universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Genome informatics and insights into the evolution of complex genomes“

verfasst von / submitted by

Dipl.-Ing. Michael Zach, BSc MSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 829
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Evolutionsbiologie
Betreut von / Supervisor:	Ass.-Prof. Dr. Oleg Simakov

Kurzfassung

Ein großer Anteil eukaryotischer Genome besteht aus repetitiven DNA-Sequenzen, darunter auch Transposonen, also sich selbst verschiebende und kopierende Sequenzen. Neue Erkenntnisse deuten darauf hin, dass mobile Elemente nicht nur zerstörerisch wirken, sondern der Evolution als Reservoir für neue Gene und Regulationsmechanismen dienen. Detektion, Annotation und Datierung von sich wiederholenden Sequenzen sind jedoch nicht-triviale Probleme der Genom-Informatik. Meine Arbeit beschäftigt sich mit eben diesen drei Gebieten.

Zuerst teste ich verschiedene Programme auf ihre Eignung zur de-novo Erkennung von repetitiven Sequenzen hin. Dazu habe ich das Genom von *Hydra vulgaris*-herangezogen, einem wichtigen Modellorganismus. Die Erkennungsrate anhand eines Assembly war am besten bei der Verwendung von RepeatModeler. Bei kurzen Illumina Reads zeigte dnaPipeTE trotz schlechter Abdeckung eine gute Leistung, unter besseren Bedingungen könnte es RepeatModeler womöglich übertreffen. Maschinelles Lernen mit Red blieb hingegen hinter den Erwartungen zurück. Als weiteres Ergebnis zeige ich, dass Ähnlichkeit zwischen Konsensussequenz-Bibliotheken keine ähnlichen Ergebnisse beim Nachweis tatsächlicher Repeats im Genom impliziert.

Darüber hinaus hinterfragt meine Arbeit kritisch die landläufige Verwendung von Konsensussequenzen, um das Alter repetitiver DNA zu schätzen. Als Alternative dazu führe ich eine völlig neue, phylogenetische Methode ein. Evolutionäre Ereignisse werden mittels eines Koaleszenzbaumes rekonstruiert und ebenso datiert. Die Ergebnisse offenbaren jedoch Mängel in beiden Ansätzen, geben aber gleichzeitig auch Ansatzpunkte für künftige Verbesserungen dieser phylogenetischen Datierungsmethode für Repeats.

Zur visuellen Annotation von Genomen habe ich außerdem eine Instanz des UCSC Genome Browser installiert mit *Hydra vulgaris* als Referenzgenom und entsprechenden Annotationen, darunter auch von verschiedenen Programmen identifizierte Repeats. Die Untersuchung der Regionen um Wnt-Gene bestätigte meine Vermutung, dass solche Nachbarschaften frei von repetitiven Sequenzen sind. Offenbar wird dies durch selektiven Druck gegen diese essentiellen Gene störende Repeats verursacht. Mein Genombrowser steht online für den produktiven Einsatz durch Kollegen und Forschungspartner zur Verfügung.

Schlagworte: repetitive DNA, Transposon, Datierung, Genom-Browser, Hydra

Abstract

Large proportions of eukaryotic genomes comprise of repetitive DNA sequences, among them transposable elements capable of moving and copying themselves. Formerly seen predominantly as disruptive, emerging evidence suggests that mobile elements have been serving as a reservoir for novelties of genes and their regulatory mechanisms throughout evolutionary history. However, detection, annotation and dating of repeats are non-trivial problems of genome informatics. In my work, I focused on these three aspects.

First, I evaluated tools for their capabilities in de-novo detection of repeats. Therefore, I used the highly repetitive genome of important model organism *Hydra vulgaris*. Detection rate on an assembly was best with RepeatModeler. However, dnaPipeTE performed well on short Illumina reads despite their low coverage, which suggests it could potentially outperform RepeatModeler. Machine learning approach Red fell short of expectations to detect repeats that no other tool could find. Apart from assessing detection accuracy of individual tools, my work provides an important outcome concerning repeat libraries built with different detection tools. Consensus sequences may be generally similar between such libraries. Nevertheless, they may detect largely complementary subsets of the full spread of repeats in the genome. It is therefore important to use a diversified collection of tools for optimum detection results.

Furthermore, my work critically questions the commonly accepted use of consensus sequences for the dating of repeat copies. As an alternative, I introduce a completely new phylogenetic method that utilizes a coalescence tree to reconstruct evolutionary events and estimates their age in this tree. However, results disclose shortcomings in both approaches but also suggest starting points for future improvements of the phylogenetic dating method.

To provide visual genome annotations, I installed an instance of UCSC Genome Browser and added *Hydra vulgaris* reference genome and various annotation tracks, such as repeats detected by various tools. Inspection of regions around important developmental genes (e.g. Wnt ligands) confirmed the expectation to find such neighbourhoods free of repeats, presumably caused by selective pressure against interferences in the regulation of these essential genes. The genome browser is available online for productive use by colleagues and research partners and open to the community.

Keywords: repetitive DNA, transposable element, repeat dating, genome browser, Hydra

Table of Contents

Introduction

> *"It has been known since antiquity that, like begets like, more or less. Cats have kittens, dogs have puppies, and acorns grow into oak trees. ...*
>
> *As the programs in the genome are decoded and executed, the single cell becomes either a cat, or a dog, or an oak tree, according to the information stored in the genes. ...*
>
> *Genome projects are essentially efforts to obtain and reverse engineer the DNA code-script of humans and other species. Sequencing the genome, then, is equivalent to obtaining an image of a mass storage device. Mapping the genome is equivalent to obtaining a file allocation table for the device. Understanding the genome, when it happens, will amount to reverse engineering the genetic programs all the way back to design and maintenance specs."*
>
> (Robbins et al. 1995)

The precursory quote reflects the great expectations during the Human Genome Project. For many people, evangelizing journalists and participating scientists alike, it was self-evident that when only the genomic sequence of *Homo sapiens* was obtained completely, we would just have to read and decode it to fully understand it. Healing of major diseases, uncovering human origin and even unveiling the mystery of life itself seemed to be just within reach. From this perspective, we may even excuse the somewhat exalted usage of the computer metaphor.

Ever since, people have been using the predominant technological achievement of their time to elucidate the otherwise inexplicable. Nowadays, we see computers as the primary reference in popular and – unsettingly – even academic science, when it comes to explain the mind and the brain or the unfolding of life from sequences of molecules.

Twenty years after, we know that it is not enough to have the source code to understand what a complex computer program really does – modern malware is a good example for this. Now, we do understand that a genome is not a computer program. Nevertheless, this work unfolds at the intersection of both – genomics and informatics …

As a field of computational molecular biology, genome informatics uses computer software to denote observations, manage genomic data and solve problems such as the identification of functional elements, the comparison of structures and the prediction of genes.

In this work, I use methods and tools from genome informatics to gain insights into the evolution of complex genomes. For this purpose, I focus on repetitive elements in the genome of Hydra and examine the approaches to detect and to classify them. Furthermore, I introduce a prominent genome browser to visualize genomic annotations and compare two methods to estimate the time scale of repeat expansions.

Repetitive DNA elements

When compared to prokaryotes, only a small proportion of eukaryotic genomes consists of genes coding for proteins. The remaining non-coding DNA is typically classified into four groups

- introns that interrupt protein coding genes and that are spliced from mRNA transcripts to only keep exon regions for translation
- genes encoding RNAs of various molecular functions
- repeats, i.e. sequence patterns that occur in multiple copies throughout the genome
- intergenic regions with currently only some functions known, such as enhancers

As the functions of these genomic portions were undetermined as well as underestimated for a long time, they have been called “junk DNA”. This term had been first used in (Ehret & De Haller 1963) and was later popularized by geneticist Susumu Ohno.

In the course of time, the functional relevance of non-coding DNA was unveiled little by little. About 40 years later, Makaowski writes “I would like to argue that even if we cannot define the specific function of these elements, we still can show that they are not useless pieces of the genomes. […They] should be called genomic scrap yard rather than junk DNA.” (Makaowski 2000). This apparently slightly change in terminology nevertheless reflects new insights into how repeats are not only useless remnants but among others generate allelic diversity (Feschotte & Pritham 2007) and function as a supply of regulatory elements such as transcriptional silencers (Dorer & Henikoff 1994) and enhancers (Feschotte 2008; Chuong et al. 2017).

Heterochromatin, the highly condensed portion of chromosomes that is almost completely comprised of non-coding DNA, was long seen as primarily inactive (Redi et al. 2001). Later results suggested its manifold influence on gene regulation and overall genomic organisation. Finally, gene activity was located in putative heterochromatic chromosome regions (Yasuhara & Wakimoto 2006), hence chromatin states are today understood to be highly dynamic. Recent technological advances, such as 3C (Dekker et al. 2002), Hi-C (van Berkum et al. 2010) and ATAC-seq (Buenrostro et al. 2015) begin to shed light on how open chromatin regions arise during development and that they play an important role in transcription.

The aim of this work is to bring repetitive elements into focus, as they comprise significant portions of complex eukaryotic genomes (Richard et al. 2008). These multi-copy segments consist of repeated nucleotides of various lengths and are typically scattered randomly throughout the genome (Jurka 1998). However, certain hotspot regions appear to be more prone to repeat insertions than other parts of the genome. For instance, centromeres and telomeres comprise of such accumulations of repeats. Therefore, repeats can be of significant influence on chromosomal rearrangement, as they may serve as substrates for recombination events and the duplication or deletion of chromosomal segments (Myers et al. 2005; McVean 2010). Identifying repetitive DNA elements is consequently essential for the research of eukaryotic genomes. Comparative analysis of repeats from different species has demonstrated their usefulness as “molecular fossils” in evolutionary studies.

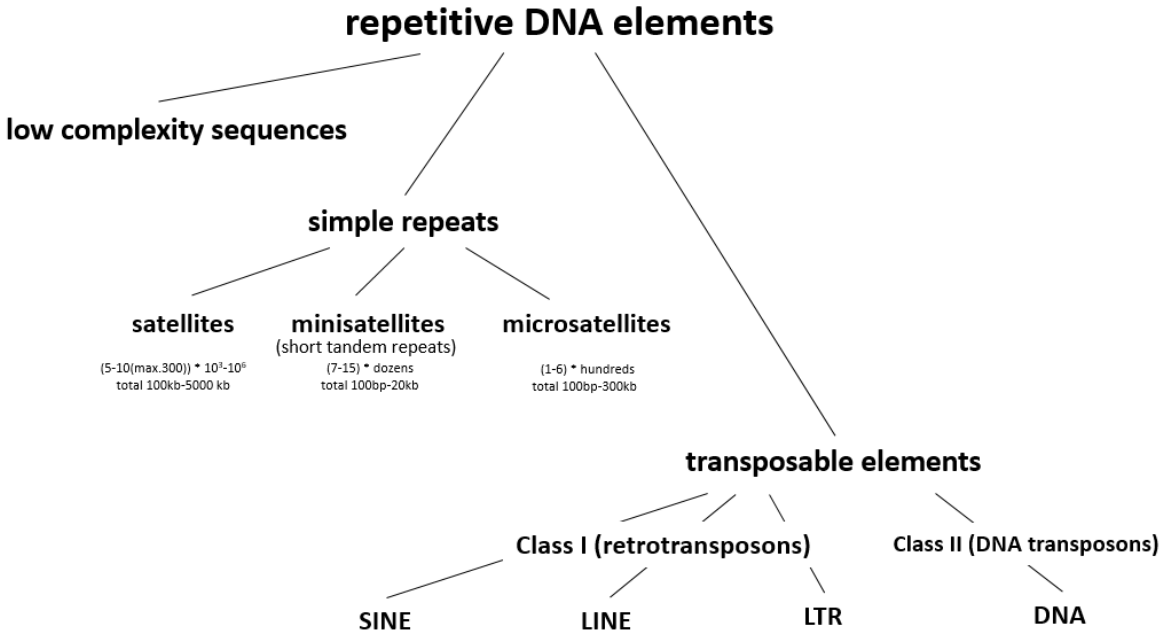


Figure 1. **Classification of repeats**.

Figure 1 provides an overview of repetitive element classes, differentiated by their structure as well as duplication and proliferation mechanisms (Jurka 1998; Kidwell 2002; Kapitonov & Jurka 2008).

I have intentionally omitted certain DNA elements here that I do not consider as repetitive in the narrower sense. Various RNA genes may be present in numerous copies throughout the genome (Ide et al. 2010). Segmental duplications originate from a partial chromosomal region duplication during unequal crossing over (Szostak & Wu 1980) and usually occur in two copies only. Nevertheless, the high sequence identity of copies makes it hard to correctly attribute reads during assembly and to work out differences between such segments (She et al. 2004). Furthermore, I left out pseudogenes that are a by-product of many genomic processes, such as the degeneration of duplicated genes (Harrison & Gerstein 2002).

Low complexity sequences

Such sequences typically consist of poly-purine or poly-pyrimidine stretches, or regions of either extremely high AT or GC content.

The term DNA sequence complexity has thereby changed from its original meaning. The original usage relates to studies applying reassociation kinetics to measure how repetitive or unique a genome was. As repetitive sequences tend to reassociate faster, the complexity of a genome was measured as the time at which half of the first denatured DNA was reassociated (Britten & Kohne 1968).

The more modern usage of the term tries to loosely apply it to actual DNA sequences, with the same notion that more complex sequences have a higher degree of uniqueness, and vice versa. Here, the term is mostly grounded on Shannon entropy (Schmitt & Herzel 1997) while more recent approaches use concepts from theoretical computer science, such as context-free grammars (Liou et al. 2013).

Low complexity repeats are of practical importance to bioinformatics, as many otherwise unrelated sequences in databases contain such regions and spurious similarities will be reported when aligning such sequences (Altschul et al. 1994). Therefore, it is very important to mask low complexity repetitive DNA sequences to prevent bias of alignments and database search results. For instance, BLAST search uses programs SEG (Wootton & Federhen 1996) and DUST (Morgulis et al. 2006) for this task.

Simple repeats

Simple Sequence Repeats (SSRs) show the numerous reiteration of short sequences with repetitions directly adjacent to each other. Examples for such repeats are $(TAA)_n$, $(CA)_n$, or $(TTAGGG)_n$, where n represents the number of iterations of each base pattern. Slippage of DNA polymerase during replication has been identified as responsible for the creation of such repetitive patterns (Richard et al. 2008).

Depending on the length of the pattern, simple repeats are further sub-divided into microsatellites (typically patterns from 1 to 6 bp long with up to several hundreds of adjacent repetitions) and minisatellites (typically patterns from 7 to 15 bp long with total lengths up to 20 kb). As the number of reiterations significantly varies between individuals of the same species, these structures additionally serve as genetic markers.

Simple repeats are sometimes referred to as “short tandem repeats” (STRs). This appears to be a somewhat misleading term, because it wrongly suggests grouping them together with so called “long tandem repeats” (LTRs), a class of repeats that originate from transposon activity described in more detail below. Such STRs may occur in special loci, for instance in centromeres and telomeres of chromosomes. In centromeres they are known to serve as attachment point for microtubules of the spindle apparatus responsible for the segregation of chromosomes during meiosis and mitosis (Choo 2001).

Transposable elements

I am following one of the most recent definitions here. “Transposable elements are discrete segments of DNA capable of moving within a host genome from one chromosome or plasmid location to another, or between hosts by using parasitic vectors that they use for lateral transfers.” (Piégu et al. 2015)

These mobile DNA elements had been first discovered by Barbara McClintock during her research on mutations in maize (McClintock 1950). As their name indicates, these DNA elements may change their positions within the genome and re-insert themselves at arbitrary loci. Thereby they possibly interrupt functional regions, such as open reading frames and regulatory sequences. Transposable elements may duplicate and proliferate in several copies inserted into other previously contiguous sequences. Therefore, they are referred to as interspersed repeats, in contrast to classes of mostly adjacent repeats I described above.

Transposon activity may interfere with molecular functions and regulatory mechanisms of the cell. As a consequence, it may directly impact actual fitness, cause unbalanced growth and subsequently induce diseases, such as cancer and autoimmunity, or reduce the fertility of progeny (Levin & Moran 2011; Elbarbary et al. 2016). Transposon insertion events into functional genes may lead into formation of pseudogenes. These inactive copies may be distinguished by interrupts in otherwise open reading frames, such as frameshifts or stop codons. Nevertheless, numerous studies have demonstrated the possible functional role of such pseudogenes among others in gene regulation and RNA stability (Castillo-Davis 2005; Pavlicek et al. 2006).

Peaceful coexistence with mobile elements in the genome depends upon adaptive control mechanisms. Therefore, different species have evolved various repression mechanisms, such as promoter methylation (Slotkin & Martienssen 2007) and small RNAs (Malone & Hannon 2009). In germ line cells of many animal species, PIWI-interacting RNAs (piRNAs) play an important role in silencing of retrotransposons to protect genome integrity of the offspring (Siomi et al. 2011).

However, emerging evidence suggests that mobile DNA can sometimes benefit host genomes by serving as a reservoir for new cellular functions. TE loci encode various enzymes as well as structural and envelope proteins that were adopted by their host genomes on several occasions during evolution of most eukaryotic lineages (Volff 2006). Recently has been argued that “the inherent genetic properties of TEs and the conflicting relationships with their hosts facilitate their recruitment for regulatory functions in diverse genomes. […] supporting the long-standing hypothesis that the waves of TE invasions endured by organisms for eons have catalysed the evolution of gene-regulatory networks.” (Chuong et al. 2017)

In addition, under stress conditions, transposable elements can influence the actions of regular genes and restructure the genome on various levels in potentially beneficial ways, a finding McClintock first presented in her Nobel lecture (McClintock 1984).

Over time, most TEs have accumulated mutations and truncation events within their own sequences, rendering them as incompetent for further transposition. However, in most genomes, some TE families remain active and continue to generate new insertion events (Huang et al. 2012).

Traditionally, TEs are categorized into two main classes, based on their intermediate substrate propagating insertions (RNA vs. DNA). Subsequently, they are further split up into families depending on various structural features (Finnegan 1989). Later on, four categories have been inserted (Kidwell 2002).This classification scheme has then been extended and formalized by (Wicker et al. 2007) and is mostly used throughout the community.

Class I of transposable elements are transposons that copy their sequence into RNA intermediates. These are then reverse-transcribed back into DNA at insertion loci by a TE-encoded reverse transcriptase. Therefore, these TEs are referred to as retrotransposons. With each replication cycle, one new copy of the RNA transposon is produced. Consequently, this class is the major contributor to the repetitive proportion of large genomes and responsible for exponential growth of repeat copies.

In contrast, with class II transposable elements no such retro-transposition is taking place. The transposon sequence is cut out or copied into a DNA intermediate that then is directly inserted at an arbitrary locus.

As previously stated, interspersed repeats are further differentiated into four categories, of which the first three are subclasses of RNA-mediated elements, while the forth is identical to class II of DNA-mediated elements.

Long-terminal-repeat (LTR) retrotransposons are distinguished by long terminal repeats (LTRs) of a few hundred base pairs up to 5 kb in length. These LTRs are directly flanking the internal coding region for reverse transcriptase, integrase and RNAse as well as structural proteins. From an evolutionary perspective, LTR retrotransposons are closely related to retroviruses, but do not share their envelope proteins and infectious particles.

Long interspersed elements (LINEs) do not contain flanking LTRs and are up to several kb in length. Their coding region typically contains an endonuclease, a reverse transcriptase and possibly further proteins. Several superfamilies of LINEs may contain an additional second open reading frame with mostly unknown function. LINEs are autonomous and contain all enzymes needed to complete their replication cycle. (Ostertag & Kazazian 2005)

In contrast, Short interspersed elements (SINEs) are non-autonomous TEs. They lack coding regions for a reverse transcriptase and other proteins typically found in LTRs and LINEs. SINEs are short, typically only 150 to 200 bp of length, and mainly consist of a polymerase III

promoter. It may autonomously initiate their transcription, using the enzymatic machinery of mostly LINEs to be transposed. (Kramerov & Vassetzky 2005)

The final category of DNA transposons is an equivalent to class II TEs. A DNA intermediate is used for transposition. The number of strand cuts distinguishes two subclasses. The first subclass cuts two strands during transposition, the intermediate is a double stranded DNA element that is pasted into the target locus. The structure consists of terminal inverted repeats (TIRs) flanking a coding region with a transposase and an optional reading frame for some superfamilies. Replication of these TEs is non-autonomous, during chromosome replication they can transpose from an already replicated position into one that still needs to be replicated (Greenblatt & Brink 1962).

The other subclass of DNA transposons cuts only one strand of DNA during transposition. This cut out single strand is assumed to immediately roll into a circle after insertion at the target site and undergo replication, while the gap region at the donor site is filled in by a replicase (Kapitonov & Jurka 2007). DNA transposons are supposed to have had major influence to the evolution of eukaryotic genomes (Feschotte & Pritham 2007).

However, I have to omit numerous details of transposable element classification here. There are some competing approaches to this simplified classification, but none of these considers the full diversity and complexity of TEs. A universal approach is needed to integrate understudied elements, such as introns and self-splicing elements (Piégu et al. 2015; Chuong et al. 2017). Furthermore, it is important to understand that this classification system was intended for naming, grouping and reference purposes only and does not imply common origin of any group above the family level. Quite the contrary, many groups of transposable elements may show similar structure but have undergone distinct evolutionary development.

Hydra as model organism

Hydra is a genus of class Hydrozoa and phylum Cnidaria living as solitary polyps in bodies of freshwater throughout temperate and tropical regions of the world. Most Hydra species are inconspicuously small but still visible to the naked eye, whereas only few grow up to about 3 cm in size. The radially symmetric and hollow body is formed as a tube or sac (Shimizu et al. 2007) and differentiated along the longitudinal axis. The blind end forms a basal disc that secretes an adhesive and provides attachment to the substrate. The free end of the body shows a mouth opening with a circle of filiform tentacles around it. Hydra is diploblastic, i.e. it

has two body layers. The outer epidermis is separated from the inner gastrodermis by a gel-like mesoglea. Excretion and respiration occur by diffusion through the epidermis. Hydra has no true muscles and is sessile in general. Nevertheless, it may retract tentacles, constringe the body and relocate within short ranges by using its tentacles and mouth. The polyp has sensory receptors and a simple nervous system. Hydra mainly feed on aquatic invertebrates that they catch with harpoon-like cnidocytes covering their tentacles. Unusual for Hydrozoa, members of this genus do not show a medusa phase. Gametes develop in epidermal evaginations of the polyp. In addition to sexual reproduction, many species of Hydra may also reproduce asexually by buds growing in the body wall and breaking away on maturity. (Westheide et al. 2013; Leclère et al. 2016)

Numerous properties make Hydra especially interesting to biologists. Hydra shows extraordinary regenerative abilities (Holstein et al. 2003) and apparently does not age or die of age (Martínez 1998). Furthermore, Hydra is considered the sister group to bilaterians (Peterson & Eernisse 2001) and a living fossil that allows for looking back in evolutionary history. Hydrozoans were one of the earliest diverging metazoan groups and document early axis patterning before the formation of trunks and the invention of bilateral symmetry (Meinhardt 2012).

