seqnames","start","end","i.end","ar_id","i.width","i.start")] %>%
 .[i.start<start,start:=start-i.width] %>%
 .[end>i.end,i.enE:=end] %>%
 .[end<i.end,i.enE:=i.end-170] %>%
 makeGRangesFromDataFrame(start.field = "start",
                          end.field = "i.end",
                          seqnames.field = "seqnames") %>%
 reduce() %>%
 as.data.table()

#if cast2-prime

array_dt[array=="Cast2",ar_id:=paste(seqnames,start,sep="_")]
array_dt[array=="Cast2" & ar_id%notin%ar_ids_with_cast_prime,array:="Cast2_pure"]
final_array_number[,array:="Cast2-mix"]
array_dt <- rbind(array_dt[array!="Cast2" & array!="Cast2-prime"],final_array_number,fill=TRUE)

#gff save for arrays
{
gff_temp <- copy(array_dt_cast2_fixed)

gff_temp[,ar_id:=NULL]


gff_temp[,width:=NULL]
gff_temp[,stranE:=NULL]
gff_temp[,score:=1000]
colnames(gff_temp) <- c("seqnames","start","end","feature","score")
```

```
gff_temp[,source:="Rblast"]
gff_temp[,stranE:="+"]
gff_temp[,frame:="."]
gff_temp[,group:=feature]
setcolorder(gff_temp,c("seqnames","source","feature","start","end","score","strand","frame","group"))
fwrite(gff_temp, file = "./data/full_array_annot.gff3", row.names=FALSE, sep="\t",quote=FALSE,col.names = FALSE)
}
mean <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),mean(width),by=array][,V1]
median <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),median(width),by=array][,V1]
number <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),.N,by=array][,N]
total_len <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),sum(width),by=array][,V1]
names <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),.N,by=array][,array]
max <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),max(width),by=array][,V1]

data.table(names,number,mean,median,max,total_len) %>% .[order(names)]
array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000)]
casts <- readDNAStringSet("./data/casts_19.fasta")
dt <- data.table(width(casts),names(casts)) %>% .[order(V2)]
dt
```
#blasting for sattelite on assembly, new and old

```{r}
sat_cont_dt <- blast_to_gff(q_name = "casts_19.fasta",s_name = "TcasONT.fasta",work_dir =
"./data/",name="blast_sat_assembly")
sat_cont_trim_reads <- blast_to_gff(q_name = "casts_19.fasta",s_name = "15x_corrected_coverage.fasta",work_dir =
"./data/",name="blast_sat_trim_15x_assembly")


sat_cont_15x_reads <- blast_to_gff(q_name = "casts_19.fasta",s_name = "15x_corrected_coverage.fasta",work_dir =
"E:/t_cast_assembly/assembly_analysis/",name="blast_sat_trim_15x_assembly")

sat_cont_dt[grep("NC",seqnames),.N,by=feature][order(feature)]
sat_cont_52_dt[grep("NC",seqnames),.N,by=feature][order(feature)]
sat_cont_reads_cor[,.N,by=feature][order(feature)]
sat_cont_trim_reads[,.N,by=feature][order(feature)]

sat_cont_trim_reads[,.N,by=feature][order(feature)]

sat_cont_15x_reads[,sum(end-start)*100/y,by=feature][order(feature)]


y <- sum(width(reads_used))
```

```
#comparison with tcas5.2
```{r}
sat_cont_52_dt <- blast_to_gff(q_name = "casts_19.fasta",s_name = "tcast_full_assembly.fasta",work_dir =
"./data/",name="blast_sat_assembly_52")
names <- unique(ext_table[,variable])
ext_fact <- c(250,250,250,250,250,1000,250,500,250,250)
ext_fac_dt <- data.table(names,ext_fact)


for (i in ext_fac_dt[,names])
{
 print(i)
 ext_factor <- ext_fac_dt[names==i,ext_fact]
 sat_copy <- copy(sat_cont_52_dt[feature==i])
 sat_copy[,enE:=end+ext_factor]
 result <- sat_copy %>% makeGRangesFromDataFrame() %>% reduce() %>% as.data.table()
 result[,array:=i]
 if (i =="Cast1")
 {
  array_dt_52 <- result
 }

 if (i !="Cast1")
 {
  array_dt_52 <- rbind(array_dt_52,result)
 }
}
array_dt_52 <- array_dt_52[grep("NC",seqnames)]
#if cast2 array has a cast2-prime within 170 bp beofre or after it is a Cast2-prime array
cast2_array_dt <- array_dt_52[array=="Cast2"]
cast2_array_dt[,ar_id:=paste(seqnames,start,sep="_")]
cast2_array_dt[,c("start", "end") := .(start-170, end+170)]


cast_prime_array_dt <- array_dt_52[array=="Cast2-prime"]
cast_prime_array_dt[,ar_id:=NULL]


setkey(cast2_array_dt,seqnames,start,end)
setkey(cast_prime_array_dt,seqnames,start,end)

ar_ids_with_cast_prime <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit() %>% .[,ar_id]
```
```

```
new_arrays <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit()

final_array_number <- new_arrays[,c("seqnames","start","end","i.end","ar_id","i.width","i.start")] %>%
 .[i.start<start,start:=start-i.width] %>%
 .[end>i.end,i.enE:=end] %>%
 .[end<i.end,i.enE:=i.end-170] %>%
 makeGRangesFromDataFrame(start.field = "start",
                          end.field = "i.end",
                          seqnames.field = "seqnames") %>%
 reduce() %>%
 as.data.table()

#if cast2-prime

array_dt_52[array=="Cast2",ar_id:=paste(seqnames,start,sep="_")]


array_dt_52[array=="Cast2" & ar_id%notin%ar_ids_with_cast_prime,array:="Cast2_pure"]

final_array_number[,array:="Cast2-mix"]

array_dt_52 <- rbind(array_dt_52[array!="Cast2" & array!="Cast2-prime"],final_array_number,fill=TRUE)




mean <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),mean(width),by=array][,V1]
median <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),median(width),by=array][,V1]
number <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),.N,by=array][,N]
total_len <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),sum(width),by=array][,V1]
names <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000),.N,by=array][,array]
max <- array_dt_52[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),max(width),by=array][,V1]

data.table(names,number,mean,median,max,total_len) %>% .[order(names)]
sat_cont_52_dt
sat_cont_dt
array_dt
```

Supplementary Code 4. General assembly statistics calculated on theTcasONT assembly and gene completeness analysis

```{r}
library(data.table)
library(Biostrings)
library(stringr)
library(tidyverse)
```
#BUSCO analyis 1,2,3
```{r}

file_names <- list.files("./scaffolding_results/busco_analysis/") %>% grep("busco_Tca",.,value=T)
busco_table <- data.table()
for (i in file_names)
{
  dt <- fread(paste0("./scaffolding_results/busco_analysis/",i),fill=TRUE,skip = 3)
  name_id=str_remove(i,"busco_")
  name_id=str_remove(name_id,"\\..+")
  print(name_id)
  dt[,name:=name_id]
  busco_table=rbind(busco_table,dt)
}

busco_table[,.N,by=c("name","V2")] %>% dcast(...~name) %>%
fwrite("./scaffolding_results/busco_analysis/busco_output_table.csv")

```
#repeat masker analysis
```{r}
rpts_polished_ONT <- fread("./data/TcasONT_repeats.gff")
rpts_t_cast <- fread("./data/Tcas52_repeats.gff")


rpts_polished_ONT[,source:="Tcast_ONT"]
rpts_t_cast[,source:="Tcas5.2"]
rpt_tot <- rbind(rpts_polished_ONT,rpts_t_cast)

rpt_tot[,.N,by=c("V2","source")]

rpt_tot[,group:=V3]
rpt_tot[grep("DNA",V3),group:="DNA"]
rpt_tot[grep("LINE",V3),group:="LINE"]
```