Genomes of *Nematostella vectensis* (Putnam et al. 2007) and *Hydra magnipapillata* (Chapman et al. 2010) were the first of phylum Cnidaria to be published. The remarkable plasticity of cnidarian morphology and life cycles is reflected in the genome properties dramatically differing between these two lineages. The genome of *Hydra magnipapillata* is about twice as large and contains large numbers of repetitive elements documenting massive expansions in evolutionary history. In contrast, Nematostella has less repeats but higher GC content than Hydra (Steele et al. 2011).

Phylogeny within genus Hydra remains work in progress (Hemmrich et al. 2007; Chapman et al. 2010; Schwentner & Bosch 2015; Martinez et al. in preparation). Concerning nomenclature it has to be noted that *Hydra magnipapillata* (Ito 1947) and *Hydra vulgaris* (Pallas 1766) had already previously been assigned to the same clade (Kawaida et al. 2010; Martínez et al. 2010) and were then finally considered to be the same species (Schwentner & Bosch 2015). The name is only curated as a synonym anymore in the NCBI taxonomy database (Federhen 2012, 2015) and World Hydrozoa Database (Schuchert 2018). Hence, I will refer to only *H. vulgaris* throughout the rest of this work.

The aim of this work is to gain insights into the genome of Hydra and its evolutionary history by means of genome informatics. Here, I will lay the focus on the examination of repetitive elements and their expansions within the genome. For this work, I chose Hydra for a number of reasons. First, Hydra has a comparatively small genome, smaller than the human one. Moreover, Hydra has significantly high proportions of repeats. Over 50% of its genome was detected to consist of repetitive DNA. These two facts alone would already suggest choosing Hydra for the aim of this work. Nevertheless, the most important fact was that many tools, which I intended to use, had originally been developed for the Human Genome Project or had been bootstrapped against other vertebrate genomes. Accordingly, it is posing a challenge to apply them to a new and significantly different model organism to test their applicability and to possibly gain new insights into the organism's genome as well.

I will first evaluate various tools and compare their different approaches for the detection and classification of repeats. The research question here is, if RepeatModeler is still to be considered the first choice, or if modern machine learning approaches yield better results. Furthermore, I compare repeat detection from raw reads and compare it to the identification of repetitive elements from quality assemblies. Next, I will explore the capabilities of a genome browser to annotate and visualize genomic data. For this purpose, I will demonstrate how interfering software dependencies of such complex tools may be met by using a lightweight approach to virtualization. Then, I will perform the actual investigation on distribution patterns of repeats in particular within regions around essential genes and interpret results in respect of selective pressure. Lastly, I will critically review the common approach to date repeat expansions in the Hydra genome as performed by Chapman et al. (2010). I will propose an alternative method and then compare both approaches in terms of accuracy and their explanatory power.

Repeat identification

Accurate detection and classification of repetitive elements is a non-trivial problem of genome informatics. Repeats, in particular TEs, comprise a great diversity, both within and among genomes. There are many types of repeats. They differ across attributes, such as structure, size, copy number and distribution between and along chromosomes. Over time, many types of repeats diverge by accumulating mutations and thus become even harder to detect. As most repeats are species specific, their silencing mechanisms as well as its impact on predominantly TE structure acts as another obstacle in detection and classification. Of non-interspersed repetitive elements, low-complexity sequences and simple repeats are major sources of false positives. Summing up these issues, accurate automated annotation of repetitive elements is a considerable challenge. (Hoen et al. 2015)

Nevertheless, various methods to identify repetitive elements have been developed. They may be classified into six groups (Lerat 2010; Girgis 2015).

Library-based methods search sequence data for matches in a collection of repeats. This collection may have been manually annotated or automatically created by some other tool. RepeatMasker (Smit et al. 1996) is a member to this category of methods and will later undergo further examination.

Learning-based methods apply Machine Learning algorithms to detect differences between repetitive and non-repetitive sequence features. For instance, such a method may focus on TEs and use the fact that their genes are different from regular genes of organisms in terms of nucleotide composition (Andrieu et al. 2004).

Signature-based methods explicitly search for unique features comprising a class of repeats, such as poly-A tails or long terminal repeats (Ellinghaus et al. 2008).

Comparative-genomics-based methods make use of the fact that repeats are species specific. A comparison of genomes of closely related species unveils repeats present only in one of both genomes (Caspi & Pachter 2006).

De-novo methods aim to discover repeats ab initio, i.e. without prior knowledge about their structure, only by scanning genomic data for repetitive sequences. Typically, this is performed

by aligning the genome to itself. This is a very computation intensive process (Z. Bao & Eddy 2002). An alternative here is counting of k-mers (Price et al. 2005).

One major challenge, the identification of repetitive elements is posing to genome informatics, is the fact that repeats are species-specific elements. In newly sequenced genomes, they are likely to be unknown. Tools are required to discover such repeats de-novo, i.e. without prior knowledge.

Another general challenge to genome informatics is that the majority of eukaryotic genomes are unfinished. These genomes have not yet been assembled completely. Most often, they consist of unordered sets of contigs and scaffolds of various size and without knowledge about their attribution to and their position on chromosomes. This fragmentariness is due to the algorithmic challenges the assembly of a genome is posing. One possible approach is to use different tools and parameters to produce multiple assemblies and then to reconsolidate these into a consensus assembly (Alhakami et al. 2017).

The aim of this chapter is to evaluate tools for the de-novo identification of repeats under such challenging conditions. I will compare the repeats discovered from an assembled and an unassembled genome, and I will check if these tools are sensitive to both transposable elements and non-interspersed repeats alike. A more comprehensive review of tools available for the identification and classification of repeats and in particular transposable elements is provided by (Saha et al. 2008a) and (Goerner-Potvin & Bourque 2018).

RepeatMasker

RepeatMasker (Smit et al. 1996) is the most widely used tool for masking and annotation of repetitive elements in DNA sequence data. It scans sequences for interspersed repeats and low complexity elements provided in a library. The program then produces a modified version of the query sequence with all identified repetitive elements masked i.e. replaced by an arbitrary character not denoting for any nucleotide. In addition, detailed annotation data for every such repeat is provided in a separate file.

For finding repetitive sequences, RepeatMasker loops through all query sequences to align them with each consensus sequence in the repeat library. The default alignment engine for this purpose used to be Cross_match (Ewing et al. 1998; Ewing & Green 1998), a highly sensitive but at the same time rather slow implementation of the Smith-Waterman alignment

algorithm (Smith & Waterman 1981). To speed up repeat identification, alternative alignment engines have been made available, such as AB-BLAST (formerly known as WU-BLAST) (Lopez et al. 2003). This engine uses a faster heuristic approach at the cost of identification sensitivity. In the current versions of RepeatMasker, the default search engine now is NCBI/RMBLAST, a RepeatMasker compatible version of the standard NCBI BLASTn program.

Even though alternative tools have been published, RepeatMasker is still considered the gold standard for masking and annotating repetitive elements. Nevertheless, RepeatMasker can only find those repeats that are to a certain degree similar to already known repeats contained in the library it uses.

Repeat libraries for RepeatMasker can be obtained from various curated collections of prototypic DNA repetitive element data. Most prominent ones here are RepBase (Jurka 1998; W. Bao et al. 2015) and Dfam (Hubley et al. 2016). RepBase provides manually annotated consensus sequence data of repeats from different eukaryotic species. Over time, the collection became too divergent for some use cases. As a consequence, Dfam was created by using a machine learning approach on alignments for repeats taken from RepBase for a limited number of eukaryotic model species only. Species include human, mouse, zebrafish, nematode and fly. Nevertheless, both databases are considered as valuable to be used as a first step in repeat identification. Therefore, various tools other than RepeatMasker make use of these databases as their default source for repeat identification and evaluation. However, RepBase and Dfam are disputed by parts of the scientific community, among other things because of their insufficient description of the model generation process applied (Arensburger et al. 2016).

Besides from these two rather generic sources, other genome projects provide data on repetitive elements for several other species. Thereby, they either focus on single species only, such as WormBase (Stein et al. 2001; Lee et al. 2018) which claims to be a central resource for *Caenorhabditis elegans* data. Alternatively, these projects may offer data for major eukaryotic groups such as PGSB (Spannagl et al. 2017) with its RPSB-REdat focusing on various plant genomes. Typically, these alternative sources offer data in formats not suitable for direct use in RepeatMasker and thus need processing prior to import.

In addition to using such pre-built external repeat libraries, users may construct their own libraries from scratch. As repeats are species-specific, newly sequenced genomes require tools for a de-novo discovery.

RepeatModeler

RepeatModeler (Smit & Hubley 2008) is one such tool for de-novo repeat family identification and modelling. It is provided in bundle with RepeatMasker. RepeatModeler employs two complementary methods of computing repeat element boundaries and relationships between repeat families. Given a collection of genomic sequence data, RepeatModeler follows a consensus approach by embedding two different but complementary methods of repeat detection into one workflow to construct and to refine a consensus model of putative repetitive sequences.

One of these component programs is RECON (Z. Bao & Eddy 2002). It applies the self-comparison strategy of subsampling the genome and aligning samples to the whole genome. After grouping repeat candidates into families, one consensus sequence for each family is calculated by aligning the ten longest members of the family using DIALIGN2 (Morgenstern 1999) and then applying a simple majority rule to select consensus residue.

The other component program is RepeatScout (Price et al. 2005). It first identifies k-mers occurring at high frequency in the genome and then extends these seed sequences by following a greedy strategy. To guarantee convergence into a consensus sequence representing repeat families, a stepwise extension of the candidates is made, while iteratively realigning them to their actual occurrences in the genome and discarding outliers.

To calculate proper de novo repeat models, RepeatModeler requires a high quality genome assembly with scaffolds containing unfragmented sequences of the most repeated portions. Typically, such assemblies were obtained from classic Sanger sequencing, which provided longer reads with medium coverage.

dnaPipeTE

The software dnaPipeTE (Goubert et al. 2015) was inspired by ideas from transcriptomics and is implemented as an automated pipeline script that combines various other tools into a common workflow. The tool does not need an assembled genome for identification of repetitive elements, but it is able to use raw short reads instead, as obtained from NGS. The basic assumption is that repeated DNA sequences are over-represented in reads, i.e. have a higher coverage than sequences that occur only once in the genome.

In short, reads are first grouped into samples that represent far less that 1x coverage of the genome. This is achieved by using a sample size of about one quarter of the genome size only. The idea here is that if a read makes it into such a sample, chances are high that it must be from a repetitive sequence. Then, contigs are built from putative repetitive reads using Trinity (Grabherr et al. 2011). In transcriptomics, particularly in RNA-seq studies, this software is used to produce contigs that represent all alternative transcripts of a putative gene. In dnaPipeTE, this feature of Trinity is used for TE families instead of genes and actual TE copies instead of RNA transcripts. In particular, Trinity builds alternative contigs for each such TE family representing a whole set of TEs with sequence divergence by base mutations, insertions, deletions and other changes of structure. As a result, these contigs are consensus sequences of one or several closely related repeat families.

This procedure is iterated trough several times (parameter `-sample_number`). All reads mapping to a putative repeat contig are added to the next sample. This is then stocked up with reads not yet used in other samples. Then, Trinity is run on this new sample again and so forth. This procedure guarantees an accumulation of reads from putative repetitive sequences and increasingly reduces the proportion of non-repetitive elements.

Finally, the dnaPipeTE pipeline runs RepeatMasker to annotate contigs and performs a BLAST search of an independently captured sample against the contig database to estimate repeat quantities.

Prior to dnaPipeTE, tools such as ReAS (R. Li et al. 2005) took a similar approach for Sanger reads. These days most projects are based on NGS data, consequently such tools tend to pass into oblivion.

Red

Red (Girgis 2015) uses a Machine Learning approach for repeat discovery. Initially, Red applies three techniques to identify candidate regions that are likely to contain repetitive elements. First, Red searches the genome for k-mers occurring at least three times. In addition, they must show a higher abundance than expected by chance. To calculate this expectation value, a Markov chain is trained on the genome. Actual k-mer counts are then adjusted to correct biases (Achaz et al. 2007) and stored in a look-up table. Next, candidate sequences are scored by individually scoring their nucleotides with the count of k-mers starting at that nucleotide. In other words, this is a measure for how frequently this nucleotide is the starting

point of a k-mer at other loci in the genome. Scoring sequences are then smoothed by applying a method inspired from signal processing. Scores are averaged with the ones of their neighbours by use of a Gaussian mask. Red is performing this step to remove short outliers and, as a result, to obtain longer adjacent sequences of either high scores (repetitive sequences) or low scores (non-repetitive sequences). Finally, an approximation of second derivative test is calculated on the scoring function of candidate sequences. The presence of a local maximum is rated as an indicator for repetitiveness. Consequently, only such sequences are then kept.

Red then uses the remaining candidate sequences for its machine-supervised training of the Hidden Markov Model. As candidate regions are clearly distinguished from non-candidate regions, the program can train itself without human contribution. Finally, Red feeds the whole genome sequence into the then well-trained model. This procedure is expected to detect repeats at a higher sensitivity than each on the initially applied methods could provide.

Hidden Markov Models (HMMs) have been used for analysis of sequence data for more than two decades already (Durbin et al. 1998). They are stochastic models of a succession of possible events, where the probability of occurrence of each such event depends on the state caused by the previous event only. Furthermore, the assumption is that states are not directly observable, but that only their outputs are visible.

Red assumes a genome to be such time series data. It interprets repetitive elements as events along the genomic sequence, while the repetitiveness of a certain region or read is the only observable output. The hidden state here is the assignment of the region in focus to individual repeats. Initially, the HMM has no prior knowledge about the succession of events i.e. repeats. It consists of a collection of states and probabilities of some defined initial values only. While then consuming the candidate sequences, the HMM is at the same time trained with the information about the occurrence of repeats that had been previously computed for these candidate sequences. This machine-supervised learning process makes the HMM to adapt its states and probabilities to approach the observable chain of output. After completion of the training phase, the Hidden Markov Model consists of states and probability values representing the learned time series of repeat occurrence. It is then ready to apply this knowledge to the genome sequence and to detect repetitive elements in the remaining portions of the genome.

This approach is fundamentally different from previously described tools. It promises to detect repeats that will not be found otherwise. Apart from that, this method implemented in Red has

some shortcomings already described by the author. As HMMs require strictly linear time series data, repeats have to be assumed as serial events along the genomic sequence. This makes it impossible to detect nested or overlapping repetitive elements and gives high false negative rates with degenerated repeat sequences.

Furthermore, Red was not designed to classify repeats. Instead, it states to process assembled and unassembled genomes equally well, a claim I will examine further in the upcoming section.

Comparison of de-novo discovery tools

There are numerous criteria to evaluate tools for de-novo repeat identification, among others the algorithmic aspects (Bergman & Quesneville 2007; Modolo & Lerat 2014) and metrics, such as sensitivity and specificity (Saha et al. 2008b).

For the aim of this work, I focus solely on comparing the results of the aforementioned three software packages from discovering repeats in the Hydra genome. I chose this genome because of its small size (1.3 Gb) and its known high amount of repetitive elements (Chapman et al. 2010).

Materials and methods

I used two datasets, an assembled genome and a not assembled one. The first one was the genome of *Hydra vulgaris* strain 105. It was originally obtained by Sanger sequencing and published by Chapman et al. (2010). The assembly had then been further improved to obtain longer scaffolds by applying a method based on chromatin reconstitution (Putnam et al. 2016). The genome size was estimated to be 1.3 Gb, the assembly size is 853.8 Mb with the longest scaffold of 4.4 Mb, an N50 length of 1.0 Mb and an N90 length of 161.8 kb. I will reference to this dataset as the “Hm105 genome” from now on and use it as input to RepeatModeler. This de-novo detection tool best operates on well-assembled genomes.

The second genome used represents a different strain of *Hydra vulgaris* (Zurich) and originated from a project yet to be published (Martinez et al. in preparation). The sequencing technology used for this genome was Illumina Next Generation Sequencing. The dataset consisted of raw Illumina reads. Furthermore, there was a mapping of these reads to the Hm105 reference genome. I used this mapping later in genome browser to compare it against the reference sequence for identification of conserved regions. Moreover, a de-novo assembly of the Zurich genome had been initialized in the past. As Illumina sequencing used two lanes

only, results were of low quality in general. The total length of reads was 607.8 Mb, 83.95% of them mapped to the reference assembly. The read mapping covered 77.25% of the assembly. To achieve a better assembly, in the future, people may be able to use mate pair libraries to stitch individual short contigs together into bigger scaffolds. However, I used this genome as input to dnaPipeTE, as this tool requires short reads at lengths typical for NGS technologies.

For repeat identification and classification, I applied a common workflow. I used the tools in comparison for the identification of repetitive sequences only. Repeat annotation in genomes was always performed by RepeatMasker to obtain comparable results.

As a reference, I obtained repeat annotations from RepBase and Dfam respectively. RepeatMasker (version open-4.0.7) was searching these databases for all consensus sequences published for the taxon Hydra (parameter `--species hydra`) and downloaded them into a local library file.

For the de-novo tools, a library of putative repeats was built with tools RepeatModeler open-1.0.10 (using system default NCBI BLAST 2.6.0+; default parameter values), dnaPipeTE v.1.3.1_07.dec.2017 (using its own versions Trinity v2.5.1 and NCBI BLAST 2.2.8; parameter `-sample_size 1000000`) and Red version 05/22/2015 (no dependencies; default parameter values).

Then, I transformed tool output to a file containing repetitive sequences only. This step was only necessary for Red. It returned the name of the scaffold as well as the start and end positions for each repeat found. RepeatModeler and dnaPipeTE both provided direct output of repeat sequences.

The following step was then to use RepeatMasker to annotate the genome using the library. This way, I obtained classification data on the repeats. This step was only explicitly necessary for dnaPipeTE and Red, skipping the contamination check for bacterial insertion elements (parameter `-no_is`). RepeatModeler runs RepeatMasker as part of its workflow and provides the statistics data in a `.tbl` file at the end of its execution already. However, also dnaPipeTE performed such a step during its workflow to classify repeats in the contigs built by Trinity. To find dnaPipeTE putative repeats in the reference assembly, I had to run RepeatMasker one more time. I performed all RepeatMasker runs with masking repetitive sequences as lowercase letters (parameter `-xsmall`).

As the number of putative repeats identified by Red was exceptionally higher than that of the other tools, I applied post-processing to Red results to obtain an additional smaller dataset from Red. Therefore, I searched for sequences having 100% matches in the genome, and only retained those found at least two times. I will reference to the original dataset obtained directly from Red as “Red full”. To the condensed dataset processed by a script, I will reference as “Red core”. For calculating proportions of the reference genome masked by applying the libraries, I used values provided in RepeatMasker .out file. For the total number of base pairs I included the N characters in the FASTA sequence denoting unknown bases. To verify these numbers I furthermore developed a script to count the bases masked as lowercase letters.

For comparison of repeat coverage, I mutually applied BLAST (Nucleotide-Nucleotide BLAST 2.6.0+) to all pairs of result datasets, i.e. for each pair I took each putative repeat library once as query and then as database to run the BLAST search against it. The BLAST parameters were set to find only one target sequence of at minimum 90% identity (parameters `-max_target_seqs 1 -perc_identity 90`), DUST was turned off (parameter `-dust no`) for not losing low complexity repetitive elements. Duplicates were filtered out and the number of result sequences was counted for each such pairwise comparison.

I calculated genome proportions by running RepeatMasker on the reference genome using repeat libraries obtained from RepBase/Dfam, RepeatModeler, dnaPipeTE and Red. Consequently, these proportions are well comparable among each other as they are based upon repetitive sequences actually found in the reference genome and their further classification.

Results

The numbers of repeat families identified, i.e. sequences in the library, vary greatly in size (Figure 2A, left column). RepBase detected only 873 repeat consensus sequences. RepeatModeler found 3028 different repeat sequences within Hydra genome, whereas dnaPipeTE classified 11300, nearly the fourfold. Red result numbers were surprising. A vast number of almost 800000 repetitive sequences was identified. These appear to be actual sequences and not necessarily consensus sequences. I condensed this large number to only 5442 sequences by a script as described previously.

When comparing proportions of the genome masked by RepeatMasker using these repeat libraries, I found RepeatModeler to mask the largest share of 55.6% followed by

RepBase/Dfam at 44.1%, dnaPipeTE at 40.6% and Red full at only 37.3%. Red core libraries provided the worst result at only 15.0% of the genome masked.

A

	number of sequences in library	absolute number of bases masked	percentage of genome masked
RepBase/Dfam	873	376318733	44,1%
RepeatModeler	3028	474433817	55,6%
dnaPipeTE	11300	346616056	40,6%
Red full	799857	318420188	37,3%
Red core	5442	128404311	15,0%

B

BLAST database \ BLAST query	RepBase/Dfam	RepeatModeler	dnaPipeTE	Red full	Red core
RepBase/Dfam		97,1%	99,0%	99,1%	98,9%
RepeatModeler	81,8%		97,8%	79,8%	50,8%
dnaPipeTE	86,4%	89,9%		91,3%	95,1%
Red full	92,4%	85,8%	94,8%		99,4%
Red core	87,2%	98,8%	99,3%	97,7%	

Figure 2. **Comparison of repeat libraries and genome masking capabilities of the tools.** (A) Numbers of sequences in each library and masking results of various tools: numbers of bases masked in genome, percentages of genome masked; (B) comparison of library sequences against each other by using BLAST at 90% sequence identity level

On comparison of sequences of de-novo tools against the human curated databases RepBase and Dfam, results show surprisingly high coverage in database consensus sequences (Figure 2B). Only for RepeatModeler, about 3% of library sequences are not covered. Comparing the library sequence coverage between the de-novo tools, results show high similarity. Almost all of dnaPipeTE sequences were similar to such found by RepeatModeler as well (97.8%), while vice versa dnaPipeTE found 89.9% only of RepeatModeler results, i.e. about 10% of library sequences were not covered by dnaPipeTE.