```
rpt_tot[grep("RC",V3),group:="RC"]
rpt_tot[grep("SINE",V3),group:="SINE"]
rpt_tot[grep("LTR",V3),group:="LTR"]

rpt_tot[,.N,by="group"]

rpt_tot[,source:=factor(source,levels = c("polished","tcast5_2"))]
lvl <- rpt_tot[,.N,by="group"][order(-N)][,group]
rpt_tot[,group:=factor(group,levels = lvl)]

rpt_tot[group!="Unkown" & group!="Retroposon" &group!="Retroposon?" &group!="Satellite" ] %>% ggplot() +
geom_bar(aes(x=group,fill=source),position="dodge") + scale_fill_npg()

rpt_tot[group!="Unkown" & group!="Retroposon" &group!="Retroposon?" &group!="Satellite"][,sum(V5-
V4),by=c("group","source")] %>% ggplot() + geom_col(aes(x=group,y=V1,fill=source),position="dodge") +
scale_fill_npg()

rpt_tot[group!="Unkown" & group!="Retroposon" &group!="Retroposon?"
&group!="Satellite" ][,.N,by=c("group","source")] %>% dcast(group~source) %>%
fwrite("./repeat_masker_results/number_of_repeats.csv",sep="\t")

rpt_tot[group!="Unkown" & group!="Retroposon" &group!="Retroposon?" &group!="Satellite" ][,sum(V5-
V4),by=c("group","source")] %>% dcast(group~source) %>%
fwrite("./repeat_masker_results/length_of_repeats.csv",sep="\t")
```
#gene analysis

```{r}

genes_tcast <- fread("./data/GCF_000002335.3_Tcas5.2_genomic.gff",skip=3,fill=T,sep="\t")
genes_ont <- fread("./data/TcasONT_genes.gff3",skip=9,fill=T,sep="\t")

genes_tcast <- merge(genes_tcast,name_links,by.x="V1",by.y="V7") %>% .[,V1:=V3.y]

genes_ont <- genes_ont[,V1:=str_remove(V1,"_RagTag")] %>%
merge(.,name_links,by.x="V1",by.y="V7") %>% .[,V1:=V3.y]

merge(genes_ont[V3.x=="gene",.N,by=V1],genes_tcast[V3.x=="gene",.N,by=V1],by="V1") %>%
fwrite("./data/gene_content_by_chromosome.tsv")


```

#chromosome lengths statistics
```{r}


```

```
tcast_52 <- readDNAStringSet("./data/tcast_full_assembly.fasta")[1:10]
ONT_assembly <- readDNAStringSet("./data/TcasONT.fasta")[1:10]
dt <- data.table(names=str_remove(names(tcast_52)," .+"),width=width(tcast_52),source="tcast52")
dt_2 <- data.table(names=str_remove(names(ONT_assembly)," .+"),width=width(ONT_assembly),source="ONT")
dt <- rbind(dt,dt_2)
dt %>% dcast(V3~source,value.var = "width") %>% fwrite("./data/chr_lengths.tsv",sep="\t")

```

Supplementary Code 5. Relationship between Cast1-Cast9 satDNAs and transposable elements/genes

```{r}
source("blast_functions.R")
library(data.table)
library(Biostrings)
library(stringr)
library(tidyverse)
library(ggplot2)
```
#finding genes in cast vicinity

```{r}
chroms <- readDNAStringSet("./data/TcasONT.fasta")
dt_width <- data.table(names=names(chroms),width=width(chroms))
arrays <- fread("./data/full_array_annot.gff")
arrays <- merge(arrays,dt_width,by.x="V1",by.y="names")

array_dt <- arrays[,c(1,3,4,5,10)]
try(setnames(array_dt,c("V1","V3","V4","V5"),c("seqnames","array","start","end")))
genes_ont <- fread("./data/TcasONT_genes.gff",skip=0,fill=T,sep="\t")
genes_ont <- genes_ont[V3=="exon"]
setnames(genes_ont,c("V1","V4","V5"),c("seqnames","start","end"))
genes_ont[start>end, c("end", "start") := .(start, end)]
array_dt[,ar_id:=paste(array,seqnames,as.character(start),sep="_")]
array_dt[,width:=abs(end-start)]
genedt <- genes_ont
cont_fac <- 50000
bin=50
try(setnames(genedt,c("seqnames","start","end"),c("V1","V4","V5")))
#######bef array
glob_tmp <- copy(array_dt[width>330])
glob_tmp[,c("start", "end") := .(start-cont_fac, start)]

glob_tmp[,ar_size:=as.character(cut(glob_tmp$width,
                           breaks=c(0,1000,10000,50000)#,
                         #labels=c("1Q","2Q","3Q"), include.lowest=TRUE
                         ))
        ]
setkey(glob_tmp,seqnames,start,end)
setkey(genedt,V1,V4,V5)
overlapdt <- foverlaps(genedt,glob_tmp)
overlapdt[,c("V5","start"):=.(V5-start,start-start)]
overlapdt <- na.omit(overlapdt)
```

```
overlapdt[,range:=as.integer(cut(overlapdt$V5,
                breaks=bin,
                labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,V5], bin))) ),
                ))]

# overlapdt[,ar_quant:=as.character(cut(overlapdt$width,
#                               breaks=c(quantile(overlapdt$width,
#                                                 probs = seq(0, 1, by = 0.25))
#                                        ),
#                               labels=c("1Q","2Q","3Q","4Q"), include.lowest=TRUE
#                               ))
#         ]


ol_bef <- overlapdt %>% .[,.N,by=c("range","array","ar_size","width","ar_id")]
ol_bef[,range:=-range*(cont_fac/bin)]

##FOR GEOM DENSITY
overlapdt[,range:=-range*(cont_fac/bin)]
ol_bef <- overlapdt
#######################
#####after array
glob_tmp <- copy(array_dt[width>330])
glob_tmp[,c("start", "end") := .(end, end + cont_fac)]
setkey(glob_tmp,seqnames,start,end)
setkey(genedt,V1,V4,V5)
glob_tmp[,ar_size:=as.character(cut(glob_tmp$width,
                                breaks=c(0,1000,10000,50000)#,
                               #labels=c("1Q","2Q","3Q"), include.lowest=TRUE
                               ))
        ]
overlapdt <- foverlaps(genedt,glob_tmp)
overlapdt[,c("V4","end"):=.(V4-end,end-end)]
overlapdt <- na.omit(overlapdt)
overlapdt[,range:=as.integer(cut(overlapdt$V4,
                breaks=bin,
                labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,V4], bin))) ),
                ))]
                #labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,width], 5))) ),

ol_af <- overlapdt %>% .[,.N,by=c("range","array","ar_size","width","ar_id")]
ol_af[,range:=range*(cont_fac/bin)]
#geom_density
overlapdt[,range:=range*(cont_fac/bin)]
```

```
ol_af <- overlapdt
plot_cast_dt <- rbind(ol_bef,ol_af)
limp <- array_dt[width>330,.N,by=array]
setnames(limp,"N","total_N")
plot_cast_dt <- merge(plot_cast_dt,limp,by="array")
plot_cast_dt[,array:=str_replace(array,"Cast2-mix","Cast2'")]
plot_cast_dt[,array:=str_replace(array,"Cast2_pure","Cast2")]
plot_cast_dt[,.N,by=c("array","ar_id","total_N","ar_size","width")]  %>%
  .[,N_scale:=N] %>%   .[] %>%
  ggplot() +
  geom_boxplot(aes(x=array,fill=array,y=N_scale)) +
  geom_hline(yintercept = 56,color="black",alpha=0.9,linetype = "dashed")+
  geom_hline(yintercept = 15,color="red",alpha=0.6,linetype = "dashed") +
  geom_hline(yintercept = 127,color="red",alpha=0.6,linetype = "dashed") +
  theme_bw() +
  scale_fill_npg() +
  theme(legend.position = "none") +
  ylab("N genes") +
  xlab("")
results <- aov(data=plot_cast_dt[array=="Cast5" &
ar_size=="(1e+04,5e+04]",.N,by=.(range,array)][,range:=as.factor(range)],formula = N~range)
res <- TukeyHSD(results)
grep("Cast5",res$`array:range`)
dt <- data.table(res$`array:range`,keep.rownames = TRUE)

```
```

scaling
```{r}
limp <- array_dt[width>330,.N,by=array]
setnames(limp,"N","total_N")

plot_cast_dt <- merge(plot_cast_dt,limp,by="array")


plot_cast_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),sum(N),by=c("range","ar_size","array")] %>%
  ggplot() +
  geom_line(aes(x=range,y=V1,color=ar_size)) +
  facet_wrap(~array) +
    theme_bw() +
```
```