Differences in library similarity between variants of Red were small. Red core results were present almost completely in the by far larger output of Red full (99.4%) whereas during reduction in size 97.7% of sequences remained, i.e. reduction by factor 150 lost 2% of library sequence coverage only. Comparison of classical approaches RepeatModeler and dnaPipeTE with machine learning approach Red (full output) show 85.8% and 94.8% similarity, i.e. Red omits 14% of library content identified by RepeatModeler and 5% of such by dnaPipeTE. Reciprocally, only 79.8% of Red sequences were present in RepeatModeler output and only

91.3% in dnaPipeTE output, i.e. Red claims to identify 20% of library sequences not found by RepeatModeler and 8% unavailable in dnaPipeTE output. However, 99.1% of the library sequences identified by Red were already covered by RepBase/Dfam consensus sequences, i.e. the database content appears to be of very good quality for Hydra. Machine learning did not come up with significantly new repetitive elements.

As 98.8% of RepeatModeler and 99.3% of dnaPipeTE putative repeats were contained in the condensed result set obtained from Red output postprocessing, the discovery rate of Red core is surprisingly high. This means de facto all of RepeatModeler and dnaPipeTE sequences are represented within the Red core set. Vice versa, similarity deviated greatly, as 95.1% of Red putative repeats were also identified by dnaPipeTE but only 50.8% by RepeatModeler. Results suggest that Red core repeats are well represented by dnaPipeTE whereas RepeatModeler does not find half of them.

Unexpectedly, matches of RepBase/Dfam consensus sequences against library sequences of the de-novo tools show low percentages. Only 81.8% of RepBase/Dfam sequences match at a 90% identity level with RepeatModeler sequences, for dnaPipeTE its 86.4% and for Red even 92.4%. Not to forget here is that numbers in Figure 2B presumably contain a contradiction or inconsistency. Comparing RepeatModeler and dnaPipeTE to Red full and Red core respectively, it remains unclear, how the true subset Red core can contain higher portions of the query sequences (98.8% and 99.3%) than the full set (85.8% and 94.8%).

Annotation results of putative repeats by RepeatMasker show very different proportions of repeat classes for RepBase/Dfam (Fig. 3A) and RepeatModeler (Fig. 3B). RepeatModeler identified more DNA transposons and a threefold higher proportion of unknown transposable elements than RepBase/Dfam. For dnaPipeTE, I included proportions identified in the reads (Fig. 3C) as well as in the assembly (Fig. 3D). Overall, dnaPipeTE in reads identified a more than 15% higher repetitive proportion of the genome than in the assembly (68.9% vs. 53.8%). The percentage of non-interspersed repeats discovered by dnaPipeTE was significantly lower in the reads, only 0.6% of low complexity and simple repeats, compared to 5.5% identified in the assembly.

Proportions of repeat classes appear to be similar between RepeatModeler and dnaPipeTE when applied to the assembly, only the share of unidentified TEs is fourfold higher in RepeatModeler. Results for genomic proportions of TEs and simple repeats appear similar between RepBase/Dfam and RepeatModeler. Nevertheless, RepeatModeler identified a

threefold higher percentage of unclassified mobile elements, reducing the share of non-repetitive content to 43.6% whereas for curated RepBase/Dfam is was 53.8%.

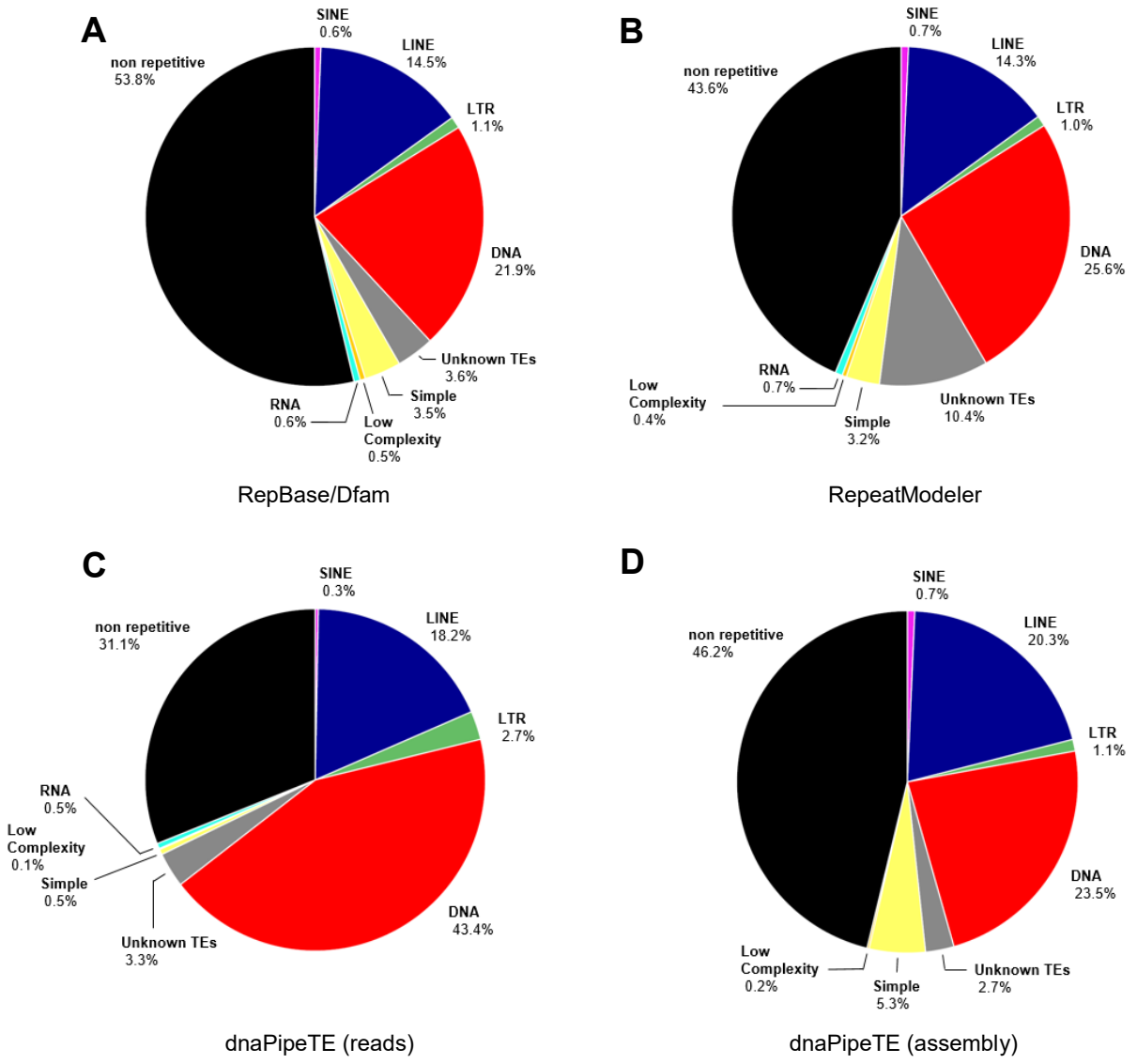


Figure 3. **Genomic portions of repeat classes.** The reference genome was masked with RepeatMasker using libraries from (A) RepBase/Dfam, (B) RepeatModeler, (C) dnaPipeTE on reads and (D) on assembly; percentages calculated from numbers of base pairs masked for each class

RepeatMasker was not able to classify putative repeats identified by Red in detail. For Red full and core, it claims 4.2% and 4.8% respectively of the genome size to consist of simple repeats. Low complexity and RNA categories show no identified sequences. For interspersed repeats the putative proportions vary, 34.4% for Red full and 11.0% for Red core. Nevertheless, RepeatMasker did not classify transposable elements further.

Discussion

The variation in numbers of library sequences identified by the tools (Figure 2A) supposedly refers to the different kind of sequences obtained. RepeatModeler and dnaPipeTE output a comparatively low number of consensus sequences. They represent the whole spectrum of repeats detected in the genome in a compact way. Red, by contrast, provides all putative repetitive sequences detected and does not compact them further. As a result, the total number of obtained library sequences is far higher with Red than with the other tools. My naïve compactification method demonstrates that the full result set from Red can be reduced to the order of magnitude observed with the other tools. The more compact result set Red core is still representative (97.7% matches at 90% sequence identity level) in terms of library similarity.

However, applying these library sequences to the genome assembly for masking actual repeats gives a completely different picture. My initial expectations were fully met here. In RepBase all sequences are manually curated. This comes at the cost of possibly missing some repeats. As a consequence, I expected more false-positives in RepeatModeler, because it possibly detects sequences as repetitive that actually are not really repeats. Results confirmed this assumption as RepeatModeler library masks 55.6% of the genome assembly as repetitive, whereas the library obtained from RepBase/Dfam masks only 44.1%, i.e. RepeatModeler masks 25% more repeats than RepBase. This result is also consistent with library sequence coverage as only 81.8% of RepBase/Dfam library sequences are similar to sequences of RepeatModeler. The conclusion here is that RepeatModeler finds a substantial amount of new repetitive elements, furthermore confirmed by the lower proportion of non-repetitive elements (Fig. 3B).

A comparison of repeats identified by RepeatModeler and dnaPipeTE respectively confirmed my initial assumptions. RepeatModeler shows almost complete (97.8%) coverage of library sequences discovered by dnaPipeTE, but vice versa dnaPipeTE was not able to discover all such library sequences (only 89.9%) of RepeatModeler. Therefore, RepeatModeler sequences mask 55.6% of the genome whereas dnaPipeTE library sequences applied to the assembly only mask 40.6% thereof. Obviously, the proportions of non-repetitive elements in Figure 3 do not perfectly match, as I would expect them to differ in more than only 2.6%. Nevertheless, this suggests that RepeatModeler is better at de-novo detection, but results depend on the quality and correctness of the assembly used. However, considering that almost 23% of the assembly were not covered by reads, results suggest that dnaPipeTE performed very well in detecting repeats from the reads available. Obviously, this tool has the potential to perform equally well or maybe better than RepeatModeler.

Results for Red are similar to somehow surprising and contradictory. RepeatModeler and dnaPipeTE library sequences are almost completely present in Red full and core repeat sets (85.8% and 98.8% vs. 94.8% and 99.3% similarity) (Fig. 2B). Vice versa, Red full and core library sequences are well represented by dnaPipeTE (91.3% and 95.1%) whereas RepeatModeler did not detect a large proportion of them (79.8% and 50.8%). The conclusion here is that Red and dnaPipeTE discovered similar library sequences, whereas Red detected many new not found by RepeatModeler. Again, actually masked repeats show a different picture. Here I see RepeatModeler detecting about 50% more repeats in the assembly than Red full. Red core lags far behind with only 15.0% masking, clearly a result of removing so many sequences from the Red full library.

Unexpected results of matching RepBase/Dfam consensus sequences against putative repeat sequences of the de-novo tools are most surprising. Only 81.8% of RepBase/Dfam sequences match with RepeatModeler sequences, while for Red 92.4% do. This is about the opposite of what was expected. The machine learning approach is doing far better here than RepeatModeler. Nevertheless, for now I have no detailed explanation for this.

I previously identified another contradiction in Red results. It was unclear how Red core as a true subset of Red full can contain higher portions of RepeatModeler and dnaPipeTE library sequences (98.8% and 99.3%) than the full set (85.8% and 94.8%). Now, I understand that the massive amount of sequences removed from the library contained a disproportional low amount of such contributing to finding actual repeats in the library. There appears to be considerable redundancy within Red full library sequences, as was obvious from the large count already. Nevertheless, Red core only identified 15% of the genome as repetitive. Therefore, it appears to be of minor use for repeat detection.

To draw a conclusion, results suggest that there is only a vague correlation between the similarity of library sequences calculated and their potential to mask actual repeats in the genome. Mutual coverage of libraries appear to be high (Fig. 2B) while percentages of repeats identified vary widely (Fig. 2A).

Concerning repeat proportions displayed in Figure 3, we have to recall to our minds that the quantification of repeats is an estimation only, not a hard and accurate measurement. Results are affected by non-quantitative sequencing technologies and accompanying steps, such as PCR, and their various biases. Furthermore, results may be influenced by the properties of the tools used for building de-novo repeat libraries and affected by shortcomings in RepeatMasker

and inaccuracies in BLAST. Nevertheless, it appears to be reasonable to discuss differences between repeat proportions of the genome suggested by each tool.

Similarities between RepBase/Dfam and RepeatModeler in terms of identified transposable elements and simple repeats were as expected. Classification of such elements is based on a common body of knowledge about previously identified repetitive elements, their structure and sequences. Nevertheless, RepeatModeler identified three times more TEs that it was not able to classify further. This result suggests that either about 6.8% of the Hydra genome consists of repeats that were not yet covered by RepBase/Dfam consensus sequences or that there are numerous false positives detected by RepeatModeler.

The lower percentage of non-repetitive DNA in dnaPipeTE on raw reads is caused by the method itself, as the iterative sampling is explicitly avoiding non-repeated genome content by filtering out rare reads, i.e. such not originating from repetitive sequences. Furthermore, already during sequencing it is more likely to obtain reads from repetitive portions of the genome, than from rare non-repeated parts. The iterative procedure dnaPipeTE is applying guarantees an accumulation of reads from putative repetitive sequences. Consequently, it increasingly reduces the proportion of non-repetitive sequences as well.

The approach dnaPipeTE took seems to be expedient, when no genome assembly but only unassembled reads are available. However, such assemblies do not per se guarantee their correctness. Errors in such complex genome assemblies may remain undetected for years, affecting the quality of results derived from them. Accordingly, some people argue that using unassembled reads for the identification of repeats may be a more accurate approach (Saha et al. 2008b). On the contrary, the analysis of raw sequence reads does not clarify spatial relationships between the elements identified. This may also influence the discovery rate of more complex transposons. Accordingly, I suppose that the increase in dnaPipeTE detection of DNA transposons compared to RNA transposons is owed to this bias.

Furthermore, the presence of flanking terminal inverted repeats in subclass 1 of DNA transposons as well as flanking long tandem repeats in LTR RNA transposons are features well detectable by dnaPipeTE. This fact might explain their relative increase when using dnaPipeTE. Figure 3C shows a substantial decrease in the detection of simple repeats while LTR RNA transposons almost tripled. Supposedly, while dnaPipeTE is iteratively adding unused reads to pre-built contigs, such new reads from tandem repeat sequences are more likely to be aligned to the flanks of an LTR contig than used for the extension of pure tandem

repeat sequences. Minimum length requirements of LTR flanking sequences could be weaker than for the classification as a simple repeat. As a consequence, they could favour the attribution as LTR transposon and impede simple repeat detection. Similar bias may account for the reduced detection of low complexity repeats. Furthermore, the decrease in SINE detection by dnaPipeTE could be ascribed to this behaviour. The 3´ sequences of SINEs are often identical to some LINE sequences. If the previously supposed bias exists, it would also account for the misalignment of SINE reads to LINE contigs and subsequently decrease their detection ratio.

Similarly, the share of unclassified TEs is by far smaller with dnaPipeTE on reads. Again, I attribute this to possible misalignment of reads from unusual and infrequent TEs to other already identified transposon families. Candidates for such failures could presumably be subclass 2 DNA transposons belonging to or similar to the superfamilies Helitron and Maverick that both contain a variable number of additional open reading frames. As relative position information is unavailable from unassembled reads, reads from such long spanning structures may not be identified as belonging to such an RNA transposon at all. Instead, they will be aligned with other putative repeat contigs or even be sorted out completely as non-repetitive content. Assemblers and Trinity alike supposedly tend to attribute reads to already formed contig sequences assuming high coverage, than to start from the premise that they need to be stringed together to long repeating sequences.

Proportions identified as RNA are questionable, because tools proceed on different assumptions. All tools identify transfer RNA (tRNA) sequences. Nevertheless, RepeatMasker on the one side annotates them as a subclass of SINE elements, while listing them as their own category in its stats file. However, RepBase and Dfam contain numerous manually curated ribosomal RNA (rRNA) consensus sequences that are completely missing in all de-novo repeat discovery tools. In Figure 10 I provide an example section with annotation of both RNA types. I will discuss it in detail later. These rRNA annotations are also included in the RepeatMasker stats file. Nevertheless, the total proportions for RNA when using the RepBase/Dfam library is even lower than when using the RepeatModeler library. This appears to be a contradiction. Results for RNA proportions as obtained from the tools were included in figures. Accordingly, they are responsible for a minor shift in other proportions.

However, to improve identification of RNA sequences and to differentiate from repeat sequences, I suggest to include convenient tools into the workflow, such as RNAmmer (Lagesen et al. 2007) for rRNA genes and tRNAscan-SE (Lowe & Eddy 1997) for tRNAs. For

other non-coding RNAs, a sequence homology search of the Rfam database (Griffiths-Jones et al. 2003) could be useful to annotate them correctly.

As a final note, I expect the proportions of simple repeats in this assembly to be significantly lower than in the actual genome itself. At the current state, centromeric and telomeric regions are not contained in this assembly. It will be necessary to fill the respective gaps, to improve the assembly quality and to further combine scaffolds into chromosomes. Presumably, this will only become possible by obtaining substantially longer reads from new sequencing technologies, such as PacBio (Rhoads & Au 2015).

Genome annotation and repeat visualization

Today's high-throughput sequencing technologies generate immense volumes of genomic annotation data to process and look into. Tools for visualizing and annotating these data have become invaluable, most often called "genome browsers".

The principal function of such browsers is the aggregation of various annotation data and their integration into an abstract graphical view. Uniform coordinates allow identifying locations within the genome, typically consisting of an identifier for a chromosome or scaffold as well as start and end positions. Users may navigate the genome by choosing parts of variable size to be depicted. They may move this view window back and forth along the x-axis and zoom in and out to determine the level of detail displayed.

Figure 4 shows a screenshot of such a genome browser. Different annotations are organized as tracks, providing one type of information per track. These tracks may then be stacked along the y-axis of the browser display, their visibility, order and type of visualization can be changed. The basic track typically displays the genome assembly itself and optionally shows coordinates. Other tracks may then show annotations, such as gaps, GC content density, genes identified or mappings of transcriptome reads or repetitive elements.

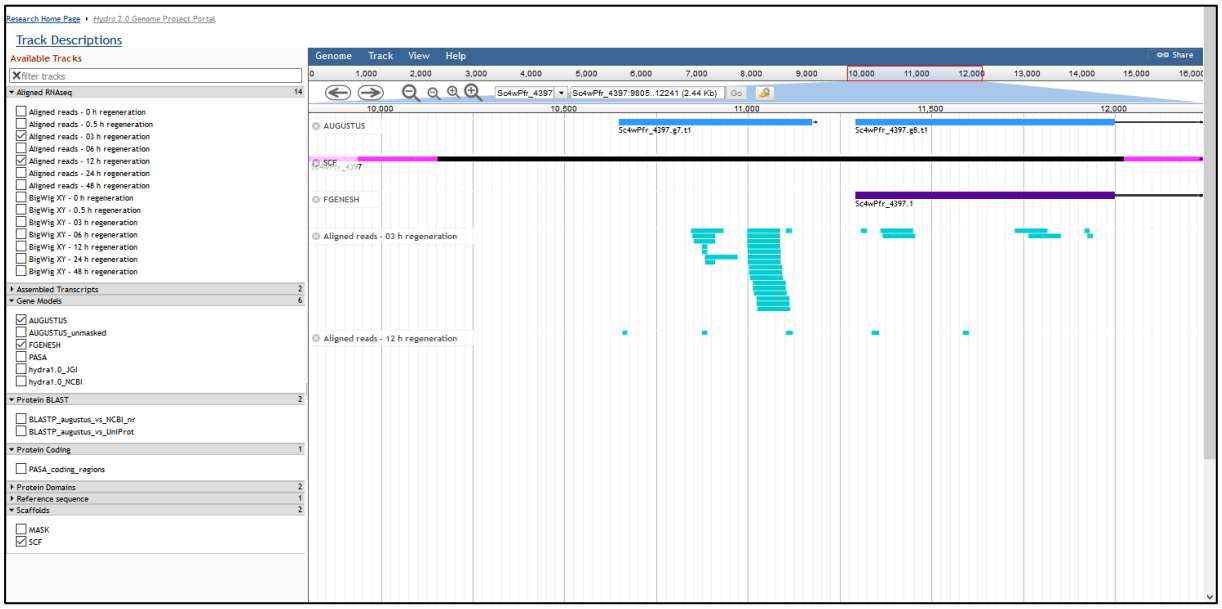


Figure 4. **Genome browser at NHGRI portal of Hydra 2.0 Genome Project**. Screenshot of https://research.nhgri.nih.gov/hydra/jbrowse/jbrowse.cgi at 2018-10-03

This piling up of multiple tracks at the same genome coordinates allows users to examine the consistency or difference in annotations of this region. Of special interest are sequence alignments of other more or less closely related species. Such alignments will allow comparative analysis and attribution of genomic features in relation to the reference genome.

Genome browsers are typically deployed as web applications, giving instant access to vast amounts of data precomputed and stored on the server. Only the parts necessary for display on the client computer need to be transmitted over the network. This approach avoids waste of computation time and makes working less resource intensive on the client side.

There are numerous genome browsers available today and just as numerous are criteria by which to compare such implementations (Wang et al. 2013). There is ongoing development of new approaches and new tools are constantly replacing older ones (NCBI 2017). I will now go into more detail on the most widely used genome browser. It is now around for almost two decades but still it may be seen as up to date.

The UCSC Genome Browser

First released in 2001, the UCSC Genome Browser (Kent et al. 2002) had been developed in the end phase of the Human Genome Project as a tool to display the newly assembled genome. It is a multiple-species genome browser providing visual access to over 180 assemblies of more than 100 species. Due to its flexible extension mechanisms, users worldwide may contribute their own data tracks and make them in turn available to others. The browser is under active development and maintained by a team from the University of California Santa Cruz and a spinoff company thereof. Updates of data and tools are provided at regular schedule and subsequently are published once a year, such as in the 2018 update paper (Casper et al. 2018). In addition, the university is running an official public browser instance with several mirrors worldwide. However, as the browser is open source software under a permissive proprietary license and free for non-commercial use, users may also run their own on-premises installations.

To deploy UCSC Genome Browser on the provided infrastructure, I used Docker container virtualization to provide the required environment for each component while keeping all together on one system and perform the rollout in a reproducible way (Boettiger 2015). Docker uses certain programs that are already part of the Linux operating system. It isolates application components from each other and provides separated views of the underlying file

system to them. This variant of virtualization generates only very small overhead and shows good general performance (Joy 2015). Moreover, it has a low impact on the performance of bioinformatics pipelines (Tommaso et al. 2015). Furthermore, isolation of components makes it possible to define dedicated interfaces between components themselves and to the outside environment and thereby to hide internals. Dependencies on other software are resolved by installing such programs within the same container. By this means, I make them invisible to the outside environment. As a result, conflicting dependencies may not interfere any more. Even different versions of the same software may coexist. An intended side effect of this approach is that installations of such a container-based infrastructure are automatically well documented by writing the according configuration files.

A major challenge in installing UCSC Genome Browser was to provide the required environment with minimal modification of the software itself. The code of the browser is largely written in C programming language and appears complex and error-prone to customize. Furthermore, I had to provide various different genomes on the same server, some of which were intended to be publically available while access to the remaining had to be restricted and limited to certain users only. Figure 5 depicts these two instances of the genome browser, each consisting of three components. Communication is limited to a private subnet and sharing a folder to access data. The browser itself is a web application that reads genome and annotation data from the shared folder and a database to provide alignment search via a BLAT server. These three components interact within their isolated environment. Only the web application offers a local interface to other programs outside of this container network.

To provide access to the public and private browser instances under the same main URL, a frontend webserver was installed – again in its own container. It receives requests to the public or private browser website and redirects them to the according instance while also handling authentication for the latter. Requests are then processed within the target browser network and the response is sent back to the frontend webserver that again forwards it back to the original client. Furthermore, the frontend webserver hosts track hubs for flexible integration of annotation data. I consider frontend webserver and browser networks the production environment of the local UCSC Genome Browser installation. They are required to be continuously available online as other teams worldwide have access to the genome data.

Strictly separated from this, I installed the development environment consisting of browser source files downloaded from the UCSC project page and Linux development tools, such as compilers, to produce the required binary executables for the platform. Furthermore, I installed

a dedicated database. A well-defined update procedure was established to guarantee minimal downtimes of the production system. All preparation work, such as downloading raw genome data, converting it to various formats or calculating annotations on it, is performed in the development environment and then loaded into the development database. After successful preparation, the genome dataset is then copied into the according browser instance database and volume folder and the database container is restarted to include the new data tables.

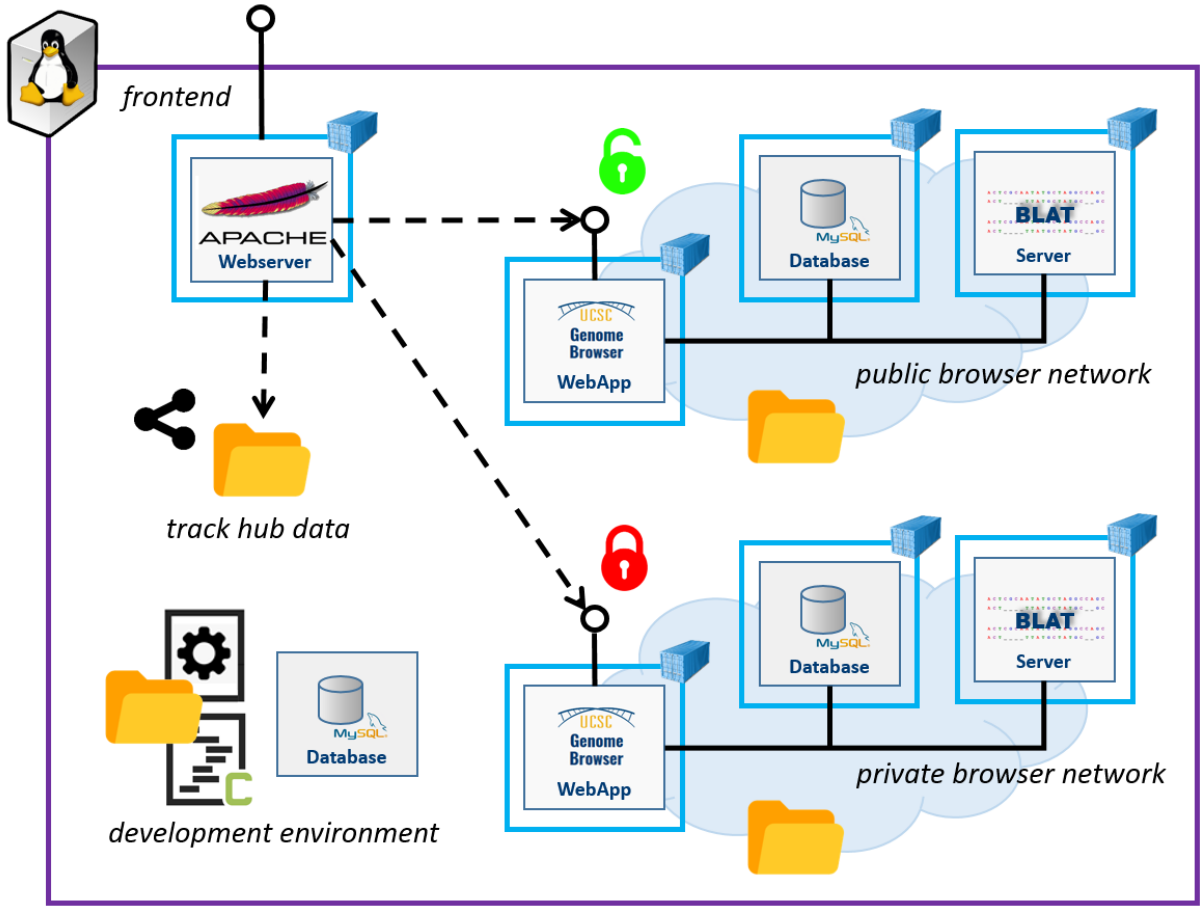


Figure 5. **UCSC Genome Browser installation overview**. Components are isolated in Docker containers and communicate on dedicated interfaces only. A frontend webserver combines various browser instances and track hubs into one website. Downtimes on the production system are minimized by separating it form the development environment.

For now, the development environment was directly installed on the host system. Nevertheless, I plan to move it into a dedicated container network and to include dedicated instances of a genome browser and a BLAT server. In the future, hence I will be able to perform tests before copying data into the production environment.

The UCSC Genome Browser implements server-side rendering, i.e. it extracts the requested data from the backend databases and renders them into pictures on the server. These graphics are then sent to the web browser on the client machine. This procedure is of double benefit, as it allows using efficient graphics algorithms on the server and to cache pictures for reuse, while keeping computing requirements low at the client side and to still display rich details.

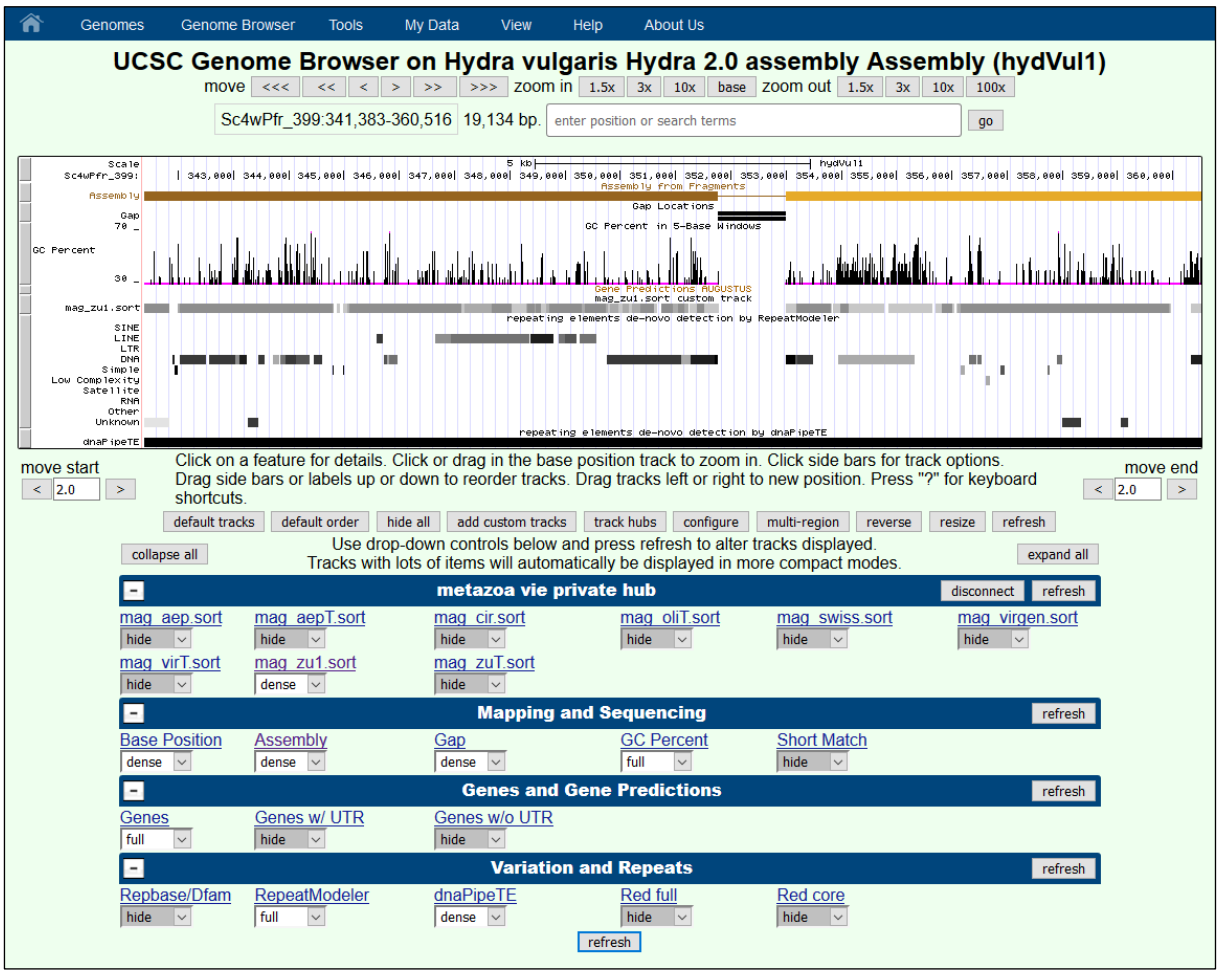


Figure 6. **UCSC Genome Browser user interface.** The central browser window displays various annotation tracks. A navigation bar on top and option buttons below this window provide access to various annotation data and different display modes.

The user interface of the UCSC Genome Browser is structured in a clear way (Figure 6). The browser window is of variable horizontal size and displays various kinds of data on tracks. Above it, navigation elements allow to move along the genome sequence and to change the magnification level. Furthermore, one may enter genomic coordinates to directly jump to some definite position or to search for annotation elements. Below the browser window, a row of

buttons for adjustment of display settings is available, followed by collapsible groups of select boxes for the rendering options of tracks.

Annotation tracks can be provided in three different ways for visualization in UCSC Genome Browser. Native tracks, such as the reference sequence, its GC content or the assembly fragments are directly stored in the backend database and immediately available to all users. Additionally, users may add custom tracks by uploading data files to the browser. These are then processed and rendered the same way as native tracks, but only available to the user herself and automatically cleared after some certain time of inactivity. Such custom tracks provide a highly efficient way for people to use their own genome annotations without running their own browser instance. To make such annotation tracks available to groups of users, as a third option one may set up a track hub. Here annotation data and track information is published on some webserver in a convenient way. Users may then use these tracks without further knowledge of browser settings. Moreover, the UCSC Genome Browser can efficiently load and cache hub data for mass usage. Hubs can be open and added to a public list in the browser, or they may be made available on some non-public URL or be otherwise restricted to a certain group of users.

One major challenge for a genome browser actually is how to visualize genomic information properly at different scales. When a large genomic area is displayed at once, vast amounts of information have to be consolidated into a picture, without too much distraction to user attention. The approach UCSC Genome Browser takes to tackle this problem, is to provide multiple views for each type of track.

Figure 7A displays a sample composition of tracks with vertical boundaries symbolized as grooves in the grey bar at the very left. The top track (a) provides the name of the chromosome or scaffold in focus and a ruler denoting relative base positions on that reference sequence. Furthermore, a scale (b) is displayed from which the magnification factor of the current view can be seen. On the track below (c) the assembly is visualized. Brown bars represent contigs displayed in pack view, a space conserving arrangement with names denoted on the left of each fragment. The next track (d) shows the gaps of the assembly in dense view, arranged on one line with half-height black horizontal bars representing gaps on the two strands of assembled DNA. At the bottom track (e) we see a full view of GC density rendered as a histogram, percentage calculated within a window of 5 bases. The purple line indicates where data is available, to distinguish between gaps in the assembly and regions of GC content below 30%.

Generally, it is important to note that tracks may be relocated within the browser window at any time without delay in rendering, as they have been received as individual graphics from the server and can be moved in a drag-and-drop manner without further involvement of the server. Only when the user zooms, moves along the vertical axis or changes the display mode of a track, a re-rendering on the server is required causing a minor delay of the browser window refresh cycle.

Figures 7B to D demonstrate changes in track rendering of the same genomic sequence at various levels of detail. Thin blue lines between windows indicate the sections displayed. Please pay attention to the scale at the top going down from 20 kb to 2 kb to 50 bases.

Scope and zoom factor of the first browser window (Fig. 7B) is identical to the previously described track composition, except some changes in track display mode. The assembly track (f) is now displayed in dense mode, i.e. all contigs are arranged at the same line. Contig colours alternate between brown and ochre to emphasize contig boundaries and gaps are indicated by thin lines. Fragment labels are omitted but may still be obtained by hovering the mouse cursor. The gap track was hidden and is not displayed anymore in these figures, as such information is well covered already by the assembly track itself. Moreover, GC percentage is displayed in dense mode now. Instead of histogram bar height, the shade of grey indicates percentage values, former peaks now denote as dark stripes. The former purple line became expendable, as gaps are well distinguishable from low percentage sections in this mode. A new track (g) has been added at the bottom displaying gene predictions. Putative gene sequences are arranged in pack mode again, with labels aligned on the left. Within a gene sequence, broad bars indicate exons connected by introns represented as thin lines. Arrows on the thin intron lines denote the strand on which the gene is located.

Zooming into the sequence renders all tracks horizontally stretched (Fig. 7C). Only two adjacent contigs are displayed and the GC percentage track reveals more variation. The gene track at the bottom now displays one complete gene sequence only. Hovering over exons and introns displays their ordinal number with the total number of such sub-sequences.

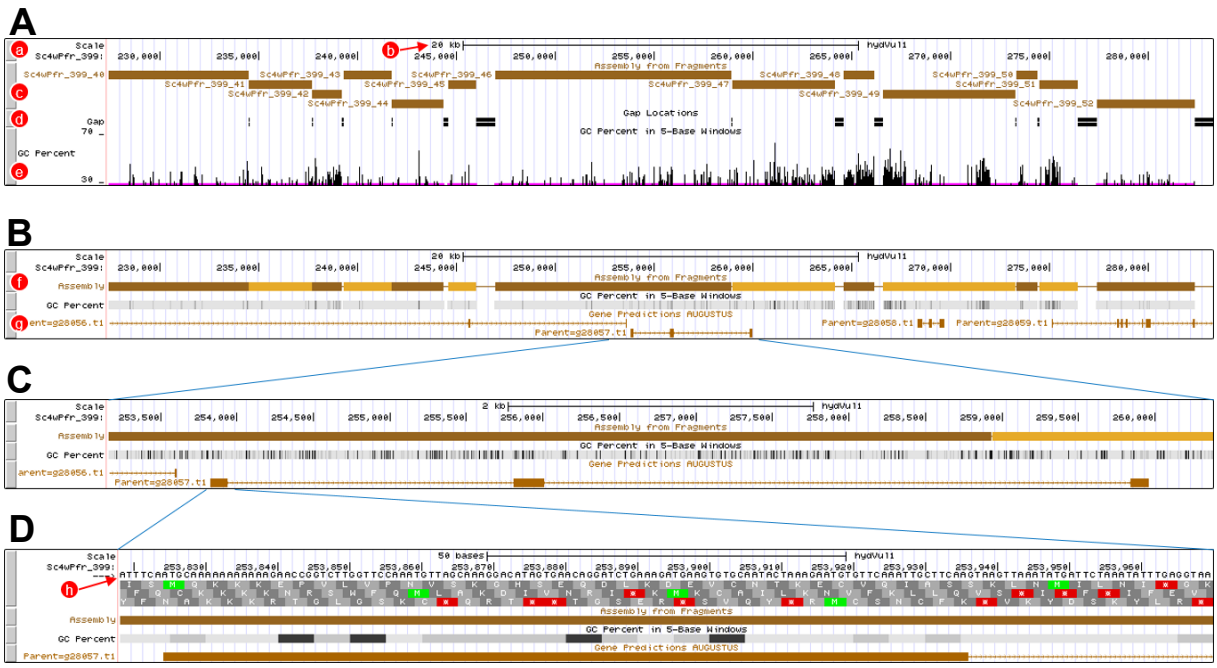


Figure 7. **Browser tracks at various levels of detail.** (A) typical browser window showing (a) reference genome, (b) scale, (c) assembly track, (d) assembly gaps and GC content proportions; (B) tracks may be displayed in compact formats such as (f) dense mode and include (g) gene predictions; (B-D) the level of detail can be increased to display the (h) actual base sequence.

Finally, after zooming deeper into the scaffold sequence, we reached a level of detail displaying the sequence of individual nucleotides (h) as well as amino acid sequences for all three possible reading frames. At this level, the assembly track has somewhat lost its expressiveness, while the GC percentage calculation procedure is now obvious, as occurrences within 5-base windows are now open to scrutiny. On the gene track, only the left exon is displayed. Comparing its boundaries against the amino acid tracks indicates that it does not perfectly match any of the reading frames.

A genome browser would be of minor benefit, if it were limited to only allow the display of the reference genome and annotations derived from it. One major feature of such browsers it to visualize alignment with sequence data from other genomes. Such alignments can be provided as tracks of assembled sequences, but it is also feasible to align large quantities of single reads to the reference genome.

Figure 8 shows various options for the alignment of reads in UCSC Genome Browser. NGS reads from the Zurich genome were first aligned with the reference genome Hm105, alignment results were stored in BAM format (H. Li et al. 2009) and uploaded to the browser as a custom

track. The aligned reads (a) are displayed in dense mode. Grey values represent the alignment quality (Fig. 8A).

To make the alignment of individual reads visible, we switched the track to pack mode and changed the colouring to indicate the strand reads are aligned to (blue for +, red for -). Misalignments are denoted as red stripes (b) within the bars (Fig. 8B). When we increase the level of detail, reads display their labels on the left and misaligned bases are denoted as characters (c) within the bars (Fig. 8C).

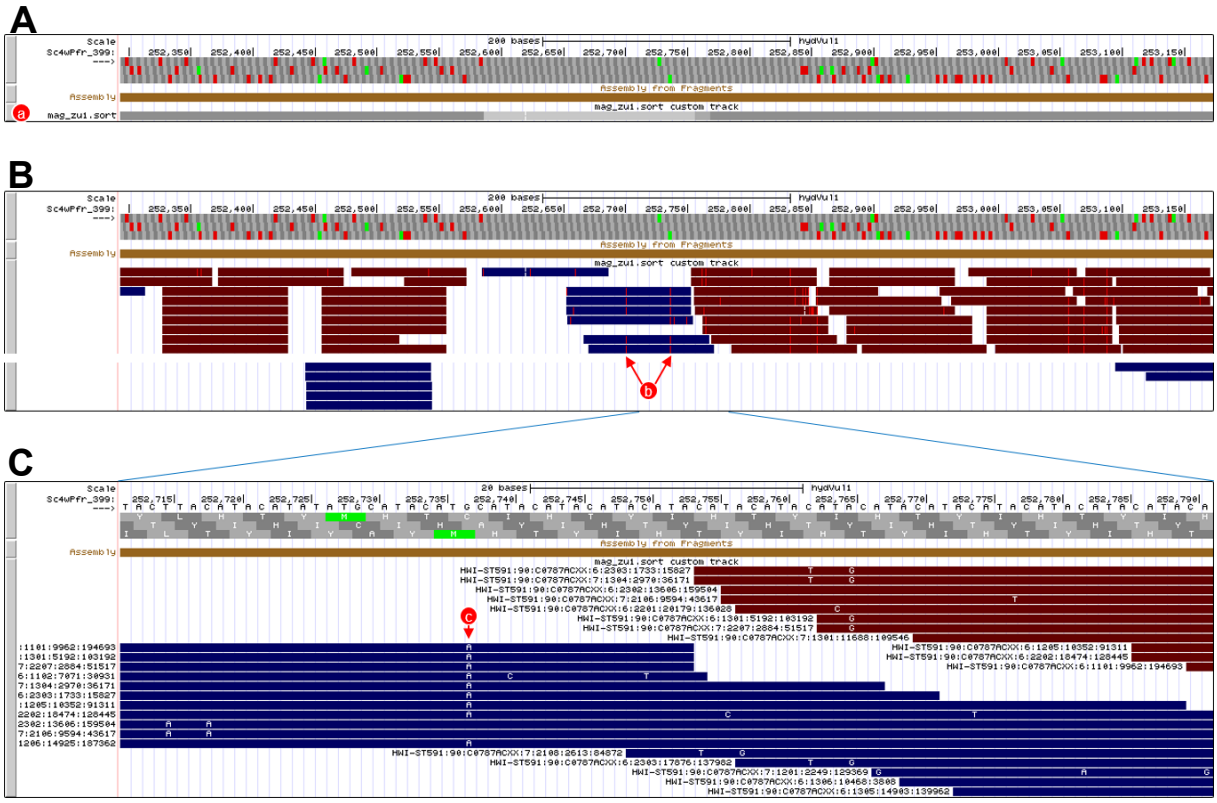


Figure 8. **Alignments of reads.** Raw reads may be displayed in compact (a) dense mode with shade of grey indicating alignment quality; (B) alternative view of packed individual reads with (b) indicators of mismatches in the alignment; (C) at high zoom level and full mode read labels are denoted and (c) mismatches are indicated as base characters.