```
  scale_color_npg() +
   geom_vline(xintercept = 0,color="red",alpha=0.3,linetype = "dashed") +
  scale_x_continuous(labels = mult_format(10000))
plot_cast_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width >
2000),sum(N),by=c("range","ar_size","array","total_N")] %>%
  .[,V1:=V1/total_N] %>%
  ggplot() +
  geom_line(aes(x=range,y=V1,color=ar_size)) +
  facet_wrap(~array) +
    theme_bw() +
  scale_color_npg() +
   geom_vline(xintercept = 0,color="red",alpha=0.3,linetype = "dashed") +
  scale_x_continuous(labels = mult_format(10000))


plot_cast_dt %>% ggplot() +
  geom_histogram(aes(x=N),bins=100) +
  facet_wrap(~array,scales="free") + xlab("")
#(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000)
plot_cast_dt[,array:=str_replace(array,"Cast2-mix","Cast2'")]

plot_cast_dt[,array:=str_replace(array,"Cast2_pure","Cast2")]

plot_cast_dt[,.N,by=c("array","ar_id","total_N","ar_size","width")]  %>%
  .[,N_scale:=N] %>%   .[] %>%
  ggplot() +
  geom_violin(aes(x=array,fill=array,y=N_scale)) +
  geom_hline(yintercept = 56,color="black",alpha=0.9,linetype = "dashed")+
  geom_hline(yintercept = 15,color="red",alpha=0.6,linetype = "dashed") +
  geom_hline(yintercept = 127,color="red",alpha=0.6,linetype = "dashed") +
  theme_bw() +
  scale_fill_npg()

plot_cast_dt %>%   .[] %>%
  ggplot() +
  geom_density2d_filled(aes(x=width,y=width)) +
  theme_bw() +
  scale_fill_npg()


```
#relationship with other repeat elements

```{r}
```

```
chroms <- readDNAStringSet("./data/TcasONT.fasta")
dt_width <- data.table(names=names(chroms),width=width(chroms))
arrays <- fread("./data/full_array_annot.gff")
arrays <- merge(arrays,dt_width,by.x="V1",by.y="names")

array_dt <- arrays[,c(1,3,4,5,10)]
try(setnames(array_dt,c("V1","V3","V4","V5"),c("seqnames","array","start","end")))
genes_ont <- fread("./data/TcasONT_repeats.gff",skip=0,fill=T,sep="\t")
genes_ont <- genes_ont[V3!="Simple_repeat" & V3!="Low_complexity"]
setnames(genes_ont,c("V1","V4","V5"),c("seqnames","start","end"))
genes_ont[start>end, c("end", "start") := .(start, end)]
array_dt[,ar_id:=paste(array,seqnames,as.character(start),sep="_")]
array_dt[,width:=abs(end-start)]
genedt <- genes_ont
cont_fac <- 50000
bin=10
try(setnames(genedt,c("seqnames","start","end"),c("V1","V4","V5")))
#######bef array
glob_tmp <- copy(array_dt[width>330])
glob_tmp[,c("start", "end") := .(start-cont_fac, start)]

glob_tmp[,ar_size:=as.character(cut(glob_tmp$width,
                           breaks=c(0,1000,10000,50000)#,
                         #labels=c("1Q","2Q","3Q"), include.lowest=TRUE
                         ))
        ]
setkey(glob_tmp,seqnames,start,end)
setkey(genedt,V1,V4,V5)
overlapdt <- foverlaps(genedt,glob_tmp)
overlapdt[,c("V5","start"):=.(V5-start,start-start)]
overlapdt <- na.omit(overlapdt)

overlapdt[,range:=as.integer(cut(overlapdt$V5,
                 breaks=bin,
                 labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,V5], bin))) ),
                 ))]

# overlapdt[,ar_quant:=as.character(cut(overlapdt$width,
#                              breaks=c(quantile(overlapdt$width,
#                                               probs = seq(0, 1, by = 0.25))
#                                      ),
#                            labels=c("1Q","2Q","3Q","4Q"), include.lowest=TRUE
#                            ))
#         ]
```

```
ol_bef <- overlapdt %>% .[,.N,by=c("range","array","ar_size","width","ar_id")]
ol_bef[,range:=-range*(cont_fac/bin)]

##FOR GEOM DENSITY
overlapdt[,range:=-range*(cont_fac/bin)]
ol_bef <- overlapdt
######################
#####after array
glob_tmp <- copy(array_dt[width>330])
glob_tmp[,c("start", "end") := .(end, end + cont_fac)]
setkey(glob_tmp,seqnames,start,end)
setkey(genedt,V1,V4,V5)
glob_tmp[,ar_size:=as.character(cut(glob_tmp$width,
                          breaks=c(0,1000,10000,50000)#,
                        #labels=c("1Q","2Q","3Q"), include.lowest=TRUE
                        ))
        ]
overlapdt <- foverlaps(genedt,glob_tmp)
overlapdt[,c("V4","end"):=.(V4-end,end-end)]
overlapdt <- na.omit(overlapdt)
overlapdt[,range:=as.integer(cut(overlapdt$V4,
                 breaks=bin,
                 labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,V4], bin))) ),
                 ))]

                 #labels=as.numeric( sub("\\((.+),.*", "\\1", levels(cut(overlapdt[,width], 5))) ),

ol_af <- overlapdt %>% .[,.N,by=c("range","array","ar_size","width","ar_id")]
ol_af[,range:=range*(cont_fac/bin)]
#geom_density
overlapdt[,range:=range*(cont_fac/bin)]
ol_af <- overlapdt

plot_cast_dt <- rbind(ol_bef,ol_af)

limp <- array_dt[width>330,.N,by=array]
setnames(limp,"N","total_N")

plot_cast_dt <- merge(plot_cast_dt,limp,by="array")
plot_cast_dt[,array:=str_replace(array,"Cast2-mix","Cast2'")]

plot_cast_dt[,array:=str_replace(array,"Cast2_pure","Cast2")]
genes_ont[,cutw:=cut_width(V4,width=100000,labels=F)]

genes_ont[,.N,by=cutw][,summary(N)]
```

```
genes_ont[,.N,by=cutw] %>% ggplot() + geom_histogram(aes(x=N))
plot_cast_dt[ V9!="CR1-3_TCa" & V9!="Gypsy-18_PBa-I"][,.N,by=c("array","ar_id","total_N","ar_size","width")]   %>%
  ggplot() +
  geom_boxplot(aes(x=array,fill=array,y=N)) +
  theme_bw() +
  scale_fill_npg() +
  geom_hline(yintercept = 101,color="black",alpha=0.9,linetype = "dashed")+
  geom_hline(yintercept = 35,color="red",alpha=0.6,linetype = "dashed") +
  geom_hline(yintercept = 233,color="red",alpha=0.6,linetype = "dashed") +
  theme(legend.position = "none") +
  xlab("") +
  ylab("N TE")


tmp <- genes_ont[,.N,by=cutw]
tmp[,array:="Genome"]

aov_dt <- rbind(plot_cast_dt[,.N,by=c("array","ar_id","total_N","ar_size","width")][,.(array,N)],tmp[,.(array,N)])
aov_dt[,is_genome:="No"]
aov_dt[grep("Cast",array),is_genome:="Yes"]
results <- aov(N~array,data=aov_dt)
TukeyHSD(results)

plot_cast_dt[,.N,by=c("range","array","total_N","ar_size","ar_id")] %>%
  .[,mean(N),by=c("range","array","total_N","ar_size")] %>%
  ggplot() +
  geom_line(aes(x=range,y=V1,color=ar_size)) +
  facet_wrap(~array,scales="free_y") +
  theme_bw()


```
#relationship of ar_size and N_genes, ggridges
```{r}
library(ggridges)
library(colorspace)

options(scipen = 10^6)
vec <- plot_cast_dt[,.N,by=ar_id][N>30,ar_id]
ggplot(plot_cast_dt[ar_id%in%vec]) +
  geom_density_ridges(aes(x=log(width,base=10),y=array),scale=1)

a = ggplot(plot_cast_dt[width>500],aes(height = stat(density))) +
  theme_minimal() +
```