The genome browser, which I installed for this work, is in productive state. It is available online at http://metazoa.csb.univie.ac.at/. Besides the genome of *Hydra vulgaris*, I have added several other genomes of cephalopods, such as *Octopus bimaculoides* and *Euprymna scolopes*. The browser has meanwhile become a valuable tool for people at the institute as

well as for our research partners. Accordingly, it is actively used in ongoing research projects and will be referenced in upcoming publications in the near future (Zarrella et al. in review).

Repeat tracks in the browser

Repeats may be visualized in UCSC Genome Browser by aligning them the same way as previously done with reads, one track per repeat class. One can summarize several such tracks into a composite track for better handling. Alternatively, the tools accompanying the browser allow a direct import of RepeatMasker output files into the database. This option is limited to native tracks and is not available for custom tracks or track hubs. Nevertheless, visual results are supposed to be superior as repeats are automatically grouped onto individual tracks by class and family information may be obtained for each repetitive element on mouse hover.

We now use the browser to compare repeats identified by the tools previously introduced (Fig. 9). RepeatMasker detected the first set of repeats by using consensus sequences from RepBase and Dfam attributed to taxon Hydra (parameter `-species hydra`). The other sets of repeats were identified using the de-novo libraries (parameter `-lib <library_name>`) previously generated by RepeatModeler, dnaPipeTE and Red.

At first glance all five groups of repeat tracks appear to show similar results. There are three major accumulations of repeats. On closer inspection of the left group of repetitive elements, some observations appear to be of interest. The group consists of a central repeat (b) that is 283 bp long, flanked by two simple repeats. The left one (a) is a sequence $(TATG)_{22}$ and 88 bp long, the right one (c) is a sequence $(ATAC)_{17}$ and 68 bp long. Both flanking repeats show a significant alignment and were consistently classified as simple repeats by all tools.

However, the alignment quality and classification of the 283 bp long central repeat varies. Using the RepBase/Dfam consensus library, the sequence is classified to be two adjacent or slightly overlapping DNA transposons of family CMC-EnSpm, while with RepeatModeler library, it was identified as a LINE retrotransposon of family L2 overlapping with a DNA transposon and Red was not able to classify this sequence at all. Interestingly, with dnaPipeTE a simple repeat $(ATAT)_n$ is aligned here with good quality. Alignments of other Hydra species on top of Figure 9B suggest that this sequence is specific to Hm105 genome. In the Zurich set of reads, there were no reads aligned with this part of the reference sequence, while some repeats detected by dnaPipeTE nevertheless align well.

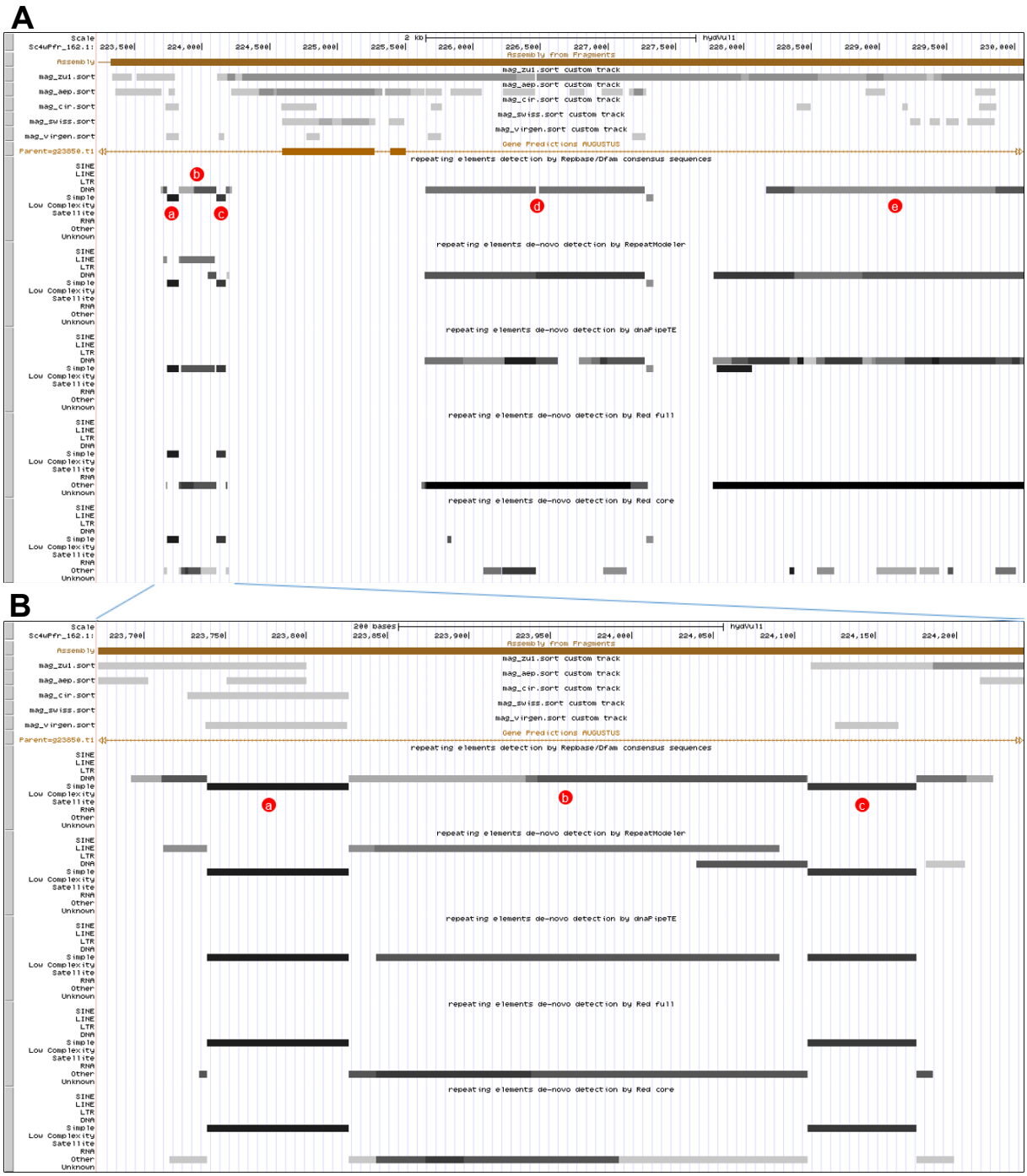


Figure 9. **Repeat tracks of various detection tools.** (A) various repeats as masked by RepeatMasker and accordingly displayed on tracks separated per class; (B) higher zoom level displaying one (b) long DNA transposon flanked by (a, c) two shorter simple repeats; (browser positions Sc4wPfr_162.1: 223,233-230,069 and Sc4wPfr_162.1:223,673-224,241)

The other two groups of repeats denoted in Figure 9A as (d) and (e) suggest that dnaPipeTE tends to identify comparatively short repeats that are then combined into nested and overlapping alignments of adjacent repeats. In contrast, Red apparently tends to favour long

and comprising sequences, frequently including several sub-sequences that RepeatModeler and dnaPipeTE classified as individual repeats. However, such insights into differences in behaviour of various tools cannot be derived from observation of a few constellations of repeats only. Further investigation will be needed to harden these initial assumptions. Nevertheless, these considerations demonstrate how visual representations of genomes and annotations upon them may be useful to gain new insights.

Figure 10 provides an example of RNAs displayed in the browser. As stated earlier, rRNA genes were only classified using the RepBase/Dfam library and are denoted on the RNA track of the browser. The de-novo tools investigated in this work do not identify rRNA sequences. The example section displays a group of three different rRNA genes (a) on the track. In contrast, RepeatMasker identifies tRNA gene sequences as a sub-class of SINEs. This classification appears reasonable, because SINEs may contain a region that is homolog to sequences of tRNA genes (Ohshima et al. 1996). For instance, the repeat (b) was contained in all libraries obtained by the different tools. As it resides at the end of a contig next to a gap, it may not be a perfect example. Nevertheless, it illustrates the case.

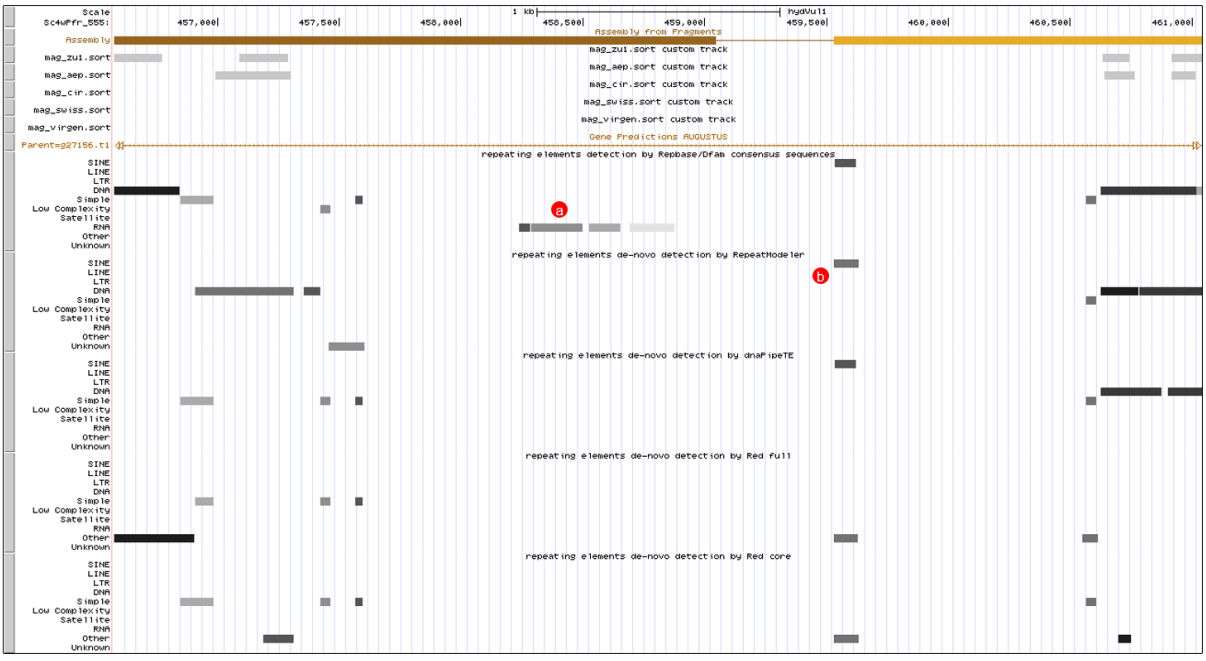


Figure 10. **RNA tracks in the genome browser.** Not all RNAs appear on the same track. (a) rRNAs from RepBase/Dfam are displayed on the RNA track of the tools own RepeatMasker mapping only; (b) tRNA is annotated as sub-class of SINE by all tools. (browser position Sc4wPfr_555:456,578-461,040)

Wnt3 and Wnt7 regions of Hydra

Wnt family of secreted proteins is considered essential in cell-to-cell signalling events of metazoan body axis formation during embryogenesis. Their genes code for complete pattern-forming systems and expression regions have a precise relation to each other (Logan & Nusse 2004; Meinhardt 2012). Investigation on these evolutionarily old and highly conserved genes and their transcriptional control has unveiled striking similarities in the molecular basis of these organizers in Hydra and higher metazoans, such as chordates (Lengfeld et al. 2009; Nakamura et al. 2011). In this work, we examine the genomic region of the Hydra ortholog of Wnt3 gene.

To allow considerations about sequence conservation, I added several alignments of some more distantly related Hydra species to the reference genome of *Hydra vulgaris* strain Hm105 in the genome browser. Of *Hydra vulgaris,* I had the Zurich strain (abbreviated zu1 in track denotation) as well as strain AEP (aep). Furthermore, I added strains of *Hydra circumcincta* (cir) and *Hydra viridissima* (virgen). I made reads of these genomes available on the track hub and mapped them to the reference genome.

Figure 11 depicts these mappings as tracks in dense display mode with grey scales denoting the quality of alignment. Immediately below the alignment tracks, the gene track depicts the Wnt3 gene as identified by AUGUSTUS software (Stanke et al. 2004). The remaining tracks in the browser window display repeats identified by the tools this work is evaluating.

Read mappings on distant Hydra species show that there is almost no conservation outside the gene region. One exception is a conserved non-coding element (Fig. 11c) in Wnt3 upstream region, potentially some regulatory element. Conservation here is even going down to the *Hydra viridissima* strain. Accordingly, I suppose there is selective pressure to keep this sequence. Furthermore, this selective pressure also keeps the 680 bp long upstream region free of repeats.

There are more findings in this region around Wnt3. RepeatMasker identified a small DNA transposon (Fig. 11a) detected by none of the other tools. This could be a novel repeat locus but possibly also nothing more than a false positive. Nearby, only RepeatModeler detected a group of three larger DNA transposons (Fig. 11b). Again, chances are that these are only false positives. Alignment quality is very good and the left TE was also partially covered by RepBase/Dfam. All tools detected small sequences of low complexity DNA within the intron regions of the Wnt3 gene (Fig. 11d). On the right side of Figure 11, all tools identified a group

of LINE elements with repeat demarcations slightly varying between them (Fig. 11f). Comparison with the aligned read suggests that these elements emerged in *Hydra vulgaris* during multiple repeat expansions, as these sequences are present in HM105 and Zurich strains. Nevertheless, two small fractions are also present in strain AEP at low alignment quality only.

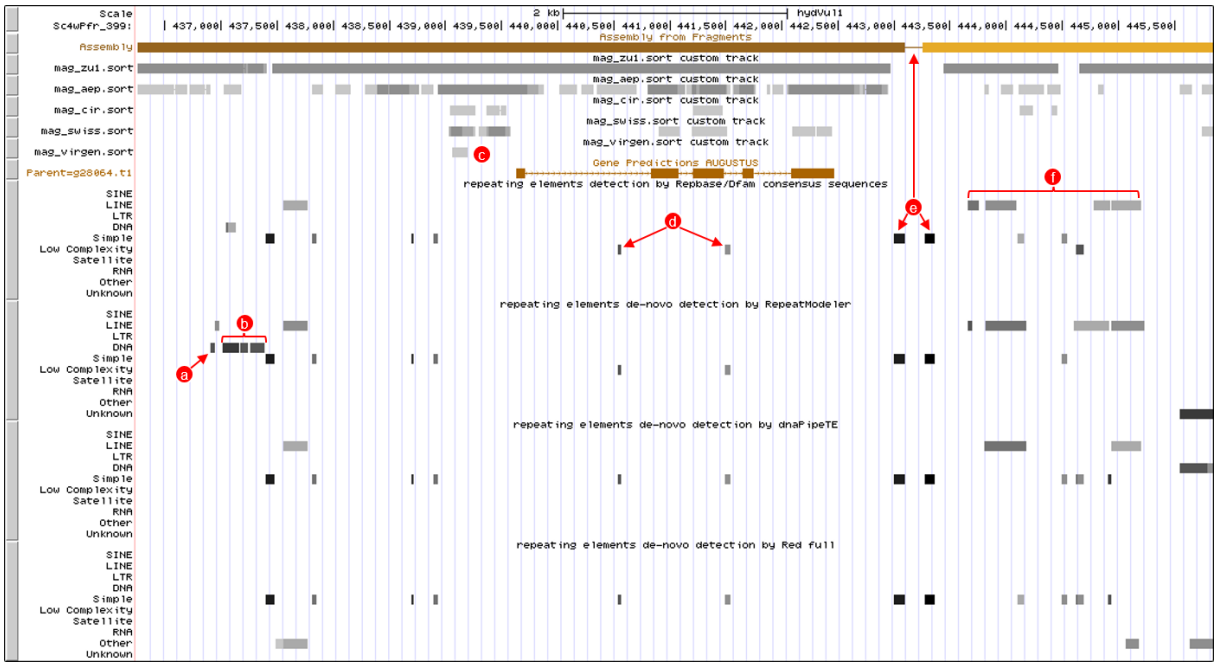


Figure 11. **Region around gene Wnt3 in *Hydra vulgaris* and related species.** (a) RepeatModeler identified more DNA transposons than other tools, but some may also be false positives; (b) group of DNA transposons only partially present in RepBase/Dfam and apparently new in *Hydra vulgaris* strain AEP; (c) conserved non-coding region, potentially a regulatory element; (d) short low complexity sequences in Wnt3 introns; (e) simple repeats flanking a gap; (f) LINE elements partially present in more *Hydra vulgaris* strains. (browser position Sc4wPfr_399:436,255-445,844 or search for gene number g28064)

Additionally, two simple repeats (Fig. 11e) may be of interest. They are characterized as $(ATAT)_n$ and $(AT)_n$ respectively which suggests sequence similarity. Nevertheless, there is a gap in the assembly separating them. This gap is possibly an artefact and nearby sequences might also be remnants of such assembly failures. This assumption is supported by the fact that this region of the assembly is not covered by any reads of other nearby Hydra species, not even the Zurich strain.

To compare the findings around Wnt3, I furthermore examined the genomic region around Wnt7 gene. Figure 12 depicts the annotations in the browser. Wnt7 is located on the opposite strain, hence the direction of transcription has to be thought as to from right to left here.

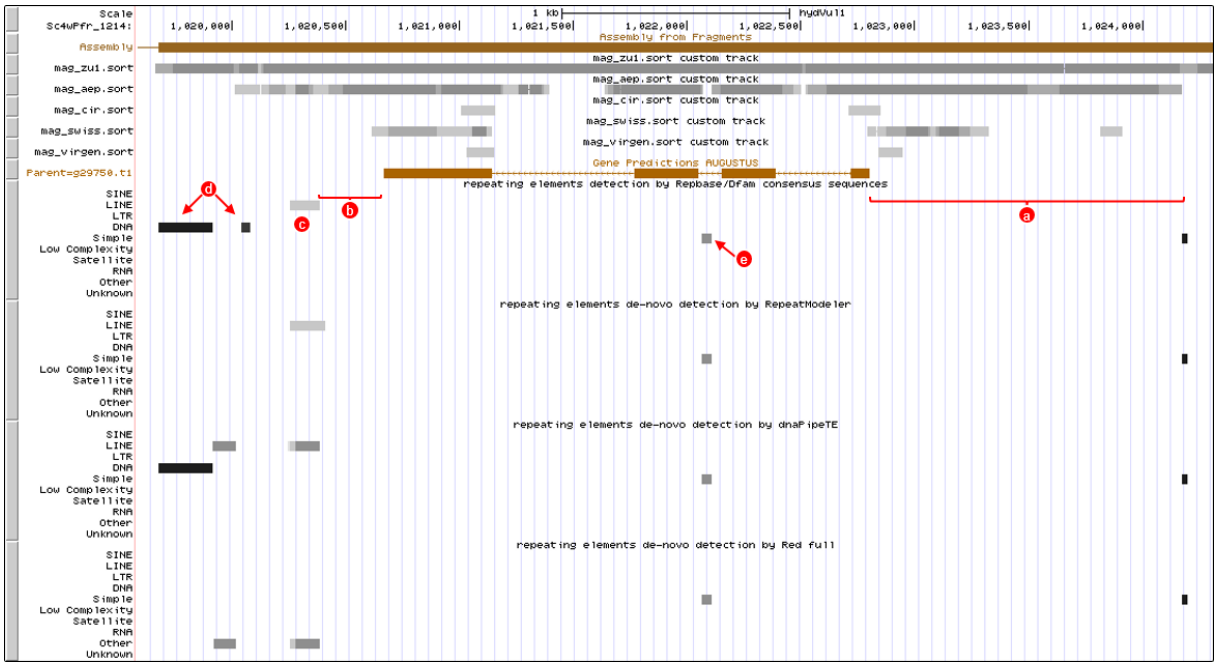


Figure 12. **Region around gene Wnt7 in *Hydra vulgaris* and related species.** (a) Upstream region is highly conserved within *Hydra vulgaris* strains Hm105, Zurich and AEP and it is free of repeats because of selective pressure; gene is on opposite strand this time, upstream region on the right side; (b) downstream region shorter but also conserved and non-repetitive; (c) LINE flanking the downstream region; (d) DNA transposons only detected by RepBase/Dfam and partially by dnaPipeTE; (e) simple repeat $(TTT)_n$ in an intron. (browser position Sc4wPfr_1214:1,019,586-1,024,310 or search for gene number g29750)

Again, we find the whole region of the gene including its 1370 bp long upstream region (Fig. 12a) and 260 bp long downstream region (Fig. 12b) to be free of repeats. However, all tools identified a short sequence within the second intron to be a simple repeat $(TTT)_n$ (Fig. 12e), whereas I suggest to interpret this as a low complexity region. Nevertheless, again mobile DNA is completely absent from this whole region, once more an indicator for the selective pressure against any disturbance of housekeeping genes and their regulatory regions. Repeats depicted on the left side of Figure 12 are a LINE retrotransposon (Fig. 12c) delimiting the otherwise repeat free downstream region of Wnt7 as well as DNA transposons (Fig. 12d) only identified by RepBase/Dfam and dnaPipeTE and completely absent in RepeatModeler. This scenario illustrates that RepeatModeler is not able to find every repeat, although results in the previous chapter suggested that it is identifying more repeats than any other tool (Fig. 2A).

Repeat dating

In the genome of every species, transposable elements represent its historical record of lineage specific expansion events and therefore, they are invaluable as “molecular fossils” in evolutionary studies. Unlike other DNA elements, interspersed repeats may not only change their positions within the genome by re-inserting themselves at arbitrary loci. They may furthermore duplicate and proliferate in several copies throughout the genome. Essentially, this process continuously increases the number of copies available over time. As any other DNA sequence, with the passing of time repeat copies thereby degenerate by random mutations and other genomic processes. Under ideal conditions, one repeat sequence is consequently spawning an exponentially growing number of copies. The sequences of these copies thereby diverge from that of their particular template and vice versa.