```
  geom_density_ridges(aes(x=width,y=array,fill=array),scale=2,alpha=0.5)  +
  scale_x_continuous(trans="log10",limits=c(500,100000)) +
  scale_fill_discrete_diverging() +
  theme(legend.position = "none")

b = ggplot() +
  geom_density_ridges(data=array_dt[width>500],aes(x=width,y=array,fill=array),scale=2,alpha=0.5)  +
  theme_minimal() +
  ylab("") +
  scale_x_continuous(trans="log10",limits=c(500,100000)) +
  scale_fill_discrete_diverging() +
  theme(legend.position = "none",
        axis.text.y = element_blank())



plot_cast_dt_densities <- plot_cast_dt %>%
  group_by(array) %>%
  group_modify(~ ggplot2:::compute_density(.x$width, NULL)) %>%
  rename(width = x)

ggplot(plot_cast_dt[width>500], aes(x = width, y = array, height = stat(density))) +
  geom_density_ridges(stat = "binline",bins=20,scale=1) +
    theme_minimal() +
  ylab("") +
  scale_x_continuous(trans="log10",limits=c(500,100000)) +
  scale_fill_discrete_diverging() +
  theme(legend.position = "none")

ggplot(plot_cast_dt_densities, aes(x = width, y = array, height = density)) +
  geom_density_ridges(stat = "identity") +
    theme_minimal() +
  ylab("") +
  scale_x_continuous(trans="log10",limits=c(1,100000)) +
  scale_fill_discrete_diverging() +
  theme(legend.position = "none")


as.data.table(plot_cast_dt_densities)[array=="Cast5"]

array_dt[width>500,.N,by=array]


dt <- plot_cast_dt[width>500,.N,by=c("array","width")] %>% .[,width_bin:=cut_width(width,width=1000)]
plot_cast_dt[,width_bin:=cut_width(width,width=1000)]
```

```
plot_cast_dt[,.N,by=ar_id] %>% ggplot() + geom_histogram(aes(N))
plot_cast_dt[,.N,by=ar_id][,summary(N)]
plot_cast_dt[,.N,by=c("ar_id","array")] %>% ggplot() + geom_col(aes(x=ar_id,y=N)) +
facet_wrap(~array,scales="free_x")
options(scipen = 10^6)
high_gene_arrays <- plot_cast_dt[,.N,by=ar_id][N>0,ar_id]
plot_cast_dt[width>500 & ar_id%in%high_gene_arrays] %>% ggplot() +
geom_density_2d_filled(aes(x=width,range),contour_var = "ndensity",geom="raster") +
  facet_wrap(~array) + theme_bw() + scale_fill_discrete_divergingx()  +
  scale_x_continuous(trans="log10",limits=c(500,100000))
high_gene_arrays <- plot_cast_dt[,.N,by=ar_id][N>71.50,ar_id]
plot_cast_dt[width>500 & ar_id%in%high_gene_arrays] %>% ggplot() +
geom_density_2d_filled(aes(x=width,range),contour_var = "ndensity") +
  facet_wrap(~array) + theme_bw() + scale_fill_discrete_divergingx()  +
  scale_x_continuous(trans="log10",limits=c(500,100000))
high_gene_arrays <- plot_cast_dt[,.N,by=ar_id][N>132.00,ar_id]
plot_cast_dt[width>500 & ar_id%in%high_gene_arrays] %>% ggplot() +
geom_density_2d_filled(aes(x=width,range),contour_var = "ndensity") +
  facet_wrap(~array) + theme_bw() + scale_fill_discrete_divergingx()  +
  scale_x_continuous(trans="log10",limits=c(500,100000))
```

Supplementary Code 6. Size profiles of Cast1-Cast9 satDNAs in both the assembly and raw reads

```{r}
source("blast_functions.R")
library(data.table)
library(Biostrings)
library(stringr)
library(tidyverse)
library(ggplot2)
```

cast array size distribution

```{r}

#not public reads
```

```
sat_cont_reads_cor <- blast_to_gff(q_name = "casts_19.fasta",s_name = "t_cast_20k.correctedReads.fasta",work_dir =
"E:/t_cast_assembly/assembly_analysis/",name="blast_sat_reads")


#fwrite(sat_cont_reads,"E:/t_cast_assembly/assembly_analysis/sat_cont_reads.tsv")

names <- unique(ext_table[,variable])
ext_fact <- c(250,250,250,250,250,1000,250,500,250,250)
ext_fac_dt <- data.table(names,ext_fact)

for (i in ext_fac_dt[,names])
{
 print(i)
 ext_factor <- ext_fac_dt[names==i,ext_fact]
 sat_copy <- copy(sat_cont_reads[feature==i])
 sat_copy[,enE:=end+ext_factor]
 result <- sat_copy %>% makeGRangesFromDataFrame() %>% reduce() %>% as.data.table()
 result[,array:=i]
 result[,width:=width-ext_factor]
 if (i =="Cast1")
 {
  array_dt_reads <- result
 }

 if (i !="Cast1")
 {
  array_dt_reads <- rbind(array_dt_reads,result)
 }

}


#if cast2 array has a cast2-prime within 170 bp beofre or after it is a Cast2-prime array
cast2_array_dt <- array_dt_reads[array=="Cast2"]
cast2_array_dt[,ar_id:=paste(seqnames,start,sep="_")]
cast2_array_dt[,c("start", "end") := .(start-170, end+170)]


cast_prime_array_dt <- array_dt_reads[array=="Cast2-prime"]
cast_prime_array_dt[,ar_id:=NULL]


setkey(cast2_array_dt,seqnames,start,end)
setkey(cast_prime_array_dt,seqnames,start,end)

ar_ids_with_cast_prime <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit() %>% .[,ar_id]
```

```
new_arrays <- foverlaps(cast2_array_dt,cast_prime_array_dt) %>% na.omit()

final_array_number <- new_arrays[,c("seqnames","start","end","i.end","ar_id","i.width","i.start")] %>%
 .[i.start<start,start:=start-i.width] %>%
 .[end>i.end,i.enE:=end] %>%
 .[end<i.end,i.enE:=i.end-170] %>%
 makeGRangesFromDataFrame(start.field = "start",
                          end.field = "i.end",
                          seqnames.field = "seqnames") %>%
 reduce() %>%
 as.data.table()

#if cast2-prime

array_dt_reads[array=="Cast2",ar_id:=paste(seqnames,start,sep="_")]


array_dt_reads[array=="Cast2" & ar_id%notin%ar_ids_with_cast_prime,array:="Cast2_pure"]

final_array_number[,array:="Cast2-mix"]

array_dt_reads <- rbind(array_dt_reads[array!="Cast2" & array!="Cast2-prime"],final_array_number,fill=TRUE)
array_dt_reads[width>330] %>% ggplot() + geom_histogram(aes(x=log(width,base=10),fill=array),bins=50) +
facet_wrap(~array,scales="free") + scale_fill_npg()
```

#cast array profiles grid arrange

```{r}

a=0
b=3000
c=600
labs <- paste0(seq(from=a,to=b/5,by=c/5))

breaks<- seq(from=a/5,to=b,by=c)
options(scipen=100000)
p1 <- array_dt_reads[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000)] %>% ggplot() +
 geom_histogram(aes(x=width,fill=array),bins=50) +
 facet_wrap(~array,scales="free",ncol=2) +
 scale_x_continuous(trans="log10") +
  scale_fill_npg() +
 theme_bw()
 #  scale_y_continuous(labels = mult_format(50))
```

```
p2 <- array_dt[(array!="Cast2-mix" & width>530) | (array=="Cast2-mix" & width > 2000)]  %>% ggplot() +
  geom_histogram(aes(x=width,fill=array),bins=50) +
  facet_wrap(~array,scales="free",ncol=2) +
  scale_x_continuous(trans="log10") +
  scale_fill_npg() +
  theme_bw()

grid.arrange(p1, p2, nrow = 1)
```

#LGX size comparison

```{r}
library(AICcmodavg)
array_dt <- fread("./data/full_array_annot.gff") %>% setnames(gff_colnames)
array_dt[,width:=abs(end-start)]
array_dt %>% ggplot() + geom_boxplot(aes(x=seqnames,y=log(width1,base=10),fill="ba")) + theme_bw() +
scale_fill_grey(start=0.7,end=0.7)

array_dt %>% ggplot() + geom_boxplot(aes(x=seqnames,y=width1,fill="ba")) + theme_bw() +
scale_fill_grey(start=0.7,end=0.7)
t.test(array_dt[seqnames!="LGX",log(width1,base=10)],array_dt[seqnames=="LGX",log(width1,base=10)])
wilcox.test(array_dt[width>350 & seqnames!="LGX",width1],array_dt[width>350 &seqnames=="LGX",width1])
array_dt[,ar_width := end-start]
l <- array_dt[,.N,by=seqnames][,array_per_mb := N*10^6/width]
array_dt[,c("seqnames","width1")] %>% as.tibble() %>% tbl_summary(.,by=c("width1"))
```

Supplementary Code 7. Creating ggbio and circos plots of the TcasONT assembly

```{r}
library(ComplexHeatmap)
library(data.table)
library(Biostrings)
library(stringr)
library(tidyverse)
library(ggplot2)

gff_colnames <- c("seqnames","source","type","start","end","score","strand","tag","name")

array_dt <- fread("assembly_analysis/full_array_annot.gff")  %>% setnames(gff_colnames)
```
#ggbio distribution of cast elements on differen chromosomes

```{r}
library(ggbio)

#namefix

array_dt[,levels:=as.numeric(str_extract(type,"\\d"))]
array_dt[,width:=end-start]
lev <- c("LG2",  "LG3",  "LG4",  "LG5",  "LG6",  "LG7",  "LG8",  "LG9",  "LG10","LGX")
array_dt[,seqnames:=factor(seqnames,levels=lev)]

#tmp <- rbind(array_dt[(array!="Cast2-mix" & width>100) | (array=="Cast2-mix" & width >
1000)],array_dt_sat[width>350],fill=TRUE)
array_dt[,levels:=as.numeric(str_extract(type,"\\d"))]


autoplot(makeGRangesFromDataFrame(array_dt,keep.extra.columns =
TRUE),layout="karyogram",aes(fill=type,color=type,
      ymin = (levels - 1) * 10/9, ymax = levels * 10/9))



autoplot(makeGRangesFromDataFrame(array_dt,keep.extra.columns = TRUE),
      layout="karyogram",
      aes(fill=type,color=type),alpha=1) +
 scale_fill_npg() +
 scale_color_npg() +
```

```
 theme(panel.grid = element_blank(),
       axis.ticks = element_blank(),
       axis.text.y = element_blank())
```

#circos

##funs

```{r}
windowize <- function (x,dt,len)
{
starts <- seq(from=dt[x,min],to=dt[x,max]-len,by=len)
   ends <- seq(from=dt[x,min]+len,to=dt[x,max],by=len)
      out_dt <- data.table(seqnames=dt[x,seqnames],start=starts,end=ends)
return(out_dt)}



#make windows from a data table start end function
diw_fun <-
function(dt_f,window=6,colvec=c("#4DBBD5B2","#DC0000B2" ),loged=FALSE,return_hits=FALSE,filter_hits=1000)
{
dt <- copy(dt_f)
chr_range <- chr_range_dt

i=window
win_size = 10^(i)
chr_range_glob_dt <- data.table()
for (i in 1:nrow(chr_range))

{
   chr_range_glob_dt <- rbind(chr_range_glob_dt,windowize(i,chr_range,win_size))
}

chr_range_glob_dt<<-chr_range_glob_dt

windows <- makeGRangesFromDataFrame(chr_range_glob_dt,ignore.strand = TRUE)

gr <- makeGRangesFromDataFrame(dt,keep.extra.columns = TRUE)

hits_dt <- windows[subjectHits(findOverlaps(gr,windows))] %>%
as.data.table(.) %>% .[,.N,by=c("seqnames","start","end")] %>% setnames(.,"N","hits")
 cols <- cbind(colo=colorRampPalette(colvec)(max(hits_dt[,hits])),hits=1:max(hits_dt[,hits])) %>%
            as.data.table %>%
```

```
                    .[,hits:=as.integer(hits)]

 if (loged == TRUE)
{
  hits_out <<-hits_dt

  cols <- cbind(colo=colorRampPalette(colvec)(max(log(hits_dt[,hits],base=2))-min(log(hits_dt[,hits]))),
           hits=min(log(hits_dt[,hits])):max(log(hits_dt[,hits],base=2)))%>%
          as.data.table %>%
                .[,hits:=as.integer(hits)]
  cols_out <- cols
  hits_dt[,hits:=round(sqrt(hits),0)]

  bed_out <-  merge(hits_dt,cols) %>% as.data.frame()


 }
if (return_hits==TRUE)
{
 bed <-  merge(hits_dt,cols)[hits>filter_hits]   %>%
   .[,hits:=NULL] %>%
   as.data.table()
 return(bed)
} else {
bed <-  merge(hits_dt,cols) %>% .[,hits:=NULL] %>% as.data.frame()
return(bed)
 }

}
```

```
#plotting
```{r}
ONT_assembly <- readDNAStringSet("./data/TcasONT.fasta")[1:10]


#loading genes
genes_ont <- fread("./data/TcasONT_genes.gff",fill=T,sep="\t")
genes_ont <- genes_ont[V3=="gene"]
setnames(genes_ont,c("V1","V4","V5"),c("seqnames","start","end"))
genes_ont[start>end, c("end", "start") := .(start, end)]
genes_ont <- genes_ont[start<end,c("seqnames","start","end")]
```
```

```
#loading repeat elements
repeats <- fread("./data/TcasONT_repeats.gff",fill=TRUE) %>%
  merge(.,name_links,by.x="seqnames",by.y="V7") %>% .[,seqnames:=V3] %>% .[,V3:=NULL]
```
```{r}

#init range dt
chr_range_dt <-
data.table(seqnames=str_remove(names(ONT_assembly),"_RagTag.+"),min=1,max=width(ONT_assembly))   %>%  mer
ge(.,name_links,by.x="seqnames",by.y="V7") %>%  .[,seqnames:=V3] %>% .[,V3:=NULL]

#sattetlies are ready

new_ch <- cbind("Empty_space",1,20000000) %>% data.table()

chr_range_dt_2 <- rbind(chr_range_dt,new_ch,use.names=FALSE)
chr_range_dt_2[,max:=as.double(max)]
chr_range_dt_2[,min:=as.double(min)]
col_fun = colorRamp2(c(0, 1), c("#4DBBD5B2","#DC0000B2"))
lgd = Legend(col_fun = col_fun, title = "Relative abundancy",at=c(0,1),labels = c("Low","High"))
grid.rect()
draw(lgd, x = unit(1, "cm"), y = unit(1, "cm"), just = c("left", "bottom"))
popViewport()
```
```{r}
TCAST_dt <- sat_cont_dt[grep("TCAST",feature)] %>% .[,c("seqnames","start","end")] #%>% .[grep("NC",seqnames)]
TCAST_dt[,seqnames:=str_remove(seqnames,"_RagTag")]
#TCAST_dt <- merge(TCAST_dt,name_links,by.x="seqnames",by.y="V7") %>% .[,seqnames:=V3] %>% .[,V3:=NULL]
TCAST_dt <- TCAST_dt[grep("LG",seqnames)]


cast_dt <- sat_cont_dt[grep("Cast",feature)][feature!="Cast7"] %>% .[,c("seqnames","start","end")]
#%>% .[grep("NC",seqnames)]
cast_dt[,seqnames:=str_remove(seqnames,"_RagTag")]
#cast_dt <- merge(cast_dt,name_links,by.x="seqnames",by.y="V7") %>% .[,seqnames:=V3] %>% .[,V3:=NULL]
cast_dt <- cast_dt[grep("LG",seqnames)]


lgd_LINE = Legend(at = c(-2,2),col_fun = )

draw(lgd, x = unit(1, "cm"), y = unit(1, "cm"), just = c("left", "bottom"))
```
```{r}
colorRampPalette(colvec)(10)
```