The idea to go the inverse way and use the divergence of sequences as a measure for time was first introduced by Zuckerkandl and Pauling in 1965 also coining the term “molecular evolutionary clock” for their hypothesis. First applied to proteins and later then to DNA, this approach proposed that the rate of evolution in a given molecule is independent of species or lineages and appears to be approximately constant over time (Zuckerkandl & Pauling 1965). Major contributions to this field are furthermore owed to Margoliash and Kimura. The former described the equidistance phenomenon, a result showing that taxa of different lineages approximately show the same genetic distance from their last common ancestor, independent of their complexity (Margoliash 1963). The latter suggested that most mutations are neutral to the phenotype and, as a consequence, are not subject of natural selection. This leaves them appropriate as a measure for evolutionary time (Kimura 1968). Nevertheless, all these theories have not been proven to be consistent until today, numerous results relativize or substantiate their validity (Ayala 1999; Scally & Durbin 2012; Yuan & Huang 2017).

However, divergence in DNA sequences i.e. the distance between them is not directly measurable from comparing sequences against each other and counting the changed nucleotides alone. Base substitutions may have happened in successions that returned to the original base at the end of a cycle. As a result, they would remain invisible to a simple comparison. To compensate for such multiple substitutions, different models of DNA sequence evolution have been proposed. Jukes and Cantor (1969) suggested the simplest such model. It assumes that bases are equally abundant in the sequence and mutate at the same rates. Subsequent research has increasingly refined models by including additional variables, such

as to differentiate between transitions and transversions (Kimura 1980) and to allow base frequencies to vary (Felsenstein 1981).

In addition to multiple substitutions, there are several other factors which possibly need correction to obtain valid results. Particularly important are AT-rich and GC-rich fragments which tend to be underrepresented in sequencing results (Benjamini & Speed 2012). This bias is problematic for distance estimations. Accordingly, appropriate correction needs to be considered early in each workflow for the dating of sequences.

Nevertheless, there are two different approaches to determine the age of repetitive sequences in the genome. I will first describe the established method and then introduce an alternative procedure.

Consensus approach

As described earlier, during identification of repetitive elements typically consensus sequences are computed to represent families of similar repeats i.e. such of putative common origin. RepeatMasker and other repeat detection tools are following this approach. A consensus sequence is an artificially generated sequence of nucleotides that is at the best possible rate similar to a set of actual sequences. The algorithmic approach to construct such a consensus sequence depends on the tools used to construct the libraries.

For RepBase/Dfam the exact procedure is not known at all, presumably repeats for new taxa are primarily identified by searching for similarity with already known sequences in the database and then let candidates undergo a human guided assessment. Automatic discovery by de-novo tool RepeatModeler is identifying consensus sequences by a combination of majority voting between best alignments and an iterative extension and evaluation strategy starting from high frequency k-mers. As dnaPipeTE is using RepeatMasker as part of its workflow, consensus sequences depend on the library used. The consensus sequences of Red are the outcome of a machine learning process that first identifies and scores putative repetitive sequences and thereby adapts its internal parameters.

In analysis of repeat expansions, every family of repeats is represented by one such consensus sequence. As all members of a repeat family are assumed descendants of a common ancestor element, the divergence of actual sequences is measured as the distance from the consensus sequence.

However, the determination of repeat ages is performed by measuring the distance between the actual repetitive element in the genome and the consensus sequence of the family this element has been classified to belong to. This approach was presumably first used for the human genome (Lander et al. 2001) and the mouse genome (Mouse Genome Sequencing Consortium 2002). Both publications show age distribution plots for repeats (Fig. 18 and Fig. 10 respectively). The latter publication goes into more detail in its supplement and describes their application of Jukes-Cantor corrected distance as a measure for sequence divergence as well as the usage of RepeatMasker output for that purpose.

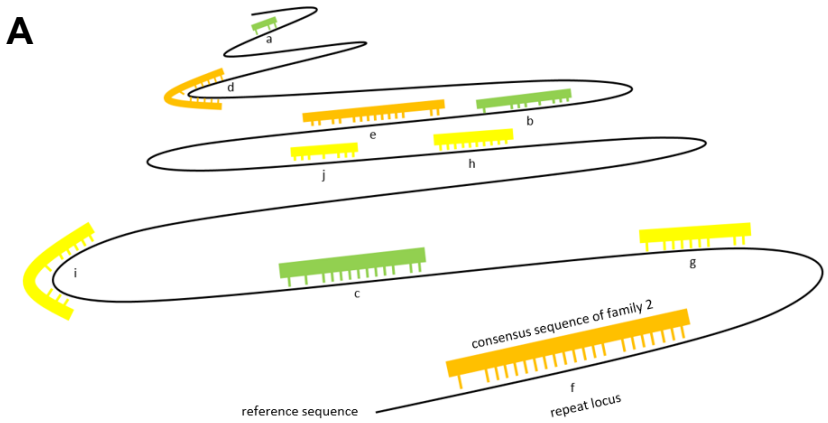
Figure 13 depicts the procedure to calculate the age distribution of repetitive elements following the consensus approach. First, the repeat loci in the reference genome are mapped to the consensus sequence of the family they had been assigned to during classification (Fig. 14A). I then calculate the distance of each locus to its family consensus sequence as the percentage of mismatches (Fig. 14B). Finally, I put the repeat loci into bins according to their distance. The number of loci of each family within each bin is counted and plotted as a stacked bar chart (Fig. 14C). Alternatively, the lengths of loci may be determined and elements in the age distribution plot may be weighted with their size in the genome (Fig. 14D).

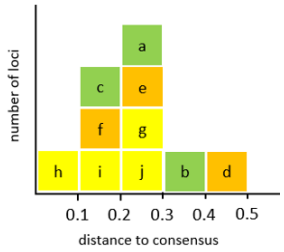
The consensus approach to repeat dating is very popular today. Furthermore, it has already been built into commonly used tools, such as RepeatMasker software.

It is important to state once again that the consensus sequence is an artefact only. It is an approximation of the common ancestor repeat sequence that spawned the family or one of its closely descendants. While repeat copies diverge from their template ancestor by base mutations and other processes in the genome, so do these ancestral elements. As an implication thereof, as each repeat sequence slightly diverges over time, multiple direct copies of it at different instants will presumably deviate in sequence at copy time already. Hence, the copy and divergence process within a family appears to be of radial nature. Paths of repeat copies are presumably not straight and linear but asymmetrically branching. Such asymmetries are caused by the fact that expansions of repeat copies happen randomly.

Moreover, copies may become inactive (e.g. by insertion of another transposable element) and be possibly reactivated after some time (e.g. by self-excision of the nested TE). Today, after repeat expansion has happened in a genome, the original repeatable sequences most likely do not exist anymore and the consensus sequences are presumably only close approximations. To continue these insights, there is furthermore no clear delimiter of families.

During sequence divergence, it may happen that some repetitive elements undergo overproportioned change and tools then misclassify them during analysis. Hence, we have to take into account that obviously repeat families may contain sequences of that are transitive copies of foreign ancestor sequences as well.





B

repeat family	consensus sequence length [bp]	repeat locus	number of matches at locus [bp]	distance to consensus = mismatches [%]	length of locus matching in consensus sequence [bp]	percentage of reference sequence matched [%]
family 1	16	a	12	25%	14	4,7%
		b	10	38%	11	2,8%
		c	13	19%	15	3,8%
family 2	25	d	14	44%	20	5,0%
		e	18	28%	21	5,3%
		f	22	12%	25	6,3%
family 3	10	g	7	30%	9	2,3%
		h	10	0%	10	2,5%
		i	8	20%	9	2,3%
		j	7	30%	7	1,8%

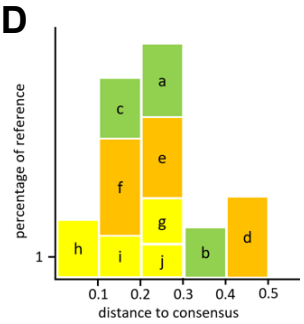


Figure 13. **Consensus approach to repeat dating.** (A) Loci of identified repeats are mapped to their family consensus sequence; (B) calculating the percentage of sequence mismatches; (C) counts of loci sorted into bins according to rate of mismatches; (D) taking lengths of matches into account.

Apart from the details of how to compute the consensus sequences, the main problem of this approach still remains that it calculates distances repeat sequences from an artefact, a fictional sequence that presumably was not the one of the ancestor repetitive element. Consensus sequences may be useful to describe a family of repeats. Nevertheless, to use them in this method of distance estimation appears not to be justified by any fundamental assumptions of biology.

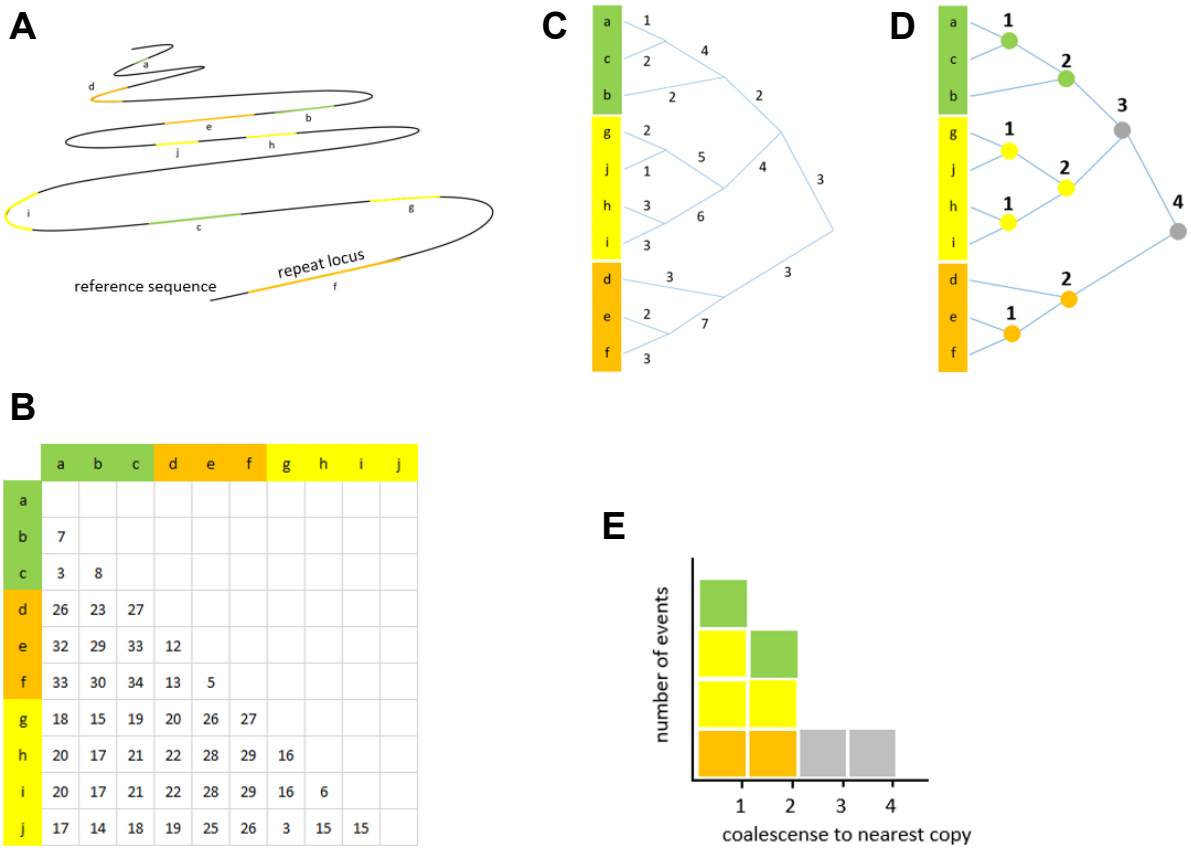
Phylogenetic approach

A more sound method to determine the age of repeat expansions is to aim for the reconstruction of the sequence phylogeny of repetitive elements. This approach results in the construction of a tree with branches representing spawning of repeat copies independently diverging over time. This tree-based approach is a completely new idea for the estimation of repeat ages. However, it appears to be well grounded on descent theory and to rest on a cornerstone of the philosophy of biology. Even as such tree-thinking may have its pitfalls (Baum et al. 2005; Sandvik 2008), it is common across all sub-disciplines of biology. Moreover, it has been successfully applied to DNA sequences of genes and was subsequently considered important for evolutionary studies (Tateno et al. 1982; Pamilo & Nei 1988; Hudson 1990). Replication of nucleotide sequences produces a bifurcating tree of genes and the same applies to the copying of transposable elements. In addition, the phylogenetic approach may rely on an extensive set of procedures and tools, nevertheless for the determination of age.

Figure 14 depicts the procedure to calculate the age distribution of repetitive elements following the phylogenetic approach. RepeatMasker using an appropriate library has previously identified repeat loci throughout the genome (Fig. 15A). First, pairwise distances between all repeat sequences are calculated and collected in a distance matrix (Fig. 15B). The whole genome is aligned to itself here. During this calculation, a correction for AT and GC bias respectively is applied. Furthermore, distances are corrected according to the Jukes-Cantor nucleotide substitution model.

Next, from the distance matrix a coalescence tree is constructed (Liu et al. 2009). Coalescent theory (Kingman 1982; Hudson 1990) aims to explain how gene variants within a population may have originated from a common ancestor. There a several degrees of strictness concerning prerequisites this theory differentiates. The simplest set assumes equal probabilities for each gene variant to be passed on to the following generation. At its core, the model attempts to merge variants back into the ancestral sequence by applying a random succession of coalescence events. This corresponds to the idea to invert the direction of time and explain the origin of actual sequence variants by inverting the processes that were responsible for their emergence. The methods developed for gene variant coalescence trees may also be applied to repeat copies. As a first step, a Neighbour-Joining tree (Saitou & Nei 1987) is built from the distance matrix. The most closely related loci are joined together and their distance is recorded in the matrix replacing both end nodes with a group node. This joining

step is iteratively executed and single nodes and group nodes are joined together until only one group node is left, i.e. the putative root (Fig. 15C).



	a	b	c	d	e	f	g	h	i	j
a										
b	7									
c	3	8								
d	26	23	27							
e	32	29	33	12						
f	33	30	34	13	5					
g	18	15	19	20	26	27				
h	20	17	21	22	28	29	16			
i	20	17	21	22	28	29	16	6		
j	17	14	18	19	25	26	3	15	15	

Figure 14. **Phylogenetic approach to repeat dating.** (A) Repeat loci throughout the genomic sequence; (B) distance matrix of loci (numbers are distance in percent); (C) Neighbour-Joining tree calculated from matrix, numbers along branches denote sub-distances according to matrix; (D) inner branching nodes represent evolutionary events; numbers denote depth i.e. age of segregation; colours symbolize attribution of events to repeat families; grey nodes are not assigned to any family; (E) plot numbers of events sorted into distance bins

However, branching nodes in the tree represent evolutionary events of sequence segregation and insertion of repeat copies. In our example, we attribute numbers to these nodes according to their depth within the tree. Furthermore, we assigned nodes to families, if all their descendant repeats are within one such family. I used colours to denote family attribution. Branching nodes in the tree, which segregate repeats of more than one family, were coloured in grey (Fig. 15D). Events were then sorted into bins again, according to their depth in the tree, i.e. distance from their nearest copy (Fig. 15E).

Typically, a vast proportion of loci is almost identical to each other and their coalescence value is almost zero. However, following this approach, such small values then become elevated to one. Then, fewer and fewer events appear at the larger distances, as underlying repeat sequences may not be identified anymore and cannot be assigned to actual families. Consequently, the method is limited to a certain extent of repeat sequence divergence. Nevertheless, I expect the shape of the age distribution plot to approach an exponential curve with most repeat loci at a very young age and then increasingly falling down to no repeats at all. As before with consensus approach, it appears important to state that the tree constructed is presumably not identical with the real descendent tree of repeat segregations.

Comparison of repeat dating approaches

After the critical notes on the consensus approach and the introduction of the alternative tree-based method, I had several expectations for the latter. First, I presume that the phylogenetic approach will provide results that better reflect the process of repeat segregation. They should apparently provide better resolution of expansion events, i.e. also smaller bursts of such expansions should become visible. Second, the repeat landscape (i.e. the age distribution plot) of the *Hydra vulgaris* genome data is skewed to the right for whatever reason. I suspect that the consensus method is responsible for this distortion. Accordingly, I expect the tree-based approach to come up without such bias. The number of repeat copies is known to grow exponentially over time. Consequently, their distribution should approximate an exponential distribution. I expect the shape of the repeat landscape plot to show a maximum of repeat copies or percentage of the genome in the most recent divergence bin at x-axis. Then numbers should decrease at exponential rate.

Materials and methods

As in previous chapters, I was using the *Hydra vulgaris* strain 105 assembly as the genome. For repeat identification, I used RepeatMasker (version open-4.0.7). I skipped the contamination check for bacterial insertion elements again when running RepeatMasker (parameter `-no_is`). The library file had been built de-novo by RepeatModeler (version open-1.0.10) previously, including NCBI/RMBLAST 2.6.0+ with default parameter values as search engine.

For the pairwise comparison of repeat loci in the phylogenetic approach, I used system default NCBI BLAST 2.6.0+. First, I built a BLAST database from the genome using default parameters. Performing a BLAST search for pairwise comparison of all repeat loci of a genome

is an enormous computation effort. It would be inappropriate to run it as a single BLAST process as computation time would be extreme. Accordingly, I prepared the search for execution as a massively parallelized set of sub-processes. I split up the set of repeat loci into several thousands of subsets and performed an individual BLAST search against the whole set for each subset. After completion of individual searches, results were combined back into one result set. This method allowed me to optimally utilize computation capacities of the cluster and is similar in intention and performance to Map-Reduce programming model (Dean & Ghemawat 2008; Maitrey & Jha 2015).

Corrections for AT/GC biases and multiple substitutions, the calculation of the distance matrices, the coalescence tree and sorting into bins were all implemented as Perl scripts, whereas plots were produced using R scripts.

Results

Comparing results in Figures 16 and 17, it is obvious that the consensus method (left panels) clearly captures up to 60% sequence identity, whereas results of the phylogenetic approach (right panels) include only very recent expansions of up to roughly 80% identity. Furthermore, numbers of repeats are definitely lower on the y-axis of tree-based approach panels. The resulting differences in curve areas clearly depict severe signal loss in the phylogenetic approach. For the phylogenetic approach, SINE retrotransposons are obviously missing completely while elements of other classes are reduced by approximately 70% (Fig. 17B). At closer look, SINEs are of course not missing. After applying the tree-based method their number is only reduced to about 2% compared to the consensus approach (Figs. 16A and B).

Per-class plots in left panels of Figure 16 depict large families of repeats (green colour) for LINE, LTR and DNA elements of various ages that apparently are missing in results of the phylogenetic approach. It has to be indicated here that black areas in plots of Figure 16 are owed to the limited capabilities of graphics resolution. Small families only take up small areas in the plots. As a result, black lines delimiting these areas are merged into a solid black area.

Figures 17B and D are of identical contour as the dataset only contained repeats of known classes. The label ‘others’ in Figure 17D means other classified repeats that do not belong to any of the enumerated superfamilies. In contrast, contour and total counts in Figure 17C are different from 17A. Unlike for the panels on the right, the dataset contained repeats not attributed to any of the main classes. Consequently, these repeats are not depicted in Figure 17A whereas they increase the total count in Figure 17C.

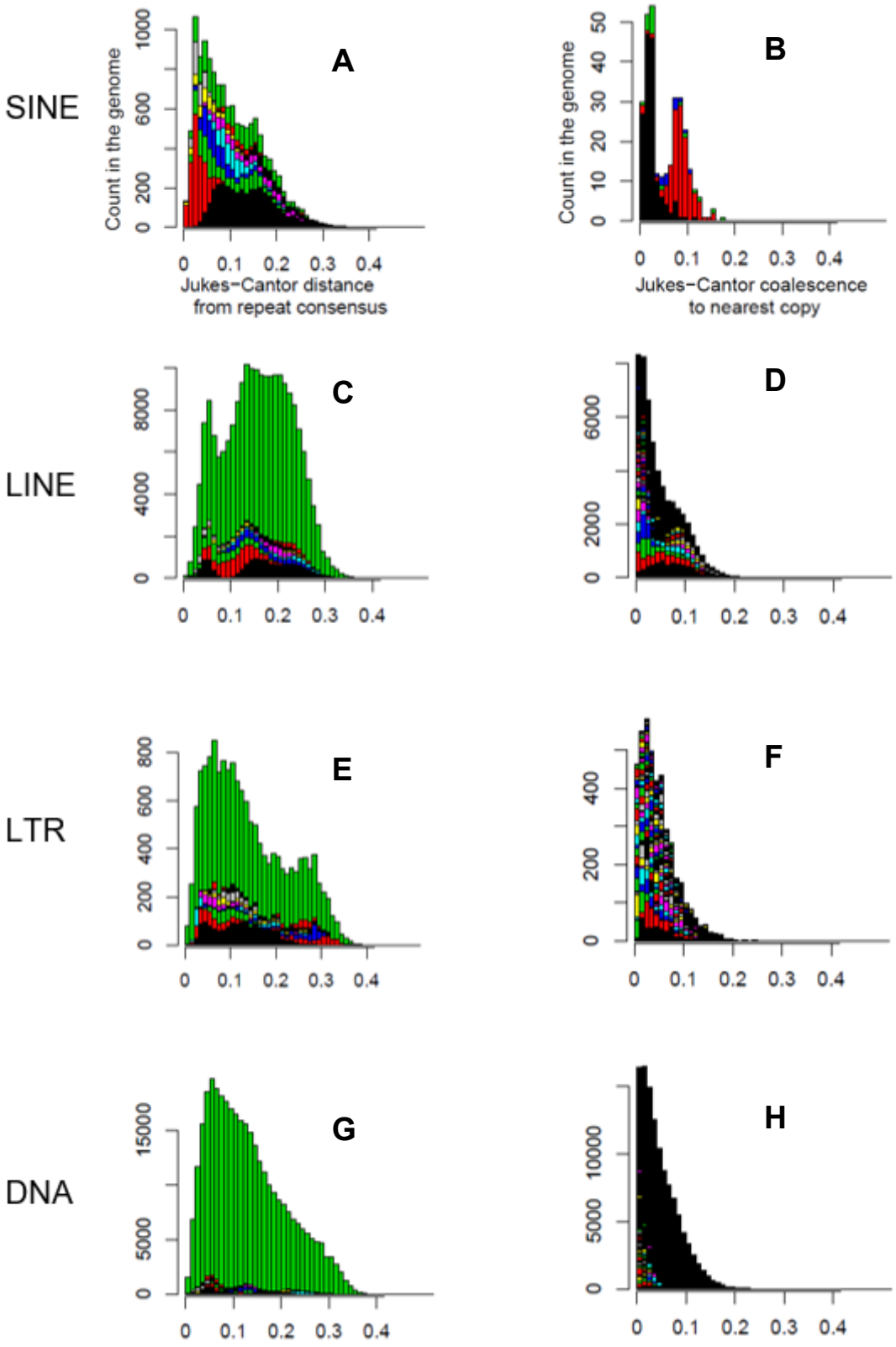


Figure 15. **Age distribution of transposable elements in the genome of *Hydra vulgaris* separated by class**. Left column depicts the consensus approach, right column the phylogenetic approach; classes are (A-B) SINE, (C-D) LINE and (E-F) LTR retrotransposons; (G-H) DNA transposons; colours represent different TE families; plots are scaled to similar sizes, please pay attention to counts denoted on y-axis.

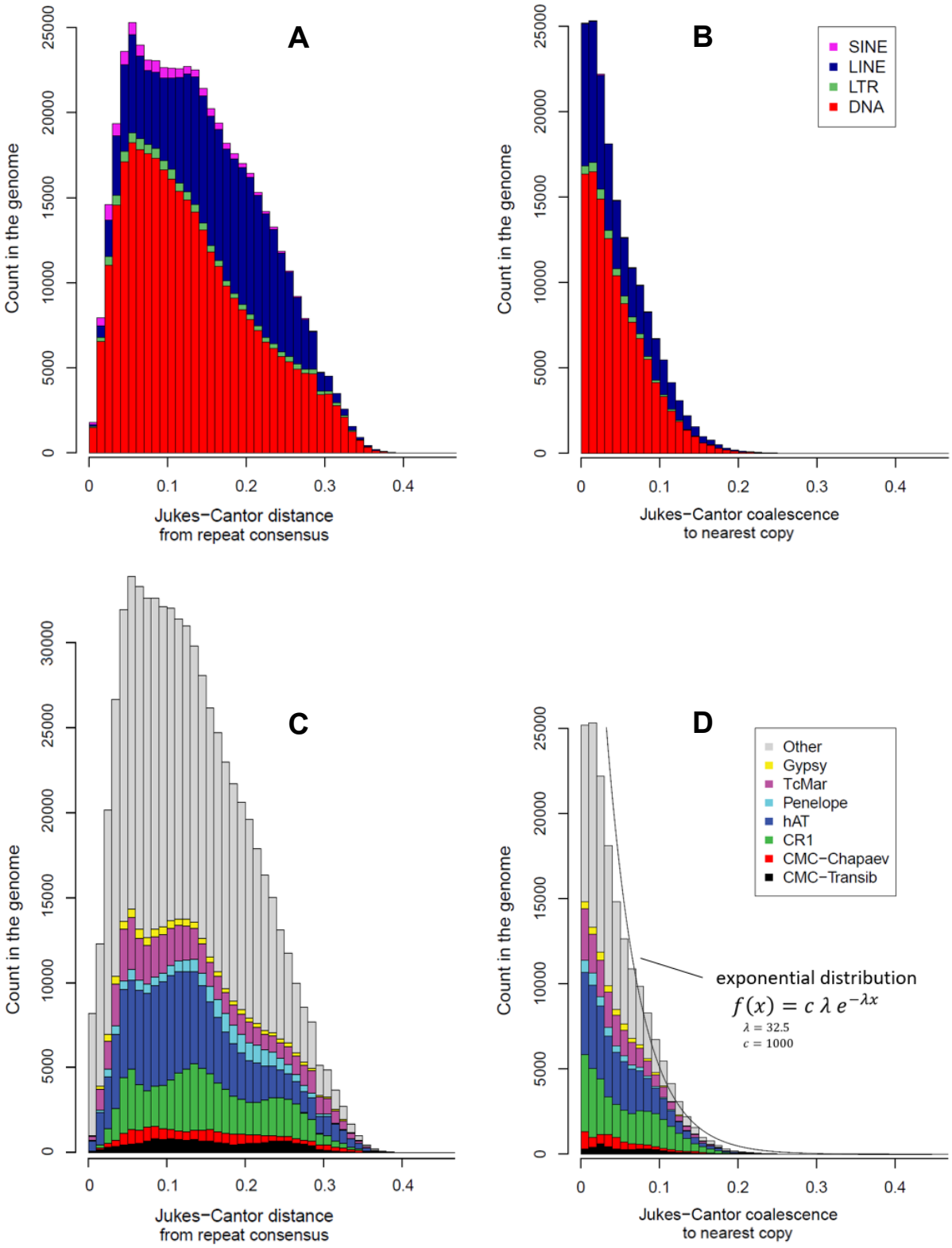


Figure 16. **Age distribution of transposable elements in the genome of *Hydra vulgaris* combined.** (A,C) left column depicts consensus approach; (B,D) right column depicts phylogenetic approach; (A,B) colours according to TE class; (C,D) superfamilies and their colours according to (Chapman et al. 2010, Fig. 1a) ; (D) approximation to exponential distribution.

Another salient difference between age distribution curves of both approaches is that consensus curves are somewhat skew to the right, with maxima apart from the left boundary at zero distance. As opposed to this, tree-based curves always show their maxima at the very left and typically a steep decrease of counts to the right. An approximation of tree-based repeat age to an exponential distribution is depicted with parameters used (Fig. 17D).

Shapes of consensus age curves at class level show more prominent peaks in general (Figs. 16C, E and G) while the plots of tree-based dating are rather uniform (Figs. 16D, F and H). An exception are the distribution curves of SINEs. They show two clear main peaks for the phylogenetic approach. At class level, repeat families seem to come and go and there is no clear stratification in these diagrams. Nevertheless, some specific elements do expand. I applied the same filtering for transposable element superfamilies and chose the same colours as in Figure 1a of the Hydra paper by Chapman et al. (2010). As a result, Figures 17C and D show clear banding. Again, the expansions in the tree-based plot are not very pronounced, whereas the consensus approach highlights three waves of expansions for repeat CR1 (Fig. 17C).

Discussion

The main point of criticism concerning consensus approach to repeat age estimation is its lack of comprehensibility in terms of philosophical justification. Consensus sequences are central to this method. However, they are only artificial sequences that probably never came into existence. Furthermore, it appears questionable if a single sequence can capture the diversity of a diverging family of repeats while completely omitting the process of repeat copy diversification. Nevertheless, the validity of the approach depends on the accurateness of consensus sequences. Misclassified elements will be compared against the consensus sequence only and never have a chance to find a closer copy outside of their family. Considering the heterogeneity of some transposable element classes, it appears likely that many repeats will consequently be assigned an incorrect age.

As an alternative, the phylogenetic approach rests on descent theory and aims to reconstruct the bifurcation tree of repeats in a very reasonable manner. All elements are compared pairwise to find close copies independently from their initial classification be RepeatMasker.

However, results obtained from the tree-based approach show some serious problems. The loss of signal is massive, I estimate that compared to the consensus method about 70% of repeats are not included in the diagrams. There are large families of repeats depicted in green

in the left panels in Figure 16 that are apparently missing in the corresponding panels at the right. I have successfully re-examined results and verified that the same RepeatMasker output data were used in both computations and that no families were accidentally excluded from result data. Nevertheless, I suppose that in the phylogenetic approach, signal loss severely reduces family sizes to values so small, these families are not depicted by a colour bar any more in the plots but by only black demarcation lines. Nevertheless, there is obviously no complete loss of elements of some specific class. SINE retrotransposons appear to be missing in Figure 17B, but their number is very small compared to numbers of other elements (Fig. 16). As a result, their bars are not visible in the plots. Another shortcoming presumably owed to signal loss is that there are only few prominent peaks compared to the consensus method. Expansions in the tree-based plots are not very pronounced, however they are real.

Another drawback of the phylogenetic approach is that it captures only very recent expansions up to roughly 80% identity. In contrast, the consensus method includes repeats of up to only 60% identity. These properties appear to be intrinsic to the tree-based method itself, there is no evidence that signals beyond 20% distance are lost, but that the target range of this method is somewhat biased to lower values. An implication of this observation is that with large repeat families and the limitations of this method, no grey nodes as depicted in Figure 14 D and E were detected in the actual real data.

On the other side, age distribution curves obtained from the consensus method give the impression to be skew. Values are apparently shifted to the right and do not show a peak at zero. This is opposed to the expectation that numbers of younger repeat copies are higher and that the most recent ones are also the most abundant in a genome. Tree-based age curves are clearly showing an exponential distribution (Fig. 17D) whereas those obtained from consensus method are not.

I am not sure how to circumvent these problems. Overall, signal loss and its resulting problems may be caused by the tools used or by insufficient quality of data. It could also be intrinsic to the tree-based method itself. Maybe one could improve the sensitivity of the method has by introducing a more relaxed mapping than BLAST can offer.

One important factor could be the length of reads. With the consensus approach, short repeat sequences as obtained from Illumina sequencing map at long consensus sequences. With the tree approach, sequences to compare with may also be short, nevertheless they are compared against each other. For short sequences, there is less chance to overlap and to carry any

matching part. Then, a calculation of distance is not possible and the pairing is lost for the tree-based approach. Two observations are supporting this assumption. First, the repeat age distribution in Figure 1a of Chapman et al. (2010) was computed following the consensus approach. However, it is showing a flat exponential distribution with a maximum close to the left and prominent peaks. Reads used for this research paper were substantially longer as they were obtained by applying Sanger sequencing and then ReAS (R. Li et al. 2005) was used to identify repeats. Another research paper is taking the same line with PacBio sequencing producing significantly longer reads. The age curve obtained by consensus approach is again not skew and shows a maximum for recent insertions at the left (Ou & Jiang 2018, Fig. 5A).

At the end, both approaches have their advantages and disadvantages, none is as desired. The tree-based method is well justified, and age distribution has its peak at zero and follows an exponential curve as I would expect from copy/paste transposons. However, the high loss of signal makes it not usable as a method at this stage. The exploration of repetitive elements for better understanding of their evolutionary dynamics is an evolving field and we will possibly see future improvements in methods.

Conclusion

This work demonstrated the use of genome informatics for gaining insight into the genome of Hydra and its evolutionary history by means of repetitive elements. I first compared different approaches for the detection and classification of repeats and evaluated tools. One major outcome is that RepeatModeler is still on top for de-novo construction of repeat libraries. While my work confirms its standing as one of the best repeat identification tools, I also show that it does not identity all repeats found by other tools. A combination of several repeat mining approaches may therefore be necessary for an accurate genome annotation. For building libraries from raw reads, dnaPipeTE appears to be first choice. While I could demonstrate that proportions of repeat classes detected from raw reads may clearly deviate from such identified from an assembly, insufficient read coverage may require the use of dnaPipeTE that may potentially perform equally well or better than RepeatModeler. Furthermore, machine learning did not answer my expectations in terms of repeat detection. Red software using Hidden Markov Models appears biased to a greedy detection strategy and tends to combine several unrelated repeats into a single repeat annotation. I then compared libraries generated by various tools and the repeats that I identified utilizing these libraries. Here, an important finding is that similarity between consensus sequence libraries does not imply similar results in detecting actual repeats in the genome.

Next in this work, I explored the capabilities of UCSC Genome Browser to annotate and visualize genomic data. I demonstrated a lightweight approach to install the software and to satisfy its dependencies on partially interfering software packages. Furthermore, I used the browser to examine distribution patterns of repeats in the neighbourhood of essential genes. My findings suggest that selective pressure prevents the insertion of repeats close to the regulatory and coding regions of essential genes. The genome browser I installed is in productive state and available online. I have added several other genomes and annotation tracks. It has meanwhile become a valuable tool for people at the institute as well as for our research partners.

In the last part of my work, I reviewed the common approach to date repeat expansions. I critically questioned use of consensus sequences for estimating ages of repeat copies. As an alternative, I introduced a method that utilizes a coalescence tree to reconstruct segregation of repeat copies and estimates age from distances between copies in this tree. However, results disclosed shortcomings in both approaches. The major outcome here is that even as

the phylogenetic method appears to be more viable, quality of actual results still prevents its practical use.

Nevertheless, first seen as impediments from a gene-centric point of view, repetitive elements nowadays appear essential to understand the structure and the evolutionary history of genomes. Notwithstanding that numerous approaches and classification schemes have already been developed, we will most likely need novel methods to better detect, classify and annotate repeats (Arensburger et al. 2016). An important topic here is that most tools can locate repetitive sequences only if they are complete extant and not fragmented, overlapping or nested with other repeats. Few tools only are up to allow for such complex constellations. One example is RepeatCraft (Wong & Simakov 2018), which facilitates de-fragmentation of repetitive sequences. Today's genome informatics has more powerful machines than ever before. Nevertheless, these are primarily utilized for coping with the vast volumes of data. Furthermore, high dimensional interactions of genomic processes and interactions between elements may be overwhelming to the human mind. After a decades-long dry spell, machine learning now seems to become a much needed assistant in science (Wu & McLarty 2012; Oustimov & Vu 2014). Apparently, for genome visualization major challenges are to come up with innovative approaches to render spatial genomic data as well as to bring time-dependencies into browsers (Goodstadt & Marti-Renom 2017; Waldispühl et al. 2018). Concerning data quality, upcoming sequencing technologies promise to provide longer high quality reads and will substantially improve genome assemblies. In the light of such innovations, it seems likely that the approaches presented in this work can be further developed accordingly.

References

Achaz, G., Boyer, F., Rocha, E. P. C., Viari, A., Coissac, E. , 2007. Repseek, a tool to retrieve approximate repeats from large DNA sequences. Bioinformatics 23(1), pp. 119–121.

Alhakami, H., Mirebrahim, H., Lonardi, S. , 2017. A comparative evaluation of genome assembly reconciliation tools. Genome Biology 18.

Altschul, S. F., Boguski, M. S., Gish, W., Wootton, J. C. , 1994. Issues in searching molecular sequence databases. Nature Genetics 6(2), pp. 119–129.

Andrieu, O., Fiston, A.-S., Anxolabéhère, D., Quesneville, H. , 2004. Detection of transposable elements by their compositional bias. BMC Bioinformatics 5(1), pp. 94.

Arensburger, P., Piégu, B., Bigot, Y. , 2016. The future of transposable element annotation and their classification in the light of functional genomics - what we can learn from the fables of Jean de la Fontaine? Mobile Genetic Elements 6(6), pp. e1256852.

Ayala, F. J. , 1999. Molecular clock mirages. BioEssays 21(1), pp. 71–75.

Bao, W., Kojima, K. K., Kohany, O. , 2015. Repbase Update, a database of repetitive elements in eukaryotic genomes. Mobile DNA 6(1), pp. 11.

Bao, Z., Eddy, S. R. , 2002. Automated De Novo Identification of Repeat Sequence Families in Sequenced Genomes. Genome Research 12(8), pp. 1269–1276.

Baum, D. A., Smith, S. D., Donovan, S. S. S. , 2005. The Tree-Thinking Challenge. Science 310(5750), pp. 979–980.

Benjamini, Y., Speed, T. P. , 2012. Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Research 40(10), pp. e72–e72.

Bergman, C. M., Quesneville, H. , 2007. Discovering and detecting transposable elements in genome sequences. Briefings in Bioinformatics 8(6), pp. 382–392.

Boettiger, C. , 2015. An Introduction to Docker for Reproducible Research. SIGOPS Oper. Syst. Rev. 49(1), pp. 71–79.

Britten, R. J., Kohne, D. E. , 1968. Repeated Sequences in DNA. Science 161(3841), pp. 529–540.

Buenrostro, J. D., Wu, B., Chang, H. Y., Greenleaf, W. J. , 2015. ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. Current Protocols in Molecular Biology 109(1), pp. 21.29.1-21.29.9.

Casper, J., Zweig, A. S., Villarreal, C., Tyner, C., Speir, M. L., Rosenbloom, K. R., Raney, B. J., Lee, C. M., Lee, B. T., Karolchik, D., Hinrichs, A. S., Haeussler, M., Guruvadoo, L., Navarro Gonzalez, J., Gibson, D., Fiddes, I. T., Eisenhart, C., Diekhans, M., Clawson, H., Barber, G. P., Armstrong, J., Haussler, D., Kuhn, R. M., Kent, W. J. , 2018. The UCSC Genome Browser database: 2018 update. Nucleic Acids Research 46(D1), pp. D762–D769.

Caspi, A., Pachter, L. , 2006. Identification of transposable elements using multiple alignments of related genomes. Genome Research 16(2), pp. 260–270.

Castillo-Davis, C. I. , 2005. The evolution of noncoding DNA: how much junk, how much func? Trends in Genetics 21(10), pp. 533–536.

Chapman, J. A., Kirkness, E. F., Simakov, O., Hampson, S. E., Mitros, T., Weinmaier, T., Rattei, T., Balasubramanian, P. G., Borman, J., Busam, D., Disbennett, K., Pfannkoch, C., Sumin, N., Sutton, G. G., Viswanathan, L. D., Walenz, B., Goodstein, D. M., Hellsten, U., Kawashima, T., Prochnik, S. E., Putnam, N. H., Shu, S., Blumberg, B., Dana, C. E., Gee, L., Kibler, D. F., Law, L., Lindgens, D., Martinez, D. E., Peng, J., Wigge, P. A., Bertulat, B., Guder, C., Nakamura, Y., Ozbek, S., Watanabe, H., Khalturin, K., Hemmrich, G., Franke, A., Augustin, R., Fraune, S., Hayakawa, E., Hayakawa, S., Hirose, M., Hwang, J. S., Ikeo, K., Nishimiya-Fujisawa, C.,

Ogura, A., Takahashi, T., Steinmetz, P. R. H., Zhang, X., Aufschnaiter, R., Eder, M.-K., Gorny, A.-K., Salvenmoser, W., Heimberg, A. M., Wheeler, B. M., Peterson, K. J., Böttger, A., Tischler, P., Wolf, A., Gojobori, T., Remington, K. A., Strausberg, R. L., Venter, J. C., Technau, U., Hobmayer, B., Bosch, T. C. G., Holstein, T. W., Fujisawa, T., Bode, H. R., David, C. N., Rokhsar, D. S., Steele, R. E. , 2010. The dynamic genome of Hydra. Nature 464(7288), pp. 592–596.

Choo, K. H. A. , 2001. Domain Organization at the Centromere and Neocentromere. Developmental Cell 1(2), pp. 165–177.

Chuong, E. B., Elde, N. C., Feschotte, C. , 2017. Regulatory activities of transposable elements: from conflicts to benefits. Nature Reviews Genetics 18(2), pp. 71–86.

Dean, J., Ghemawat, S. , 2008. MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51(1), pp. 107–113.

Dekker, J., Rippe, K., Dekker, M., Kleckner, N. , 2002. Capturing Chromosome Conformation. Science 295(5558), pp. 1306–1311.

Dorer, D. R., Henikoff, S. , 1994. Expansions of transgene repeats cause heterochromatin formation and gene silencing in Drosophila. Cell 77(7), pp. 993–1002.

Durbin, R., Eddy, S. R., Krogh, A., Mitchison, G. , 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

Ehret, C. F., De Haller, G. , 1963. Origin, development, and maturation of organelles and organelle systems of the cell surface in Paramecium. Journal of Ultrastructure Research 9, pp. 1–42.

Elbarbary, R. A., Lucas, B. A., Maquat, L. E. , 2016. Retrotransposons as regulators of gene expression. Science 351(6274), pp. aac7247.

Ellinghaus, D., Kurtz, S., Willhoeft, U. , 2008. LTRharvest, an efficient and flexible software for de novo detection of LTR retrotransposons. BMC Bioinformatics 9(1), pp. 18.

Ewing, B., Green, P. , 1998. Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. Genome Research 8(3), pp. 186–194.

Ewing, B., Hillier, L., Wendl, M. C., Green, P. , 1998. Base-Calling of Automated Sequencer Traces UsingPhred. I. Accuracy Assessment. Genome Research 8(3), pp. 175–185.

Federhen, S. , 2012. The NCBI Taxonomy database. Nucleic Acids Research 40(D1), pp. D136–D143.

Federhen, S. , 2015. Type material in the NCBI Taxonomy Database. Nucleic Acids Research 43(D1), pp. D1086–D1098.

Felsenstein, J. , 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. Journal of Molecular Evolution 17(6), pp. 368–376.

Feschotte, C. , 2008. Transposable elements and the evolution of regulatory networks. Nature Reviews Genetics 9(5), pp. 397–405.

Feschotte, C., Pritham, E. J. , 2007. DNA Transposons and the Evolution of Eukaryotic Genomes. Annual Review of Genetics 41(1), pp. 331–368.

Finnegan, D. J. , 1989. Eukaryotic transposable elements and genome evolution. Trends in Genetics 5, pp. 103–107.

Girgis, H. Z. , 2015. Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. BMC Bioinformatics 16, pp. 227.

Goerner-Potvin, P., Bourque, G. , 2018. Computational tools to unmask transposable elements. Nature Reviews Genetics pp. 1.

Goodstadt, M., Marti-Renom, M. A. , 2017. Challenges for visualizing three-dimensional data in genomic browsers. FEBS Letters pp. 2505–2519.

Goubert, C., Modolo, L., Vieira, C., ValienteMoro, C., Mavingui, P., Boulesteix, M. , 2015. De Novo Assembly and Annotation of the Asian Tiger Mosquito (Aedes albopictus) Repeatome with dnaPipeTE from Raw Genomic Reads and Comparative Analysis with the Yellow Fever Mosquito (Aedes aegypti). Genome Biology and Evolution 7(4), pp. 1192–1205.

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., Palma, F. di, Birren, B. W., Nusbaum, C., Lindblad-Toh, K., Friedman, N., Regev, A. , 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nature Biotechnology 29(7), pp. 644–652.

Greenblatt, I. M., Brink, R. A. , 1962. Twin Mutations in Medium Variegated Pericarp Maize. Genetics 47(4), pp. 489–501.

Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., Eddy, S. R. , 2003. Rfam: an RNA family database. Nucleic Acids Research 31(1), pp. 439–441.

Harrison, P. M., Gerstein, M. , 2002. Studying Genomes Through the Aeons: Protein Families, Pseudogenes and Proteome Evolution. Journal of Molecular Biology 318(5), pp. 1155–1174.