```
lev <- c("LG2", "LG3", "LG4", "LG5", "LG6", "LG7", "LG8", "LG9", "LG10","LGX")
array_dt[,seqnames:=factor(seqnames,levels=lev)]


circos.par("track.height"=0.1)
circos.genomicInitialize(chr_range_dt_2)
circos.genomicTrack(diw_fun(genes_ont,window = 6), stack = TRUE,
    panel.fun = function(region, value, ...) {
        i = getI(...)
        circos.genomicRect(region, value, ytop = i + 0.6, ybottom = i - 0.6,
            col = value$colo,border = value$colo, ...)
})
circos.genomicTrack(diw_fun(repeats[class!="Simple_repeat" & class!="Low_complexity"],loged = FALSE,window = 6),
stack = TRUE,
    panel.fun = function(region, value, ...) {
        i = getI(...)
        circos.genomicRect(region, value, ytop = i + 0.6, ybottom = i - 0.6,
            col = value$colo,border = value$colo,  ...)
})
circos.genomicTrack(diw_fun(array_dt,window = 6), stack = TRUE,
    panel.fun = function(region, value, ...) {
        i = getI(...)
        circos.genomicRect(region, value, ytop = i + 0.6, ybottom = i - 0.6,
            col = value$colo,border = value$colo,...)
})
hits_out
v1 <- diw_fun(genes_ont,window = 5,return_hits = FALSE) %>% .[order(seqnames,start)]
v2 <- diw_fun(array_dt,window = 6.5,return_hits = FALSE) %>% .[order(seqnames,start)]
v1[,merge_id:=paste(seqnames,start,sep="_")]
v2[,merge_id:=paste(seqnames,start,sep="_")]
v1 <- merge(v1,v2,by="merge_id")
v1 %>% ggplot() + geom_point(aes(x=hits.x,y=hits.y))
```

Supplementary Code 8. Monomer consensus and junction regions of Cast1-Cast9 satDNAs in the assembly

izvlačenje monomernih sekvenci castova s kromosoma

```{r}
library("dplyr")
library("data.table")
library("BSgenome")
library("msa")
library("rvcheck")
library("ggtree")
library("ape")
library("metablastr")
library(GenomicRanges)
source("blast_functions.R")
```

#extracting monomers from assembly

```{r}
sat_cont_dt <- blast_to_gff(q_name = "casts_19.fasta",s_name = "TcasONT.fasta",
                            work_dir = "./data/",name="blast_sat_assembly")

chroms <- readDNAStringSet("./data/TcasONT.fasta")


sat_cont_dt[,direction:="5-prime"]
sat_cont_dt[s_start>s_end,direction:="3-prime"]
sat_cont_dt[s_start>s_end, c("s_end", "s_start") := .(s_start, s_end)]
sat_cont_dt <- sat_cont_dt[grep("LG",subject_id)] %>% .[grep("Cast",query_id)]




for (i in unique(sat_cont_dt[,query_id]))
{
sat_cont_dt_temp <- sat_cont_dt[query_id==i]

i
crom_monomers <- sat_cont_dt_temp %>% makeGRangesFromDataFrame(seqnames.field = "subject_id",start.field =
"s_start",end.field = "s_end",keep.extra.columns = TRUE)

seqs <- getSeq(chroms[1:10],crom_monomers)

seqs[sat_cont_dt_temp$direction=="3-prime"]<-reverseComplement(seqs[sat_cont_dt_temp$direction=="3-prime"])
```

```
names(seqs) <- sat_cont_dt_temp[,paste(query_id,subject_id,s_start,sep="_")]

writeXStringSet(seqs,paste0("./data/phylogeny/",i,"_monomers.fasta"))
}
```
#extracting junction regions
```{r}
dt_width <- data.table(names=names(chroms),width=width(chroms))
arrays <- fread("E:/t_cast_assembly/assembly_analysis/full_array_annot.gff")
arrays <- merge(arrays,dt_width,by.x="V1",by.y="names")

arrays[,bef_start:=V4-499]
arrays[,bef_end:=V4]
arrays[,aff_start:=V5]
arrays[,aff_end:=V5+499]

arrays[bef_start<0,bef_start:=1]
arrays[bef_end<0,bef_end:=1]

arrays[aff_start>width,bef_start:=width]
arrays[aff_end>width,aff_end:=width]

gr_bef<- gr <- arrays %>% makeGRangesFromDataFrame(start.field = "bef_start",end.field = "bef_end",seqnames.field = "V1")
seqs_bef <- getSeq(chroms,gr_bef)
names(seqs_bef) <- arrays[,paste0(V3,"_",V1,"_",V4)]

gr_af<- gr <- arrays %>% makeGRangesFromDataFrame(start.field = "aff_start",end.field = "aff_end",seqnames.field = "V1")
seqs_aff <- getSeq(chroms,gr_af)
names(seqs_aff) <- arrays[,paste0(V3,"_",V1,"_",V4)]
for (i in unique(arrays[,V3]))
{
 tmp_seqsa_bef <- seqs_bef[grep(i,names(seqs_bef))]

 tmp_seqsa_af <- seqs_aff[grep(i,names(seqs_aff))]

 names(tmp_seqsa_bef) <- paste0(names(tmp_seqsa_bef),"_before")

 names(tmp_seqsa_af) <- paste0(names(tmp_seqsa_af),"_after")

 c(tmp_seqsa_bef,tmp_seqsa_af) %>%
writeXStringSet(paste0("E:/t_cast_assembly/assembly_analysis/junction_regions_revamp/",i,".500bp.around_regions.fasta"))
```

```
}
```
#distance matrices and heatmaps
```{r}
# load package
library(pheatmap)
library("ComplexHeatmap")
library(circlize)
library("multipanelfigure")
namevec <- c("Cast1","Cast2-mix","Cast2_pure","Cast3","Cast4","Cast5","Cast6","Cast7","Cast8","Cast9")
myplots <- list()
for(i in namevec)
{
dt <- fread(paste0("./data/matrices/",i,".matrix.csv")) %>% as.data.frame(row.names = "V1")
dt$V1 <- NULL
#dt=max(dt)-dt
col_fun = colorRamp2(c(0,50,100), c("#1A5276","#F4ED7E", "#AD3212"))
obj = paste0(i,"heatmap")
h1=Heatmap(as.matrix(dt),show_column_names = FALSE,col = col_fun,column_title=i,name=" ",
          heatmap_legend_param = list(
        title = "similarity", at = c(0, 50, 100)
    ))
myplots[[i]] <- h1
png(paste0("./data/matrices/",i,"_heatmap.png"),width=1024,height=1024)
draw(h1)
dev.off()
}
getwd()
figure1 <- multi_panel_figure(
  width = 350, height = 350,
  columns = 3, rows = 4,unit = "mm")


for (i in 1:10)
{
h1 <- myplots[[i]]
figure1 %<>% fill_panel(h1)
}

figure1
```
```{r}
dt <- readxl::read_xlsx("E:/Supplementary tables.xlsx",sheet = "12_K-S",skip = 1)

vec <- c(dt$`Significatly more genes`,dt$`Significantly less genes`)
```

```
cat(cbind(p.adjust(vec,method = "fdr")),sep="\n")
```

```
```

Supplementary Code 9. PCA and UMAP plots of Cast1-Cast9 satDNAs

```{r}
library(ape)
library(colorspace)
library(FactoMineR)
library(ggplot2)
library(dplyr)
library(data.table)


```
#PCA trees