Hemmrich, G., Anokhin, B., Zacharias, H., Bosch, T. C. G. , 2007. Molecular phylogenetics in Hydra, a classical model in evolutionary developmental biology. Molecular Phylogenetics and Evolution 44(1), pp. 281–290.

Hoen, D. R., Hickey, G., Bourque, G., Casacuberta, J., Cordaux, R., Feschotte, C., Fiston-Lavier, A.-S., Hua-Van, A., Hubley, R., Kapusta, A., Lerat, E., Maumus, F., Pollock, D. D., Quesneville, H., Smit, A., Wheeler, T. J., Bureau, T. E., Blanchette, M. , 2015. A call for benchmarking transposable element annotation methods. Mobile DNA 6, pp. 13.

Holstein, T. W., Hobmayer, E., Technau, U. , 2003. Cnidarians: An evolutionarily conserved model system for regeneration? Developmental Dynamics 226(2), pp. 257–267.

Huang, C. R. L., Burns, K. H., Boeke, J. D. , 2012. Active Transposition in Genomes. Annual Review of Genetics 46, pp. 651–675.

Hubley, R., Finn, R. D., Clements, J., Eddy, S. R., Jones, T. A., Bao, W., Smit, A. F. A., Wheeler, T. J. , 2016. The Dfam database of repetitive DNA families. Nucleic Acids Research 44(D1), pp. D81–D89.

Hudson, R. R. , 1990. Gene genealogies and the coalescent process. In *Oxford Surveys in Evolutionary Biology* (Vol. 7, pp. 1–44). Oxford: Oxford University Press.

Ide, S., Miyazaki, T., Maki, H., Kobayashi, T. , 2010. Abundance of Ribosomal RNA Gene Copies Maintains Genome Integrity. Science 327(5966), pp. 693–696.

Ito, T. , 1947. A new fresh-water polyp, Hydra magnipapillata, n. sp. From Japan. Sci. Rep. Tohoku University, 4th Ser. (Biol.) 18, pp. 6–10.

Joy, A. M. , 2015. Performance comparison between Linux containers and virtual machines. In *2015 International Conference on Advances in Computer Engineering and Applications* (pp. 342–346).

Jurka, J. , 1998. Repeats in genomic DNA: mining and meaning. Current Opinion in Structural Biology 8(3), pp. 333–337.

Kapitonov, V. V., Jurka, J. , 2007. Helitrons on a roll: eukaryotic rolling-circle transposons. Trends in Genetics 23(10), pp. 521–529.

Kapitonov, V. V., Jurka, J. , 2008. A universal classification of eukaryotic transposable elements implemented in Repbase. Nature Reviews Genetics 9(5), pp. 411–412.

Kawaida, H., Shimizu, H., Fujisawa, T., Tachida, H., Kobayakawa, Y. , 2010. Molecular phylogenetic study in genus Hydra. Gene 468(1), pp. 30–40.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., Haussler, and D. , 2002. The Human Genome Browser at UCSC. Genome Research 12(6), pp. 996–1006.

Kidwell, M. G. , 2002. Transposable elements and the evolution of genome size in eukaryotes. Genetica 115(1), pp. 49–63.

Kimura, M. , 1968. Evolutionary rate at the molecular level. Nature 217(5129), pp. 624–626.

Kimura, M. , 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. Journal of Molecular Evolution 16(2), pp. 111–120.

Kingman, J. F. C. , 1982. On the genealogy of large populations. Journal of Applied Probability 19(A), pp. 27–43.

Kramerov, D. A., Vassetzky, N. S. , 2005. Short Retroposons in Eukaryotic Genomes. In *International Review of Cytology* (Vol. 247, pp. 165–221). Academic Press.

Lagesen, K., Hallin, P., Rødland, E. A., Stærfeldt, H.-H., Rognes, T., Ussery, D. W. , 2007. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Research 35(9), pp. 3100–3108.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J. P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, Y., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J. C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R. H., Wilson, R. K., Hillier, L. W., McPherson, J. D., Marra, M. A., Mardis, E. R., Fulton, L. A., Chinwalla, A. T., Pepin, K. H., Gish, W. R., Chissoe, S. L., Wendl, M. C., Delehaunty, K. D., Miner, T. L., Delehaunty, A., Kramer, J. B., Cook, L. L., Fulton, R. S., Johnson, D. L., Minx, P. J., Clifton, S. W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J. F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R. A., Muzny, D. M., Scherer, S. E., Bouck, J. B., Sodergren, E. J., Worley, K. C., Rives, C. M., Gorrell, J. H., Metzker, M. L., Naylor, S. L., Kucherlapati, R. S., Nelson, D. L., Weinstock, G. M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D. R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H. M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R. W., Federspiel, N. A., Abola, A. P., Proctor, M. J., Myers, R. M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D. R., Olson, M. V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G. A., Athanasiou, M., Schultz, R., Roe, B. A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W. R., de la Bastide, M., Dedhia, N., Blöcker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J. A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D. G., Burge, C. B., Cerutti, L., Chen, H. C., Church, D., Clamp, M., Copley, R. R., Doerks, T., Eddy, S. R., Eichler, E. E., Furey, T. S., Galagan, J., Gilbert, J. G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L. S., Jones, T. A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W. J., Kitts, P., Koonin, E. V., Korf, I., Kulp, D., Lancet, D., Lowe, T. M., McLysaght, A., Mikkelsen, T., Moran, J. V., Mulder, N., Pollara, V. J., Ponting, C. P., Schuler, G., Schultz, J., Slater, G., Smit, A. F., Stupka, E., Szustakowki, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y. I., Wolfe, K. H., Yang, S. P., Yeh, R. F., Collins, F., Guyer, M. S., Peterson, J., Felsenfeld, A., Wetterstrand, K. A., Patrinos, A., Morgan, M. J., de Jong, P., Catanese, J. J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y. J., Szustakowki, J., International Human Genome Sequencing Consortium. , 2001. Initial sequencing and analysis of the human genome. Nature 409(6822), pp. 860–921.

Leclère, L., Copley, R. R., Momose, T., Houliston, E. , 2016. Hydrozoan insights in animal development and evolution. Current Opinion in Genetics & Development 39, pp. 157–167.

Lee, R. Y. N., Howe, K. L., Harris, T. W., Arnaboldi, V., Cain, S., Chan, J., Chen, W. J., Davis, P., Gao, S., Grove, C., Kishore, R., Muller, H.-M., Nakamura, C., Nuin, P., Paulini, M., Raciti, D., Rodgers, F., Russell, M., Schindelman, G., Tuli, M. A., Van Auken, K., Wang, Q., Williams, G., Wright, A., Yook, K., Berriman, M., Kersey, P., Schedl, T., Stein, L., Sternberg, P. W. , 2018. WormBase 2017: molting into a new stage. Nucleic Acids Research 46(D1), pp. D869–D874.

Lengfeld, T., Watanabe, H., Simakov, O., Lindgens, D., Gee, L., Law, L., Schmidt, H. A., Ozbek, S., Bode, H., Holstein, T. W. , 2009. Multiple Wnts are involved in Hydra organizer formation and regeneration. Developmental Biology 330(1), pp. 186–199.

Lerat, E. , 2010. Identifying repeats and transposable elements in sequenced genomes: how to find your way through the dense forest of programs. Heredity 104(6), pp. 520–533.

Levin, H. L., Moran, J. V. , 2011. Dynamic interactions between transposable elements and their hosts. Nature Reviews Genetics 12(9), pp. 615–627.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. , 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25(16), pp. 2078–2079.

Li, R., Ye, J., Li, S., Wang, J., Han, Y., Ye, C., Wang, J., Yang, H., Yu, J., Wong, G. K.-S., Wang, J. , 2005. ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. PLoS Computational Biology 1(4), pp. e43.

Liou, C.-Y., Tseng, S.-H., Cheng, W.-C., Tsai, H.-Y. , 2013. Structural Complexity of DNA Sequence [Research article].

Liu, L., Yu, L., Kubatko, L., Pearl, D. K., Edwards, S. V. , 2009. Coalescent methods for estimating phylogenetic trees. Molecular Phylogenetics and Evolution 53(1), pp. 320–328.

Logan, C. Y., Nusse, R. , 2004. The Wnt Signaling Pathway in Development and Disease. Annual Review of Cell and Developmental Biology 20(1), pp. 781–810.

Lopez, R., Silventoinen, V., Robinson, S., Kibria, A., Gish, W. , 2003. WU-Blast2 server at the European Bioinformatics Institute. Nucleic Acids Research 31(13), pp. 3795–3798.

Lowe, T. M., Eddy, S. R. , 1997. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Research 25(5), pp. 955–964.

Maitrey, S., Jha, C. K. , 2015. MapReduce: Simplified Data Analysis of Big Data. Procedia Computer Science 57, pp. 563–571.

Makaowski, W. , 2000. Genomic scrap yard: how genomes utilize all that junk. Gene 259(1), pp. 61–67.

Malone, C. D., Hannon, G. J. , 2009. Small RNAs as Guardians of the Genome. Cell 136(4), pp. 656–668.

Margoliash, E. , 1963. Primary Structure and Evolution of Cytochrome C. Proceedings of the National Academy of Sciences 50(4), pp. 672–679.

Martínez, D. E., Iñiguez, A. R., Percell, K. M., Willner, J. B., Signorovitch, J., Campbell, R. D. , 2010. Phylogeny and biogeography of Hydra (Cnidaria: Hydridae) using mitochondrial and nuclear DNA sequences. Molecular Phylogenetics and Evolution 57(1), pp. 403–410.

Martinez, D. E., Steele, R. E., Simakov, O. , in preparation. The Zurich genome of Hydra vulgaris (working title).

Martínez, D. E. , 1998. Mortality Patterns Suggest Lack of Senescence in Hydra. Experimental Gerontology 33(3), pp. 217–225.

McClintock, B. , 1950. The origin and behavior of mutable loci in maize. Proceedings of the National Academy of Sciences 36(6), pp. 344–355.

McClintock, B. , 1984. The significance of responses of the genome to challenge. Science 226(4676), pp. 792–801.

McVean, G. , 2010. What drives recombination hotspots to repeat DNA in humans? Philosophical Transactions of the Royal Society B: Biological Sciences 365(1544), pp. 1213–1218.

Meinhardt, H. , 2012. Modeling pattern formation in hydra: a route to understanding essential steps in development. International Journal of Developmental Biology 56(6–8), pp. 447–462.

Modolo, L., Lerat, E. , 2014. Identification and analysis of transposable elements in genomic sequences. Genome Analysis: Current Procedures and Applications pp. 165–181.

Morgenstern, B. , 1999. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics 15(3), pp. 211–218.

Morgulis, A., Gertz, E. M., Schäffer, A. A., Agarwala, R. , 2006. A Fast and Symmetric DUST Implementation to Mask Low-Complexity DNA Sequences. Journal of Computational Biology 13(5), pp. 1028–1040.

Mouse Genome Sequencing Consortium. , 2002. Initial sequencing and comparative analysis of the mouse genome. Nature 420(6915), pp. 520–562.

Myers, S., Bottolo, L., Freeman, C., McVean, G., Donnelly, P. , 2005. A Fine-Scale Map of Recombination Rates and Hotspots Across the Human Genome. Science 310(5746), pp. 321–324.

Nakamura, Y., Tsiairis, C. D., Özbek, S., Holstein, T. W. , 2011. Autoregulatory and repressive inputs localize Hydra Wnt3 to the head organizer. Proceedings of the National Academy of Sciences of the United States of America 108(22), pp. 9137–9142.

NCBI. , 2017, October 24. NCBI's Genome Data Viewer (GDV) to replace Map Viewer.

Ohshima, K., Hamada, M., Terai, Y., Okada, N. , 1996. The 3' ends of tRNA-derived short interspersed repetitive elements are derived from the 3' ends of long interspersed repetitive elements. Molecular and Cellular Biology 16(7), pp. 3756–3764.

Ostertag, E. M., Kazazian, H. H. , 2005. LINEs in mind. Nature 435(7044), pp. 890–891.

Ou, S., Jiang, N. , 2018. LTR_retriever: A Highly Accurate and Sensitive Program for Identification of Long Terminal Repeat Retrotransposons1[OPEN]. Plant Physiology 176(2), pp. 1410–1422.

Oustimov, A., Vu, V. , 2014. Artificial neural networks in the cancer genomics frontier. Translational Cancer Research 3(3), pp. 191–201.

Pallas, P. S. , 1766. Elenchus zoophytorum sistens generum adumbrationes generaliores et specierum cognitarum succintas descriptiones, cum selectis auctorum synonymis /. Hagae-Comitum : Apud Petrum van Cleef,.

Pamilo, P., Nei, M. , 1988. Relationships between gene trees and species trees. Molecular Biology and Evolution 5(5), pp. 568–583.

Pavlicek, A., Gentles, A. J., Pačes, J., Pačes, V., Jurka, J. , 2006. Retroposition of processed pseudogenes: the impact of RNA stability and translational control. Trends in Genetics 22(2), pp. 69–73.

Peterson, K. J., Eernisse, D. J. , 2001. Animal phylogeny and the ancestry of bilaterians: inferences from morphology and 18S rDNA gene sequences. Evolution & Development 3(3), pp. 170–205.

Piégu, B., Bire, S., Arensburger, P., Bigot, Y. , 2015. A survey of transposable element classification systems – A call for a fundamental update to meet the challenge of their diversity and complexity. Molecular Phylogenetics and Evolution 86, pp. 90–109.

Price, A. L., Jones, N. C., Pevzner, P. A. , 2005. De novo identification of repeat families in large genomes. Bioinformatics 21(suppl_1), pp. i351–i358.

Putnam, N. H., O'Connell, B. L., Stites, J. C., Rice, B. J., Blanchette, M., Calef, R., Troll, C. J., Fields, A., Hartley, P. D., Sugnet, C. W., Haussler, D., Rokhsar, D. S., Green, R. E. , 2016. Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. Genome Research 26(3), pp. 342–350.

Putnam, N. H., Srivastava, M., Hellsten, U., Dirks, B., Chapman, J., Salamov, A., Terry, A., Shapiro, H., Lindquist, E., Kapitonov, V. V., Jurka, J., Genikhovich, G., Grigoriev, I. V., Lucas, S. M., Steele, R. E., Finnerty, J. R., Technau, U., Martindale, M. Q., Rokhsar, D. S. , 2007. Sea Anemone Genome Reveals Ancestral Eumetazoan Gene Repertoire and Genomic Organization. Science 317(5834), pp. 86–94.

Redi, C., Garagna, S., Zacharias, H., Zuccotti, M., Capanna, E. , 2001. The other chromatin. Chromosoma 110(3), pp. 136–147.

Rhoads, A., Au, K. F. , 2015. PacBio Sequencing and Its Applications. Genomics, Proteomics & Bioinformatics 13(5), pp. 278–289.

Richard, G.-F., Kerrest, A., Dujon, B. , 2008. Comparative Genomics and Molecular Dynamics of DNA Repeats in Eukaryotes. Microbiology and Molecular Biology Reviews 72(4), pp. 686–727.

Robbins, R. J., Benton, D., Snoddy, J. , 1995. Informatics and the Human Genome Project. IEEE Engineering in Medicine and Biology Magazine 14(6), pp. 694–701.

Saha, S., Bridges, S., Magbanua, Z. V., Peterson, D. G. , 2008a. Computational Approaches and Tools Used in Identification of Dispersed Repetitive DNA Sequences. Tropical Plant Biology 1(1), pp. 85–96.

Saha, S., Bridges, S., Magbanua, Z. V., Peterson, D. G. , 2008b. Empirical comparison of ab initio repeat finding programs. Nucleic Acids Research 36(7), pp. 2284–2294.

Saitou, N., Nei, M. , 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4(4), pp. 406–425.

Sandvik, H. , 2008. Tree thinking cannot taken for granted: challenges for teaching phylogenetics. Theory in Biosciences 127(1), pp. 45–51.

Scally, A., Durbin, R. , 2012. Revising the human mutation rate: implications for understanding human evolution. Nature Reviews Genetics 13(10), pp. 745–753.

Schmitt, A. O., Herzel, H. , 1997. Estimating the Entropy of DNA Sequences. Journal of Theoretical Biology 188(3), pp. 369–377.

Schuchert, P. , 2018. World Hydrozoa Database - Hydra magnipapillata Itô, 1947. Accessed at: http://www.marinespecies.org/hydrozoa/aphia.php?p=taxdetails&id=292543 on 2018-10-10.

Schwentner, M., Bosch, T. C. G. , 2015. Revisiting the age, evolutionary history and species level diversity of the genus Hydra (Cnidaria: Hydrozoa). Molecular Phylogenetics and Evolution 91, pp. 41–55.

She, X., Jiang, Z., Clark, R. A., Liu, G., Cheng, Z., Tuzun, E., Church, D. M., Sutton, G., Halpern, A. L., Eichler, E. E. , 2004. Shotgun sequence assembly and recent segmental duplications within the human genome. Nature 431(7011), pp. 927–930.

Shimizu, H., Takaku, Y., Zhang, X., Fujisawa, T. , 2007. The aboral pore of hydra: evidence that the digestive tract of hydra is a tube not a sac. Development Genes and Evolution 217(8), pp. 563–568.

Siomi, M. C., Sato, K., Pezic, D., Aravin, A. A. , 2011. PIWI-interacting small RNAs: the vanguard of genome defence. Nature Reviews Molecular Cell Biology 12(4), pp. 246–258.

Slotkin, R. K., Martienssen, R. , 2007. Transposable elements and the epigenetic regulation of the genome. Nature Reviews Genetics 8(4), pp. 272–285.

Smit, A. F. A., Hubley, R. , 2008. RepeatModeler Open-1.0. Available from http://www.repeatmasker.org.

Smit, A. F. A., Hubley, R., Green, P. , 1996. RepeatMasker Open-3.0. Available from http://www.repeatmasker.org.

Smith, T. F., Waterman, M. S. , 1981. Identification of common molecular subsequences. Journal of Molecular Biology 147(1), pp. 195–197.

Spannagl, M., Nussbaumer, T., Bader, K., Gundlach, H., Mayer, K. F. X. , 2017. PGSB/MIPS PlantsDB Database Framework for the Integration and Analysis of Plant Genome Data. In A. D. . van Dijk (Ed.), *Plant Genomics Databases: Methods and Protocols* (pp. 33–44). New York, NY: Springer New York.

Stanke, M., Steinkamp, R., Waack, S., Morgenstern, B. , 2004. AUGUSTUS: a web server for gene finding in eukaryotes. Nucleic Acids Research 32(suppl_2), pp. W309–W312.

Steele, R. E., David, C. N., Technau, U. , 2011. A genomic view of 500 million years of cnidarian evolution. Trends in Genetics 27(1), pp. 7–13.

Stein, L., Sternberg, P., Durbin, R., Thierry-Mieg, J., Spieth, J. , 2001. WormBase: network access to the genome and biology of Caenorhabditis elegans. Nucleic Acids Research 29(1), pp. 82–86.

Szostak, J. W., Wu, R. , 1980. Unequal crossing over in the ribosomal DNA of Saccharomyces cerevisiae. Nature 284(5755), pp. 426–430.

Tateno, Y., Nei, M., Tajima, F. , 1982. Accuracy of estimated phylogenetic trees from molecular data. Journal of Molecular Evolution 18(6), pp. 387–404.

Tommaso, P. D., Palumbo, E., Chatzou, M., Prieto, P., Heuer, M. L., Notredame, C. , 2015. The impact of Docker containers on the performance of genomic pipelines. PeerJ 3, pp. e1273.

van Berkum, N. L., Lieberman-Aiden, E., Williams, L., Imakaev, M., Gnirke, A., Mirny, L. A., Dekker, J., Lander, E. S. , 2010. Hi-C: A Method to Study the Three-dimensional Architecture of Genomes. Journal of Visualized Experiments : JoVE (39).

Volff, J.-N. , 2006. Turning junk into gold: domestication of transposable elements and the creation of new genes in eukaryotes. BioEssays 28(9), pp. 913–922.

Waldispühl, J., Zhang, E., Butyaev, A., Nazarova, E., Cyr, Y. , 2018. Storage, visualization, and navigation of 3D genomics data. Methods 142, pp. 74–80.

Wang, J., Kong, L., Gao, G., Luo, J. , 2013. A brief introduction to web-based genome browsers. Briefings in Bioinformatics 14(2), pp. 131–143.

Westheide, W., Rieger, G., Rieger, R. , 2013. Spezielle Zoologie. Teil 1: Einzeller und Wirbellose Tiere (3., überarb. u. aktual. Aufl. 2013). Berlin: Springer Spektrum.

Wicker, T., Sabot, F., Hua-Van, A., Bennetzen, J. L., Capy, P., Chalhoub, B., Flavell, A., Leroy, P., Morgante, M., Panaud, O., Paux, E., SanMiguel, P., Schulman, A. H. , 2007. A unified classification system for eukaryotic transposable elements. Nature Reviews Genetics 8, pp. 973.

Wong, W. Y., Simakov, O. , 2018. RepeatCraft: a meta-pipeline for repetitive element de-fragmentation and annotation. Bioinformatics (in review).

Wootton, J. C., Federhen, S. , 1996. Analysis of compositionally biased regions in sequence databases. In *Methods in Enzymology* (Vol. 266, pp. 554–571). Academic Press.

Wu, C. H., McLarty, J. W. , 2012. Neural Networks and Genome Informatics. Elsevier.

Yasuhara, J. C., Wakimoto, B. T. , 2006. Oxymoron no more: the expanding world of heterochromatic genes. Trends in Genetics 22(6), pp. 330–338.

Yuan, D., Huang, S. , 2017. Genetic equidistance at nucleotide level. Genomics 109(3), pp. 192–195.

Zarrella, I., Herten, K., Maes, G. E., Tai, S., Yang, M., Seuntjens, E., Ritschard, E. A., Zach, M., Styfhals, R., Sanges, R., Simakov, O., Ponte, G., Fiorito, G. , in review. The survey and reference assisted assembly of the Octopus vulgaris genome. Scientific Data.

Zuckerkandl, E., Pauling, L. , 1965. Evolutionary Divergence and Convergence in Proteins. In V. Bryson & H. J. Vogel (Eds.), *Evolving Genes and Proteins* (pp. 97–166). Academic Press.