```{r}

names <- c("Cast1","Cast2","Cast2-prime","Cast3","Cast4","Cast5","Cast6","Cast7","Cast8","Cast9")

for (i in names)
{
  print(i)

msa <- readDNAMultipleAlignment(paste0("./data/monomers/",i,".fasta.aligned"),
                                format="fasta")
print("done reading data")
x <- dist.dna(as.DNAbin(msa),model="F81",as.matrix=TRUE,pairwise.deletion=TRUE)
print("done distance")
matrix <- as.data.table(x,row.names="V1")
fwrite(matrix,paste0("assembly_analysis/phylogeny/consensus_monomers/",i,".aligned.matrix.csv"))
pca_res <- PCA(matrix)

print("PCA donte")
saveRDS(pca_res, file = paste0("./data/monomers/",i,".aligned.PCA.rds"))
}



dt_tot <- data.table()
eig_tot <- data.table()
for (i in names)
{
pca_res <- readRDS(paste0("./data/monomers/",i,".aligned.PCA.rds"))

dt <- pca_res$var$coord %>% as.data.table(keep.rownames = TRUE)
```

```
dt[,chr:=str_extract(rn,"LG(\\d+|X)")]
dt[,name:=i]
eigenvalues <- pca_res$eig %>% as.data.table(keep.rownames = TRUE)
eigenvalues <- eigenvalues[1:10]
eigenvalues[,name:=i]
eigenvalues[,val:=1:10]
eig_tot <- rbind(eig_tot,eigenvalues)
dt_tot <- rbind(dt_tot,dt)
}


qualitative_hcl(10,c=100)
dt_tot %>% ggplot() + geom_point(aes(x=Dim.1,y=Dim.2,color=chr,fill=chr),alpha=0.8,size=0.1)  +
  theme_bw() +
  scale_color_discrete_qualitative(c1=100) +
  facet_wrap(~name,scales="free",ncol=5) +
  xlab("PC1") +
   guides(color = guide_legend(override.aes = list(size = 3))) +
  ylab("PC2") + theme(legend.position = "none")



eig_tot[,mean(`percentage of variance`),by=c("name","val")] %>%
  ggplot() + geom_col(aes(x=as.factor(val),y=V1),color="black",fill="#4cb9d2") +
  xlab("Principal Component") +
  theme_bw() + facet_wrap(~name)
```

Supplementary Code 10. Graph network plots

````
---
title: "R Notebook"
output: html_notebook
---


```{r}
library(ape)
library(networkD3)
library(dplyr)
library(stringr)
library(data.table)

```


```{r}
chr_vec <- c("LG10","LG2","LG3","LG4","LG5","LG6","LG7","LG8","LG9","LGX")
vec <- qualitative_hcl(10,c=100)
cat(chr_vec,sep="\",\"")
cat(vec,sep="\",\"")
#change according to alignmet
alignment_path = "./data/filtered_monomers/Cast2_aligned.fasta"

var <- "Cast2"
msa <- readDNAMultipleAlignment(alignment_path,format="fasta")
matrix <- dist.dna(as.DNAbin(msa),model="F81",as.matrix=TRUE,pairwise.deletion=TRUE) %>% as.data.table()

i = "Cast2"
matrix <- fread(paste0("./data/filtered_monomers/",i,".aligned.matrix.csv"))
names <- colnames(matrix)
matrix <- cbind(names,matrix)
colnames(matrix) <- str_remove(colnames(matrix),paste0("_",var))
matrix[,names:=str_remove(names,paste0("_",var))]
matrix[,name_id:=str_extract(names,"LG(\\d+|X)_\\d+")]
dt <- melt(matrix,id.vars=c("names","name_id"))
dt[,var_id:=str_extract(variable,"LG(\\d+|X)_\\d+")]


ld_ar_dt <- dt[var_id!=name_id]
#calculate mean distances between arrays
````
````

````
ld_ar_dt[,mval:=mean(value,na.rm = TRUE),by=.(var_id,name_id)]
#find the closest array for each array
ld_ar_dt[,mmval:=min(mval,na.rm = TRUE),by=.(name_id)]
tmp <- unique(ld_ar_dt[order(mval)][,.(var_id,name_id,mval)])[, head(.SD, 5), by=.(var_id)]
tmp <- tmp[order(var_id)]
g <- igraph::graph_from_data_frame(tmp,directed=F )
p <- igraph_to_networkD3(g)
p$nodes$group = str_extract(p$nodes$name,"LG(\\d+|X)")
#p$links$value = 1/p$links$value
graph = forceNetwork(Links = p$links, Nodes = p$nodes, Source = 'source',
          Target = 'target', NodeID = 'name', Group = 'group', Value = "value",
          zoom = TRUE, linkDistance = 30,
          linkWidth = 1,
          arrows = FALSE,
          charge=-50,
          legend = TRUE, opacity = 0.8,
          colourScale=JS('d3.scaleOrdinal(["#F05E84","#AE9000","#66A200","#00AE48","#00B39C","#00ABD7","#3892F9"
,"#C16AF4","#EE50C9"],
            ["LG10","LG3","LG4","LG5","LG6","LG7","LG8","LG9","LGX"]);'),
          bounded = FALSE)


htmlwidgets::saveWidget(graph,file = paste0(var,".html"))


```
````
````

Supplementary Code 11. Rust code for generating the k-mer counting program using in edge detection, main.rs

```
use bio;

use std::{error::Error, str::FromStr};
use std::fs::File;
use bio::io::fasta;
use std::fs;
use debruijn::*;
use debruijn::kmer::*;
use ndarray::{ Array2,ArrayBase,OwnedRepr,Dim};
mod utils;
mod ploting;
use ploting::plot_roll_mean;
use std::io::Write;
use rayon::prelude::*;
use log::info;
use env_logger;



fn process_fasta(sequence_path: &str, monomer_path: &str,outpath: &str) -> Result<(), Box<dyn Error>> {

    let file = File::open(sequence_path)?;
    let reader = fasta::Reader::new(file);

    info!("Processing {} -> {}",sequence_path,monomer_path);
    let kmers_in_monomer = utils::create_kmers_from_sat(monomer_path).unwrap();


    reader.records().par_bridge().for_each(|result| {

        let record = result.expect("Error during fasta record parsing");

        let kmers = Kmer32::kmers_from_ascii(record.seq());

        let mut dist_mat: ArrayBase<OwnedRepr<u32>,
         Dim<[usize; 2]>> = Array2::zeros((kmers.len(), kmers.len()));
        let mut kmer_in_array_pos_dist: Vec<i32> = Vec::new();

        for i in 0..kmers.len() {

         let mut dist_vec: Vec<u32> = Vec::new();
```

```
        for z in 0..kmers_in_monomer.len()-1 {

          let dist = kmers[i].hamming_dist(kmers_in_monomer[z]);
          dist_vec.push(dist)

        }

        let min = dist_vec.iter().cloned().min().unwrap();
        kmer_in_array_pos_dist.push(min as i32)

      }

      let roll_mean: Vec<f64> = utils::calculate_means_around_index(&kmer_in_array_pos_dist);

      let _ = plot_roll_mean(&record,roll_mean.clone(),outpath);


      //writing the kmer tables, both the pos in array and the roll mean
      let outf = outpath;
      let outfn = outf.to_owned()+ "/data/" + &record.id().to_string().to_owned() + "_kmers_in_mono.txt";
      let mut file = File::create(outfn).unwrap();
      writeln!(file, "index\tactual\troll_mean").unwrap();
      for (i, (elem1, elem2)) in kmer_in_array_pos_dist.iter().zip(roll_mean.clone().iter()).enumerate() {
        writeln!(file, "{}\t{}\t{}", i, elem1, elem2).unwrap();
      }


    }
);
    Ok(())

}


fn init_logger() {
    // Read the RUST_LOG environment variable or use a default log level
    let log_level = std::env::var("RUST_LOG").unwrap_or_else(|_| String::from("info"));

    // Initialize the logger with the specified log level
    env_logger::Builder::from_default_env()
        .filter_level(log::LevelFilter::from_str(&log_level).unwrap())
        .init();
}
```

```rust
fn remove_all_files_and_folders_in_folder(folder_path: &str) -> std::io::Result<()> {
    // Read directory entries and remove each file or subfolder
    for entry in fs::read_dir(folder_path)? {
        let entry = entry?;
        let path = entry.path();

        if path.is_file() {
            fs::remove_file(&path)?;
            println!("Deleted file: {:?}", path);
        } else if path.is_dir() {
            fs::remove_dir_all(&path)?;
            println!("Deleted folder: {:?}", path);
        }
    }

    Ok(())
}

fn main() {
    init_logger();

    let folder_path: &str = "./results/kmer_analysis";
    let data_path: &str = "./results/kmer_analysis/data/";
    let pic_path: &str = "./results/kmer_analysis/pictures/";


    // Create the folder if it doesn't exist
    if let Err(err) = fs::create_dir(folder_path) {
        if err.kind() != std::io::ErrorKind::AlreadyExists {
            eprintln!("Error creating folder: {:?}", err);
            return;
        }
    }

    // Check if the folder is not empty
    let is_empty = fs::read_dir(folder_path)
        .map(|entries| entries.count() == 0)
        .unwrap_or(true);

    if !is_empty {
        // Delete all files in the folder
        if let Err(err) = remove_all_files_and_folders_in_folder(&folder_path) {
            eprintln!("Error deleting files: {:?}", err);
            return;
        }
```

```
        info!("All files in the folder have been deleted.");
    } else {
        println!("The folder is empty.");
    }


    // create new data and folder paths
    if let Err(err) = fs::create_dir(data_path) {
        if err.kind() != std::io::ErrorKind::AlreadyExists {
            eprintln!("Error creating folder: {:?}", err);
            return;
        }
    }

    if let Err(err) = fs::create_dir(pic_path) {
        if err.kind() != std::io::ErrorKind::AlreadyExists {
            eprintln!("Error creating folder: {:?}", err);
            return;
        }
    }
    let json_file = "pairs.json";

    // Read the JSON file
    let pairs = utils::read_json_file(json_file);

    // Print the Monomer-Array Pairs
    info!(
        "Doing edge finding for the following RU:Array pairs: \n {}",
        pairs.iter()
            .map(|(monomer, array)| format!("{} -> {}", monomer, array))
            .collect::<Vec<String>>()
            .join("\n")
    );
    pairs.iter().for_each(|(monomer, array)| {
        let _ = process_fasta(array,monomer, folder_path);
    });

}
```

Supplementary Code 12.Rust helper reader function code for generating the k-mer counting program using in edge detection, utils.rs

```
use bio;
use serde_json::Error;

use std::fs::File;
use bio::io::fasta;
use debruijn::*;
use debruijn::kmer::*;
use std::io::{self, Write};
use std::collections::HashMap;
use std::io::BufReader;
use serde::{Deserialize, Serialize};
use std::io::{Read};


//function creates a hash table of all kmers in a sattellite, will be expanded into iterator over
// multiple fasta files
pub fn create_kmers_from_sat(mononomer_path: &str) -> Result<Vec<IntKmer<u64>>, std::io::Error>{


    let fasta_file = File::open( mononomer_path)?;

    let mut kmer_total: Vec<IntKmer<u64>> = Vec::new();

    let reader = fasta::Reader::new(fasta_file);
    for result in reader.records() {

        let record = result.expect("Error during fasta record parsing");

        let newseq = Vec::from_iter(record.seq().iter().cloned().chain(record.seq().iter().cloned()));
        let k32_tmp = Kmer32::kmers_from_ascii(&newseq);

        'outer: for i in k32_tmp.iter().to_owned() {
            for j in kmer_total.iter().to_owned(){

                if i.hamming_dist(*j)==0 {

                    break 'outer;
                }
            }
            kmer_total.push(*i)
```

```
        }
        }


        println!("\nNumber of unique kmers: {} in {}\n",kmer_total.len(),mononomer_path);
        return Ok(kmer_total)
    }



#[derive(Debug, Deserialize, Serialize)]
struct MonomerArrayPair {
        monomer_path: String,
        array_path: String,
}


pub fn get_monomer_array_pairs() -> Result<HashMap<String,String>,std::io::Error> {
        let json_file = "pairs.json";

        // Read the JSON file
        let pairs = read_json_file(json_file);

        // Print the Monomer-Array Pairs
        println!("Monomer-Array Pairs:");
        for (monomer_path, array_path) in &pairs {
            println!("{} -> {}", monomer_path, array_path);
        }
        return Ok(pairs)
    }


pub fn calculate_means_around_index(data: &[i32]) -> Vec<f64> {
        let mut means = Vec::new();

        for i in 10..(data.len() - 10) {
            let sum: i32 = data[i - 5..=i + 10].iter().cloned().sum();
            let count = 21.0; // Count of elements in the range [i - 5, i + 5]
            let mean = f64::from(sum) / count;

            means.push(mean);
        }

        means
```

```
}


pub fn read_json_file(json_file: &str) -> HashMap<String, String> {
    // Open the JSON file
    let file = File::open(json_file).expect("Failed to open JSON file");
    let reader = BufReader::new(file);

    // Deserialize the JSON content into a HashMap<String, String>
    let pairs: HashMap<String, String> = serde_json::from_reader(reader).expect("Failed to deserialize JSON");

    pairs
}
```

Supplementary Code 13.Rust plotting functions code for generating the k-mer counting program using in edge detection, plotting.rs

```
use bio;
use serde_json::Error;

use std::fs::File;
use bio::io::fasta;
use debruijn::*;
use debruijn::kmer::*;
use std::io::{self, Write};
use std::collections::HashMap;
use std::io::BufReader;
use serde::{Deserialize, Serialize};
use std::io::{Read};


//function creates a hash table of all kmers in a sattellite, will be expanded into iterator over
// multiple fasta files
pub fn create_kmers_from_sat(mononomer_path: &str) -> Result<Vec<IntKmer<u64>>, std::io::Error>{


    let fasta_file = File::open( mononomer_path)?;

    let mut kmer_total: Vec<IntKmer<u64>> = Vec::new();

    let reader = fasta::Reader::new(fasta_file);
    for result in reader.records() {

        let record = result.expect("Error during fasta record parsing");

        let newseq = Vec::from_iter(record.seq().iter().cloned().chain(record.seq().iter().cloned()));
        let k32_tmp = Kmer32::kmers_from_ascii(&newseq);

        'outer: for i in k32_tmp.iter().to_owned() {
            for j in kmer_total.iter().to_owned(){

                if i.hamming_dist(*j)==0 {

                    break 'outer;
                }
            }
            kmer_total.push(*i)
```

```
        }
        }


        println!("\nNumber of unique kmers: {} in {}\n",kmer_total.len(),mononomer_path);
        return Ok(kmer_total)
    }



#[derive(Debug, Deserialize, Serialize)]
struct MonomerArrayPair {
        monomer_path: String,
        array_path: String,
}


pub fn get_monomer_array_pairs() -> Result<HashMap<String,String>,std::io::Error> {
        let json_file = "pairs.json";

        // Read the JSON file
        let pairs = read_json_file(json_file);

        // Print the Monomer-Array Pairs
        println!("Monomer-Array Pairs:");
        for (monomer_path, array_path) in &pairs {
            println!("{} -> {}", monomer_path, array_path);
        }
        return Ok(pairs)
    }


pub fn calculate_means_around_index(data: &[i32]) -> Vec<f64> {
        let mut means = Vec::new();

        for i in 10..(data.len() - 10) {
            let sum: i32 = data[i - 5..=i + 10].iter().cloned().sum();
            let count = 21.0; // Count of elements in the range [i - 5, i + 5]
            let mean = f64::from(sum) / count;

            means.push(mean);
        }

        means
    }
```

```rust
pub fn read_json_file(json_file: &str) -> HashMap<String, String> {
    // Open the JSON file
    let file = File::open(json_file).expect("Failed to open JSON file");
    let reader = BufReader::new(file);

    // Deserialize the JSON content into a HashMap<String, String>
    let pairs: HashMap<String, String> = serde_json::from_reader(reader).expect("Failed to deserialize JSON");

    pairs
}
```